

ADVANCES IN *SOFT COMPUTING* 44

Emilio Corchado
Juan M. Corchado
Ajith Abraham
Editors

Innovations in Hybrid Intelligent Systems

 Springer

Advances in Soft Computing

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Abraham Ajith, Yasuhiko Dote,
Takeshi Furuhashi, Mario Köppen,
Azuma Ohuchi, Yukio Ohsawa (Eds.)
*Soft Computing as Transdisciplinary Science
and Technology*, 2005
ISBN 978-3-540-25055-5

Barbara Dunin-Keplicz, Andrzej
Jankowski, Andrzej Skowron,
Marcin Szczuka (Eds.)
*Monitoring, Security, and Rescue
Techniques in Multiagent Systems*, 2005
ISBN 978-3-540-23245-2

Frank Hoffmann, Mario Köppen,
Frank Klawonn, Rajkumar Roy (Eds.)
*Soft Computing Methodologies and
Applications*, 2005
ISBN 978-3-540-25726-4

Mieczyslaw A. Kłopotek, Sławomir T.
Wierzchoń, Krzysztof Trojanowski
(Eds.)
*Intelligent Information Processing and
Web Mining*, 2005
ISBN 978-3-540-25056-2

Abraham Ajith, Bernard de Batts,
Mario Köppen, Bertram Nickolay (Eds.)
*Applied Soft Computing Technologies: The
Challenge of Complexity*, 2006
ISBN 978-3-540-31649-7

Mieczyslaw A. Kłopotek, Sławomir T.
Wierzchoń, Krzysztof Trojanowski
(Eds.)
*Intelligent Information Processing and
Web Mining*, 2006
ISBN 978-3-540-33520-7

Ashutosh Tiwari, Joshua Knowles,
Erel Avineri, Keshav Dahal,
Rajkumar Roy (Eds.)
Applications and Soft Computing, 2006
ISBN 978-3-540-29123-7

Bernd Reusch, (Ed.)
*Computational Intelligence, Theory and
Applications*, 2006
ISBN 978-3-540-34780-4

Miguel López-Díaz, María ç. Gil,
Przemysław Grzegorzewski, Olgierd
Hryniewicz, Jonathan Lawry
*Soft Methodology and Random Information
Systems*, 2006
ISBN 978-3-540-34776-7

Ashraf Saad, Erel Avineri, Keshav Dahal,
Muhammad Sarfraz, Rajkumar Roy (Eds.)
Soft Computing in Industrial Applications,
2007
ISBN 978-3-540-70704-2

Bing-Yuan Cao (Ed.)
Fuzzy Information and Engineering, 2007
ISBN 978-3-540-71440-8

Patricia Melin, Oscar Castillo,
Eduardo Gómez Ramírez, Janusz Kacprzyk,
Witold Pedrycz (Eds.)
*Analysis and Design of Intelligent Systems
Using Soft Computing Techniques*, 2007
ISBN 978-3-540-72431-5

Oscar Castillo, Patricia Melin,
Oscar Montiel Ross, Roberto Sepúlveda Cruz,
Witold Pedrycz, Janusz Kacprzyk (Eds.)
*Theoretical Advances and Applications of
Fuzzy Logic and Soft Computing*, 2007
ISBN 978-3-540-72433-9

Katarzyna M. Węgrzyn-Wolska,
Piotr S. Szczepaniak (Eds.)
Advances in Intelligent Web Mastering, 2007
ISBN 978-3-540-72574-9

Emilio Corchado, Juan M. Corchado,
Ajith Abraham (Eds.)
Innovations in Hybrid Intelligent Systems, 2007
ISBN 978-3-540-74971-4

Emilio Corchado, Juan M. Corchado,
Ajith Abraham (Eds.)

Innovations in Hybrid Intelligent Systems

 Springer

Editors

Prof. Emilio Corchado
Escuela Politécnica Superior
Campus Vena, Edificio C
Universidad de Burgos
C/Francisco de Vitoria s/n
09006 Burgos
Spain
E-mail: escorchado@ubu.es

Prof. Juan M. Corchado
Departamento de Informática y Automática
Facultad de Ciencias
Universidad de Salamanca
Plaza de la Merced S/N
37008 Salamanca
Spain

Prof. Ajith Abraham
Centre for Quantifiable Quality of Service in
Communication Systems (Q2S)
Centre of Excellence
Norwegian University of Science and Technology
O.S. Bragstads plass 2E
7491 Trondheim
Norway

Library of Congress Control Number: 2007935489

ISSN print edition: 1615-3871

ISSN electronic edition: 1860-0794

ISBN-10 3-540-74971-3 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-74971-4 Springer Berlin Heidelberg New York

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, reuse of illustrations, recitation, broadcasting, reproduction on microfilm 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 for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2007

Printed in Germany

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: by the authors and SPS using a Springer L^AT_EX macro package

Printed on acid-free paper SPIN: 12028535 89/SPS 5 4 3 2 1 0

Preface

The 2nd International Workshop on Hybrid Artificial Intelligence Systems (HAIS 2007) combines symbolic and sub-symbolic techniques to construct more robust and reliable problem solving models. Hybrid intelligent systems are becoming popular due to their capabilities in handling many real world complex problems, involving imprecision, uncertainty and vagueness, high-dimensionality. They provide us with the opportunity to use both, our knowledge and raw data to solve problems in a more interesting and promising way. This multidisciplinary research field is in continuous expansion in the artificial intelligence research community and we believe that CAEPIA 2007 is an excellent forum to run this event. HAIS 2007 provides an interesting opportunity to present and discuss the latest theoretical advances and real-world applications in this multidisciplinary research field.

This volume of *Advances in Soft Computing* contains accepted papers presented at HAIS 2007 held in University of Salamanca, Salamanca, Spain, November 2007.

The global purpose of HAIS conferences has been to provide a broad and interdisciplinary forum for Hybrid Artificial Intelligence Systems and Associated Learning Paradigms, which are playing increasingly important roles in an important number of applications fields.

Since its first edition in Brazil in 2006, HAIS is becoming a reference for researchers on fundamental and theoretical aspects related to Hybrid Artificial Intelligence Systems based on the hybrid use of Agents and Multiagent Systems, Bioinformatics and Bio-inspired Models, Fuzzy Systems, Artificial Vision, Artificial Neural Networks, Optimization Models and so on.

HAIS 2007 received 112 technical submissions and, the International Program Committee selected -after a rigorous peer-review- 62 papers for publication in this volume in the *Advances in Soft Computing* Series.

The large number of submitted papers is certainly a proof of the vitality and attractiveness of the fields related to HAIS, but it also shows a strong interest in the HAIS conferences.

HAIS 07 has also teamed up with *Neurocomputing Journal* for a special issue on Hybrid Artificial Intelligent Systems that has been scheduled for selected papers from HAIS 2007. The extended papers, together with contributed articles received in response to subsequent open calls, will go through further rounds of peer refereeing in the remits of this journal.

We would like to thank the work of our Programme Committee Members who performed admirably under tight deadline pressures. Our warmest special thanks go to our Keynote Speakers: Prof. Ajith Abraham, Norwegian University of Science and Technology, Norway, and Prof. José R. Dorronsoro, Autonomous University of Madrid, Spain.

Particular thanks go to the Organising Committee for his suggestions about organisation and promotion of HAIS 2007.

We wish to thank also Mr. Janusz Kacprzyk, Mr. Heather King, Mrs. Petra Jantzen and Mrs. Carmen Wolf from Springer for their help and collaboration in this demanding scientific publication project.

We thank as well all the authors and participants for their great contributions that made this conference possible and all the hard work worthwhile.

Finally we would to thank CAEPIA local committee Chair, Prof. Juan M. Corchado for his massive guidance, collaboration and support.

In this edition, at the time of writing this document, we were fortunate to receive support from Junta de Castilla y León and Spanish Ministry of Industry.

November 2007

Emilio Corchado
Juan M. Corchado
Ajith Abraham

Organizaton

General Co-chairs

Emilio Corchado -University of Burgos (Spain)

Juan M. Corchado - University of Salamanca (Spain)

Ajith Abraham - Norwegian University of Science and Technology (Norway)

Program Commitee

Ajith Abraham - Norwegian University of Science and Technology (Norway)

Jim Aiken - Plymouth Marine Laboratory (UK)

Davide Anguita - University of Genoa (Italy)

Bruno Apolloni - Universita degli Studi di Milano (Italy)

Akira Asano - Hiroshima University (Japan)

Javier Bajo - Pontifical University of Salamanca (Spain)

Bruno Baruque - University of Burgos (Spain)

Manuel Cantón - University of Almería (Spain)

Yiu-ming Cheung - Hong Kong Baptist University (Hong Kong)

P.J. Costa-Branco - Instituto Superior Tecnico (Portugal)

Rafael Corchuelo - University of Sevilla (Spain)

Bernard de Baets - Ghent University (Belgium)

Andre CPLF de Carvalho- University of São Paulo (Brazil)

José Dorronsoro - Autonomous University of Madrid (Spain)

Richard J. Duro - Universidade da Coruña (Spain)

Bogdan Gabrys - Bournemouth University (UK)

Matjaz Gams - Jozef Stefan Institute Ljubljana (Slovenia)

Xiao-Zhi Gao - Helsinki University of Technology (Finland)

José García - University of Alicante (Spain)

Paolo Gastaldo - University of Genoa (Italy)

Bernard Grabot - LGP/ENIT (France)

Manuel Graña - University of Pais Vasco (Spain)

Jerzy Grzymala-Busse - University of Kansas (USA)

Anne Håkansson - Uppsala University (Sweden)

Saman Halgamuge - The University of Melbourne (Australia)

Francisco Herrera - University of Granada (Spain)

Álvaro Herrero - University of Burgos (Spain)

R.J. Howlett - University of Brighton (UK)

Hisao Ishibuchi - Osaka Prefecture University (Japan)

Lakhmi Jain - University of South Australia (Australia)

Abonyi János - University of Veszprem (Hungary)

Juha Karhunen - Helsinki University of Technology (Finland)

Miroslav Karny - Academy of Sciences of Czech Republic (Czech Republic)
Andreas König - University of Kaiserslautern (Germany)
Frank Klawonn – Univ. of Applied Sciences Braunschweig/Wolfenbuettel (Germany)
Honghai Liu - University of Portsmouth (UK)
José Manuel Molina - University Carlos III of Madrid (Spain)
Ryohei Nakano - Nagoya Institute of Technology (Japan)
Anil Nerode - Cornell University (USA)
Ngoc Thanh Nguyen - Wroclaw University of Technology (Poland)
Maria do Carmo Nicoletti - Universidade Federal de Sao Carlos (Brazil)
José Francisco Martínez - INAOE (México)
Giancarlo Mauri - University of Milano - Bicocca (Italy)
José Mira - UNED (Spain)
Vasile Palade - Oxford University Computing Lab. (UK)
Gwi-Tae Park - Intelligent System Research Lab. (Korea)
Juan Pavón - University Complutense of Madrid (Spain)
Witold Pedrycz - University of Alberta (Canada)
Carlos Pereira - Universidade de Coimbra (Portugal)
Francesco Picasso - University of Genoa (Italy)
Jorge Posada - VICOMTech (Spain)
Guenter Raidl - Vienna University of Technology (Austria)
Perfecto Reguera - University of Leon (Spain)
Bernardete Ribeiro - University of Coimbra (Portugal)
Ramón Rizo - University of Alicante (Spain)
Rajkumar Roy - Cranfield University (UK)
Dymirt Ruta - British Telecom (UK)
Juan Sánchez - University of Salamanca (Spain)
Dante Tapia - University of Salamanca (Spain)
Eiji Uchino - Yamaguchi University (Japan)
Tzai-Der Wang - Cheng Shiu University (Taiwan)
Lei Xu - Chinese University of Hong Kong (Hong Kong)
Ronald R. Yager - Iona College (USA)
Xin Yao - University of Birmingham (UK)
Hujun Yin - University of Manchester (UK)
Rodolfo Zunino - University of Genoa (Italy)

Organizing Commite

Emilio Corchado - University of Burgos (Spain)
Juan M. Corchado - University of Salamanca (Spain)
Ajith Abraham - Norwegian University of Science and Technology (Norway)
Álvaro Herrero - University of Burgos (Spain)
Bruno Baruque - University of Burgos (Spain)
Javier Bajo - Pontifical University of Salamanca (Spain)
M.A. Pellicer - University of Burgos (Spain)

Contents

Agents and Multiagent Systems

Analysis of Emergent Properties in a Hybrid Bio-inspired Architecture for Cognitive Agents

Oscar J. Romero, Angélica de Antonio 1

Using Semantic Causality Graphs to Validate MAS Models

Guillermo Viguera, Jorge J. Gómez, Juan A. Botía, Juan Pavón 9

A Multiagent Framework to Animate Socially Intelligent Agents

Francisco Grimaldo, Miguel Lozano, Fernando Barber 17

Context Aware Hybrid Agents on Automated Dynamic Environments

Juan F. de Paz, Sara Rodríguez, Juan M. Sánchez, Ana de Luis, Juan M. Corchado 25

Sensitive Stigmergic Agent Systems – A Hybrid Approach to Combinatorial Optimization

Camelia Chira, Camelia-M. Pinteá, D. Dumitrescu 33

Fuzzy Systems

Agent-Based Social Modeling and Simulation with Fuzzy Sets

Samer Hassan, Luis Garmendia, Juan Pavón 40

Stage-Dependent Fuzzy-valued Loss Function in Two-Stage Binary Classifier

Robert Burduk 48

A Feature Selection Method Using a Fuzzy Mutual Information Measure	
<i>Javier Grande, María del Rosario Suárez, José Ramón Villar</i>	56
Interval Type-2 ANFIS	
<i>Gerardo M. Mendez, Ma. De Los Angeles Hernandez</i>	64
A Vision-Based Hybrid Classifier for Weeds Detection in Precision Agriculture Through the Bayesian and Fuzzy k-Means Paradigms	
<i>Alberto Tellaeché, Xavier-P. BurgosArtizzu, Gonzalo Pajares, Angela Ribeiro</i>	72
<hr/>	
Artificial Neural Networks	
<hr/>	
Development of Multi-output Neural Networks for Data Integration – A Case Study	
<i>Paul Trundle, Daniel Neagu, Marian Craciun, Qasim Chaudhry</i>	80
Combined Projection and Kernel Basis Functions for Classification in Evolutionary Neural Networks	
<i>P.A. Gutiérrez, C. Hervás, M. Carbonero, J.C. Fernández</i>	88
Modeling Heterogeneous Data Sets with Neural Networks	
<i>Lluís A. Belanche Muñoz</i>	96
A Computational Model of the Equivalence Class Formation Psychological Phenomenon	
<i>José Antonio Martín H., Matilde Santos, Andrés García, Javier de Lope</i>	104
Data Security Analysis Using Unsupervised Learning and Explanations	
<i>G. Corral, E. Armengol, A. Fornells, E. Golobardes</i>	112
Finding Optimal Model Parameters by Discrete Grid Search	
<i>Álvaro Barbero Jiménez, Jorge López Lázaro, José R. Dorronsoro</i>	120
<hr/>	
Clustering and Multiclassifier Systems	
<hr/>	
A Hybrid Algorithm for Solving Clustering Problems	
<i>Enrique Domínguez, José Muñoz</i>	128
Clustering Search Heuristic for the Capacitated p-Median Problem	
<i>Antonio Augusto Chaves, Francisco de Assis Correa, Luiz Antonio N. Lorena</i>	136

Experiments with Trained and Untrained Fusers*Michal Wozniak* 144**Fusion of Visualization Induced SOM***Bruno Baruque, Emilio Corchado* 151

Robots

Open Intelligent Robot Controller Based on Field-Bus and RTOS*Zonghai Chen, Haibo Wang* 159**Evolutionary Controllers for Snake Robots Basic Movements***Juan C. Pereda, Javier de Lope, Maria Victoria Rodellar* 167**Evolution of Neuro-controllers for Multi-link Robots***José Antonio Martín H., Javier de Lope, Matilde Santos* 175**Solving Linear Difference Equations by Means of Cellular Automata***A. Fúster-Sabater, P. Caballero-Gil, O. Delgado* 183

Metaheuristics and Optimization Models

Automated Classification Tree Evolution Through Hybrid Metaheuristics*Miroslav Bursa, Lenka Lhotska* 191**Machine Learning to Analyze Migration Parameters in Parallel Genetic Algorithms***S. Muelas, J.M. Peña, V. Robles, A. LaTorre, P. de Miguel* 199**Collaborative Evolutionary Swarm Optimization with a Gauss Chaotic Sequence Generator***Rodica Ioana Lung, D. Dumitrescu* 207**A New PSO Algorithm with Crossover Operator for Global Optimization Problems***Millie Pant, Radha Thangaraj, Ajith Abraham* 215**Solving Bin Packing Problem with a Hybridization of Hard Computing and Soft Computing***Laura Cruz-Reyes, Diana Maritza Nieto-Yáñez, Pedro Tomás-Solis, Guadalupe Castilla Valdez* 223

Design of Artificial Neural Networks Based on Genetic Algorithms to Forecast Time Series
Juan Peralta, German Gutierrez, Araceli Sanchis 231

Experimental Analysis for the Lennard-Jones Problem Solution
Héctor J. Fraire Huacuja, David Romero Vargas, Guadalupe Castilla Valdez, Carlos A. Camacho Andrade, Georgina Castillo Valdez, José A. Martínez Flores 239

Application of Genetic Algorithms to Strip Hot Rolling Scheduling
Carlos A. Hernández Carreón, Héctor J. Fraire Huacuja, Karla Espriella Fernandez, Guadalupe Castilla Valdez, Juana E. Mancilla Tolama 247

Synergy of PSO and Bacterial Foraging Optimization – A Comparative Study on Numerical Benchmarks
Arijit Biswas, Sambarta Dasgupta, Swagatam Das, Ajith Abraham 255

Artificial Vision

Bayes-Based Relevance Feedback Method for CBIR
Zhiping Shi, Qing He, Zhongzhi Shi 264

A Novel Hierarchical Block Image Retrieval Scheme Based Invariant Features
Mingxin Zhang, Zhaogan Lu, Junyi Shen 272

A New Unsupervised Hybrid Classifier for Natural Textures in Images
María Guijarro, Raquel Abreu, Gonzalo Pajares 280

Visual Texture Characterization of Recycled Paper Quality
José Orlando Maldonado, David Vicente Herrera, Manuel Graña Romay 288

Case-Based Reasoning

Combining Improved FYDPS Neural Networks and Case-Based Planning – A Case Study
Yanira de Paz, Quintín Martín, Javier Bajo, Dante I. Tapia 296

CBR Contributions to Argumentation in MAS
Stella Heras, Vicente Julián, Vicente Botti 304

Case-Base Maintenance in an Associative Memory Organized by a Self-Organization Map <i>A. Fornells, E. Golobardes</i>	312
Hybrid Multi Agent-Neural Network Intrusion Detection with Mobile Visualization <i>Álvaro Herrero, Emilio Corchado, María A. Pellicer, Ajith Abraham</i>	320
<hr/>	
Learning Models	
<hr/>	
Knowledge Extraction from Environmental Data Through a Cognitive Architecture <i>Salvatore Gaglio, Luca Gatani, Giuseppe Lo Re, Marco Ortolani</i>	329
A Model of Affective Entities for Effective Learning Environments <i>Jose A. Mocholí, Javier Jaen, Alejandro Catalá</i>	337
<hr/>	
Bioinformatics	
<hr/>	
Image Restoration in Electron Cryotomography – Towards Cellular Ultrastructure at Molecular Level <i>J.J. Fernández, S. Li, R.A. Crowther</i>	345
SeqTrim – A Validation and Trimming Tool for All Purpose Sequence Reads <i>Juan Falgueras, Antonio J. Lara, Francisco R. Cantón, Guillermo Pérez-Trabado, M. Gonzalo Claros</i>	353
A Web Tool to Discover Full-Length Sequences – Full-Longther <i>Antonio J. Lara, Guillermo Pérez-Trabado, David P. Villalobos, Sara Díaz-Moreno, Francisco R. Cantón, M. Gonzalo Claros</i>	361
Discovering the Intrinsic Dimensionality of BLOSUM Substitution Matrices Using Evolutionary MDS <i>Juan Méndez, Antonio Falcón, Mario Hernández, Javier Lorenzo</i>	369
Autonomous FYDPS Neural Network-Based Planner Agent for Health Care in Geriatric Residences <i>Juan F. de Paz, Yanira de Paz, Javier Bajo, Sara Rodríguez, Juan M. Corchado</i>	377
Structure-Preserving Noise Reduction in Biological Imaging <i>J.J. Fernández, S. Li, V. Lucic</i>	385

Ensemble of Support Vector Machines to Improve the Cancer Class Prediction Based on the Gene Expression Profiles <i>Ángela Blanco, Manuel Martín-Merino, Javier De Las Rivas</i>	393
NATPRO-C13 – An Interactive Tool for the Structural Elucidation of Natural Compounds <i>Roberto Theron, Esther del Olmo, David Díaz, Miguel Vaquero, José Francisco Adserias, José Luis López-Pérez</i>	401
Application of Chemoinformatic Tools for the Analysis of Virtual Screening Studies of Tubulin Inhibitors <i>Rafael Peláez, José Luis López, Manuel Medarde</i>	411
Identification of Glaucoma Stages with Artificial Neural Networks Using Retinal Nerve Fibre Layer Analysis and Visual Field Parameters <i>Emiliano Hernández Galilea, Gustavo Santos-García, Inés Franco Suárez-Bárcena</i>	418
Dimensional Reduction in the Protein Secondary Structure Prediction – Nonlinear Method Improvements <i>Gisele M. Simas, Sílvia S.C. Botelho, Neusa Grando, Rafael G. Colares</i>	425
<hr/>	
Other Applications	
<hr/>	
Focused Crawling for Retrieving Chemical Information <i>Zhaojie Xia, Li Guo, Chunyang Liang, Xiaoxia Li, Zhangyuan Yang</i>	433
Optimal Portfolio Selection with Threshold in Stochastic Market <i>Shuzhi Wei, Zhongxing Ye, Genke Yang</i>	439
Classification Based on Association Rules for Adaptive Web Systems <i>Saddys Segrera, María N. Moreno</i>	446
Statistical Selection of Relevant Features to Classify Random, Scale Free and Exponential Networks <i>Laura Cruz Reyes, Eustorgio Meza Conde, Tania Turrubiates López, Claudia Guadalupe Gómez Santillán, Rogelio Ortega Izaguirre</i>	454
Open Partner Grid Service Architecture in eBusiness <i>Hao Gui, Hao Fan</i>	462

An Architecture to Support Programming Algorithm Learning by Problem Solving <i>Francisco Jurado, Miguel A. Redondo, Manuel Ortega</i>	470
Explain a Weblog Community <i>Alberto Ochoa, Antonio Zamarrón, Saúl González, Arnulfo Castro, Nahitt Padilla</i>	478
Implementing Data Mining to Improve a Game Board Based on Cultural Algorithms <i>Alberto Ochoa, Saúl González, Arnulfo Castro, Nahitt Padilla, Rosario Baltazar</i>	486
Author Index	495

Hybrid Artificial Intelligence Systems

Ajith Abraham

Norwegian University of Science and Technology, Norway
ajith.abraham@ieee.org

Abstract. The emerging need for hybrid intelligent systems is currently motivating important research and development work. It is well known that the intelligent systems, which can provide human like expertise such as domain knowledge, uncertain reasoning, and adaptation to a noisy and time varying environment, are important in tackling practical computing problems. The integration of different learning and adaptation techniques, to overcome individual limitations and achieve synergetic effects through hybridization or fusion of these techniques, has in recent years contributed to a large number of new intelligent system designs. These ideas have led to the emergence of several different kinds of intelligent system architectures.

This talk presents some of the generic hybrid architectures which have evolved over the past decade in the hybrid intelligent systems community. We further attempt to discuss the importance of these architectures with an emphasis on the best practices for selection and combination of intelligent methods. Two application examples will be presented to demonstrate how such systems could be used for solving real world problems.

Analysis of Emergent Properties in a Hybrid Bio-inspired Architecture for Cognitive Agents

Oscar J. Romero and Angélica de Antonio

Politécnica de Madrid University, LSIS Department,
Boadilla del monte, 28660, Madrid, Spain
ojrlopez@hotmail.com, angelica@fi.upm.es

Abstract. In this work, a hybrid, self-configurable, multilayered and evolutionary architecture for cognitive agents is developed. Each layer of the subsumption architecture is modeled by one different Machine Learning System MLS based on bio-inspired techniques. In this research an evolutionary mechanism supported on Gene Expression Programming to self-configure the behaviour arbitration between layers is suggested. In addition, a co-evolutionary mechanism to evolve behaviours in an independent and parallel fashion is used. The proposed approach was tested in an animat environment using a multi-agent platform and it exhibited several learning capabilities and emergent properties for self-configuring internal agent's architecture.

Keywords: Subsumption Architecture, Hybrid Behaviour Co-evolution, Gene Expression Programming, Artificial Immune Systems, Extended Classifier Systems, Neuro Connectionist Q-Learning Systems.

1 Introduction

In the last decades, Cognitive Architectures have been an area of study that collects disciplines as artificial intelligence, human cognition, psychology and more, to determine necessary, sufficient and optimal distribution of resources for the development of agents exhibiting emergent intelligence. One of the most referenced is the Subsumption Architecture proposed by Brooks [1].

According to Brooks [1], the Subsumption Architecture is built in layers. Each layer gives the system a set of pre-wired behaviours, where the higher levels build upon the lower levels to create more complex behaviours: The behaviour of the system as a whole is the result of many interacting simple behaviours. Another characteristic is its lack of a world model, which means that its responses are always and only reflexive as proposed by Brooks.

However, Subsumption Architecture results in a tight coupling of perception and action, producing high reactivity, poor adaptability, no learning of new environments, no internal representation and the need that all patterns of behaviours are pre-wired.

The present research focuses on developing a Hybrid Multilayered Architecture for Cognitive Agents based on Subsumption theory. Additionally this work proposes an Evolutionary Model which allows the Agent to self-configure and evolve its processing layers through the definition of inhibitory and suppressor processes, between behaviour layers instead of having a pre-configured structure of them. On the

other hand, instead of using an Augmented Finite Machine System as Subsumption theory states [1] where no internal representation is done, in this paper we propose that each behaviour layer is driven by a different bio-inspired machine learning system (chosen from a repertoire where behaviour co-evolution occurs) which learns from the environment and generates an internal world-model by means of an unsupervised and reinforced learning.

The remainder of the paper is organized as follows. The evolutionary approach for self-configurable cognitive agents is detailed in Section 2. Section 3 discusses the experimental results and emergent properties obtained. Finally concluding remarks are shown in Section 4.

2 Proposed Hybrid, Self-configurable and Evolutionary Approach for Cognitive Agents

In order to design a hybrid, self-configurable, scalable, adaptable, and evolutionary architecture for cognitive systems which exhibits emergent behaviours and learning capabilities, the proposed work is explained as follows.

Consider a virtual environment where there exists several agents interacting with objects, food, each others, etc.; it arises some mayor questions and constraints:

- Change of environmental conditions, i.e. about objects: quantity, type of object, location, size, etc., about other agents: intentions and desires, goals, etc.
- There are a variable number of desired behaviours: avoiding-obstacles, wandering, feeding, hunting, escaping-from-depredators, etc.
- How many behaviours can be integrated into a single agent? And how can agents arbitrate these behaviours?
- When does a single agent know if it has to inhibit or suppress a behaviour without using a pre-established applicability predicate or rule?
- How can a behaviour, that drives one of the layers in a single multilayered agent, generate a model of the world, couple with the environment via the agent's sensors and actuators, learn from its own interaction with the environment and receive a reinforcement of its actions, so the internal state of the behaviour evolve?

These questions address the following proposed approach of a hybrid, self-configurable and bio-inspired architecture for cognitive agents. The Fig. 1 shows a hybrid architecture from which all the questions mentioned above can be solved. An internal architecture based on subsumption principles but with few variations can be observed in every agent:

- Each processing layer is connected randomly with one different bio-inspired learning machine system (Extended Classifier System XCS [2], Artificial Immune System AIS [3], Neuro Connectionist Q-Learnig System NQL [4], Learning Classifier System LCS [5] and scalable to others) which replaces the typical Augmented Finite State Machines proposed by Brook's architecture [1].
- After being trained in the agent, every behaviour is sent to a behaviour repertoire according to its category, where a co-evolutionary mechanism based on [6] is applied so that every behaviour not only will learn in a local way inside of each

agent but also will evolve in a global way, and afterwards it will be selected by another agent in the next generation.

- There is an evolutionary process driven by a Gene Expression Programming Algorithm GEP [7], which is in charge of self-configuring the agent's behaviour arbitration: defining the number of layers, behaviours, connections and hierarchies between them (inhibition, suppression, aggregation, etc.). GEP Algorithm generates a set of applicability predicates per agent where it is determined which behaviour will be activated at a certain situation and its time of activation.

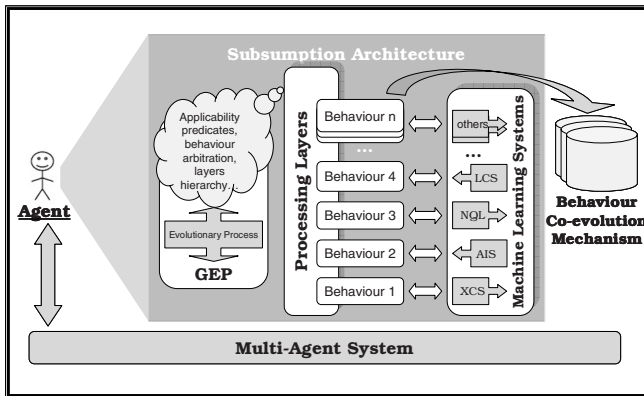


Fig. 1. Hybrid and Evolutionary Architecture for Cognitive Agents

2.1 Hybrid Architecture: Behaviours Driven by Machine Learning Systems

Every behaviour layer in the multilayered architecture is associated to a Machine Learning System MLS, that allows the architecture being hybrid and not only reactive since each behaviour will be able to exert deliberative processes using the acquired knowledge. Besides, this mechanism gives plasticity to the architecture because every behaviour “learns” in an unsupervised, independent and parallel way through its interaction with the environment, generating internal representations, rules and both specific and generalized knowledge. This mechanism is favored by the MLSs characteristics: robustness, fault tolerance, use of bio-inspired techniques, adaptability and it does not require a previous definition of knowledge (unsupervised learning). There are two principles formulated by Stone [8] that have motivated this proposed layered learning approach:

- “Layered learning is designed for domains that are too complex for learning a mapping directly from an agent’s sensory inputs to its actuator outputs. Instead the layered learning approach consists of breaking a problem down into several behavioral layers and using MLSs at each level. Layered learning uses a bottom up incremental approach to hierarchical task decomposition.”
- “MLS is used as a central part of layered learning to exploit data in order to train and or adapt the overall system. MLS is useful for training behaviours that are difficult to fine-tune manually.”

To make the hybridization process consistent, a common interface for all MLSs (XCS [2], AIS [3], NQL [4], LCS [5], etc.) is proposed so, although each MLS has a different internal process, they all have a similar structure that it lets the system to be scalable introducing new MLSs if is required and connecting them in an easy way with each behaviour layer in the agent's multilayered architecture.

2.2 Hybrid Behaviour Co-evolution: Evolving Globally

A co-evolutionary mechanism [6] is proposed to evolve each type of behaviour separately in its own genetic pool. Most evolutionary approaches use a single population where evolution is performed; instead, the behaviours are discriminated in categories and make them evolve in separate behaviour pools without any interaction. First, each agent defines a specific set of behaviours that builds its own multilayered structure. For each required agent's behaviour, a behaviour instance is chosen from the pool (this instance is connected with one MLS). Subsequently each agent will interact with the environment and each agent's behaviour will learn a set of rules and generate an own knowledge base. After certain period of time a co-evolutionary mechanism is activated. For each behaviour pool is applied a probabilistic selection method of behaviours where those behaviours that had the best performance (fitness) will have more probability to reproduce. Then, a crossover genetic operator is applied between each pair of selected behaviours: a portion of knowledge acquired by each agent's behaviour (through its MLS) is interchanged with the other one. Finally, new random rules are generated until complete the maximum size of rules that behaviours can have in their own knowledge base, so a new pair of behaviours is created and left in the corresponding behaviour pool to be selected by an agent in the next generation.

2.3 Self-configurable Architecture: Behaviour Arbitration

If each agent has an arbitrary set of behaviours, how to determine: the interaction between them, the hierarchy levels, the Subsumption process (inhibition and suppression) and the necessary layers to do an adequate processing? These questions are solved next. The internal multilayered structure of each agent is decomposed in atomic components which can be estimated and used to find the optimal organization of behaviours during the agent's lifetime [8]. The main goal is that the agent in an automatic way self-configures its own behaviours structure. The GEP algorithm proposed by Ferreira [7] is used to evolve internal structures of each agent and generate a valid arbitration of behaviours. GEP uses two sets: a function set and a terminal set [7]. In our work, the proposed function set is: {AND, OR, NOT, IFMATCH, INHIBIT, SUPPRESS}. The AND, OR and NOT functions are logic operators used to group and exclude subsets of objects, behaviours, etc. The conditional function IFMATCH is a typical applicability predicate that matches with a specific problem situation. This function has four arguments; the first three arguments belong to the rule's antecedent: the first indicates what object is sensed, second one is the activated sensor and the third argument is the current behaviour running on the agent. If the first three arguments are applicable then the fourth argument, the rule's consequent, is executed. The fourth argument should be a INHIBIT or SUPPRESS function, or maybe and AND/OR function if more elements are necessary. The

INHIBIT/SUPPRESS functions have two arguments (behaviourA, behaviourB) and indicate that behaviourA inhibits/suppresses behaviourB. On the other hand, the terminal set is composed by the behaviour set, the environmental element set (objects, agents, food, etc.) and an agent's sensor set. Additionally "wildcard" elements are included so whichever sensor, behaviour or object can be referenced.

Each agent has a chromosome with information about its self structure, i.e. the agent A can have the following chromosome (applicability predicate):

IFMATCH: There is a wall AND is Activated *looking-for-food* behaviour AND Reading by *sensor1*

THEN: *Avoiding-obstacle* INHIBIT *wandering* AND *looking-for-food* behaviours, Analyzing this rule we can infer that the agent has three behaviour layers: avoiding-obstacle, wandering and looking-for-food, and the two last ones are inhibited by the first one when sensor1 identifies a wall in front of the agent.

However, these chromosomes don't have always a valid syntax, so the GEP mechanism is used to evolve the chromosome until it becomes in a valid syntactic rule. Each gene is become to a tree representation and then a genetic operator set is applied between genes of the same agent and genes of other agents [7]: selection, mutation, root transposition, gene transposition, two-point recombination and gene recombination. After certain number of evolutionary generations, valid and better adapted agent's chromosomes are generated.

2.4 Emergent Properties of the Architecture

Brooks postulates in his paper [1] the possibility that intelligence can emerge out of a set of simple, loosely coupled behaviours, and emergent properties arise (if at all) due to the complex dynamics of interactions among the simple behaviours and that this emergence is to a large extent accidental. According to that, the proposed architecture articulates a behaviour set that learns about environmental conditions in an independent and parallel manner, and on the other hand evolve inside a categorized pool. Each simple behaviour can be applied to a subset of specific situations but not to the whole problem space, however the individual level interaction between behaviours (inside each agent) allows covering a wider range of problem states and some characteristics are generated: robustness, knowledge redundancy, fault tolerance and a big plasticity level, so emergent properties in the individual and inside of the society (Multi-agent systems) appear. Then, the emergent properties arise from three points of view in a bottom-up approach:

- Atomic: in each behaviour of the multilayered architecture, when the associated MLS learns from the environment, in an automate way.
- Individual: the agent self-configures its internal structure (chromosome), hierarchy and arbitration of behaviours through an evolutionary process driven by GEP.
- Social: a hybrid behaviour co-evolution mechanism is applied to all agent's behaviours, so behaviours learn not only themselves via the MLS associated but also cooperating and sharing the acquired knowledge between them.

It is important to notice that emergence in different levels, from atomic to social point of view, provokes an overall emergence of the system, where some kind of

intelligence we hope to arise. The experimentation focused on discovering some characteristics of identity in the animats, i.e. we expected to see some animat agents behaving like depredators and others behaving like preys after the system has evolved several generations. Depredators should include some behaviours like avoiding-obstacles, looking-for-water, persecuting-preys, rounding-up, hunting-preys, etc. and Preys should include some behaviours like avoiding-obstacles, looking-for-food, looking-for-water, hiding, escaping, etc. Nevertheless, expected emergent properties can vary according to the environment and the behaviour set.

3 Experimentation

An artificial life environment called Animat (animal + robot) [5] is proposed to test the experiments. The environment simulates virtual agents competing for getting food and water, avoiding obstacles, hunting, escaping from depredators, etc. This animat environment was selected because is more friendly to see emergent behaviours but it is not the only environment where the proposed architecture could be applicable.

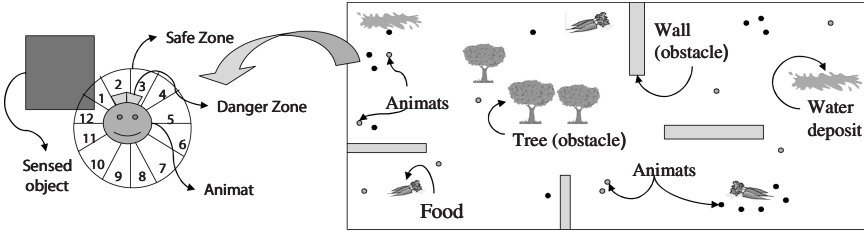


Fig. 2. Simulated Animat Environment and sensor animat distribution

Each animat controlled by an agent disposes a set of 14 proximity sensors (see Fig. 2) simulating a limited sight sense. 12 sensors read a safe zone and 2 sensors read a danger zone (to avoid collisions), as proposed by D. Romero [9]. Additionally, an environment with objects, food, water deposits, animats, obstacles, traps, etc. is simulated. Several performance experiments of all algorithms (XCS, AIS; NQL, LCS and GEP), analysis of learning convergence, fitness progression and analysis of syntactically well-formed genes in GEP, were done in our research, however, we only going to present the relevant results of analyzed emergent properties next.

Analysis of evolved architectures: after the whole system has evolved during a specific number of generations, we have analyzed the final structures of the best adapted agents where emergent properties arose.

Fig. 3 shows the genotype (Expression Trees - ET in GEP) and phenotype respectively of an initial architecture of a random agent without any evolutionary phase; in contrast, fig. 4 shows the genotype and phenotype of the evolved architecture of the same selected agent. In fig. 3.b the chromosome represents four behaviours: looking-for-water, looking-for-food, avoiding-obstacles and hiding, where l-f-w inhibits l-f-f and hiding and l-f-w suppresses a-o, but there is a

contradictory process when l-f-f tries to suppress l-f-w and l-f-f has been already inhibited by l-f-w (see fig. 3.b). This is solved with the evolved architecture in fig. 4.b, which proposes a new structure adding escaping-from-depredators behaviour and excluding hiding behaviour.

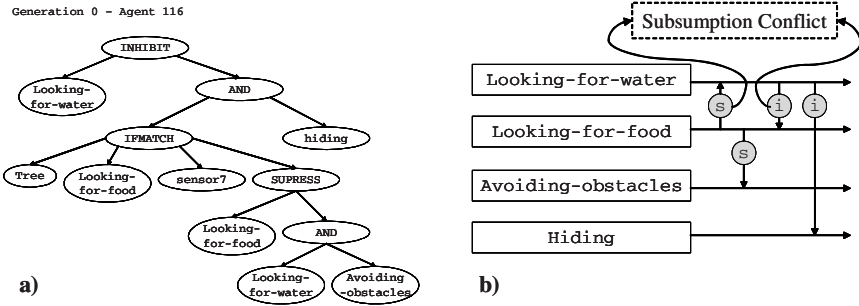


Fig. 3. Genotype and Phenotype of an initial Agent's Architecture

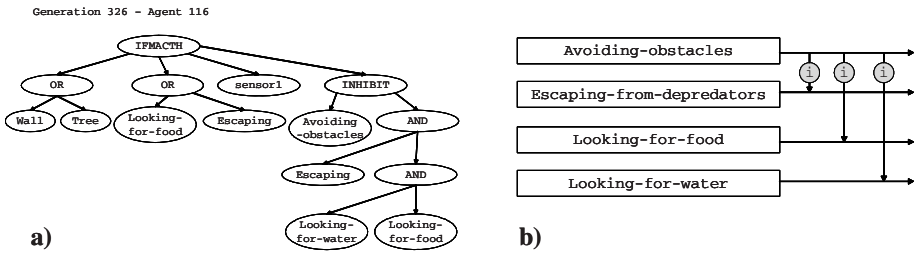


Fig. 4. Genotype and Phenotype of the Agent's Architecture after 326 evolutionary generations

As depicted in fig. 4.b, the initial contradictory inhibitory/suppressor processes in the agent's architecture (see fig. 3.b) are solved, and only hierarchical inhibitory processes are proposed by the evolved architecture. Furthermore, we can deduce too that evolved architecture has collected a set of specific behaviours becoming the agent to an animat with a identity of prey. It is important to notice in evolved architecture that *escaping-from-depredators* behaviour inhibits *looking-for-food* and *looking-for-water* behaviours but if the animat is escaping and its *sensor7* reads a *wall* or a *tree*, then *escaping-from-depredators* behaviour is inhibited by *avoiding-obstacles* behaviour until the obstacle is not in front of the animat anymore, and after that the animat continues its getaway, so we can say that emergent behaviour arises.

Finally, the experimentation demonstrate that specific parameter configurations in MLSs, GEP and Co-evolutionary mechanism are required to reach certain robustness, learning and adaptation capacities in the overall system. Nevertheless, emergent properties didn't arise always or in a quick way, in several experiments animats died quickly and they couldn't learn to survive.

4 Conclusions

The integration of multiple Machine Learning Systems in controlling the behaviours layers of a hybrid Subsumption Architecture approach, instead of using the typical Augmented Finite State Machines, have demonstrated important advantages in learning about the world of the agent, making internal knowledge representations and adapting to environmental changes.

The evolutionary and learning mechanisms used in this work, provided a plasticity feature allowing the agent to self-configure its own multilayered behaviour-based architecture; thus it can avoid creating exhaustive and extensive knowledge bases, pre-wired behaviour-based structures and pre-constrained environments.

In our future work we expect to continue working on designing more adaptive and self-configurable architectures, using fuzzy techniques in the MLSs to improve the sensors readings. In the future, one concrete application of this research will be the development of a Cognitive Module for Emotive Pedagogical Agents where the agent will be able to self-learn about its own perspectives, believes, desires, intentions, emotions, skills and perceptions.

Acknowledgments. Supported by the Programme Alban, the European Union Programme of High Level Scholarships for Latin America, scholarship No. E05D056455CO”.

References

1. R.A. Brooks, A Robust Layered Control System For A Mobile Robot, *IEEE Journal Of Robotics And Automation*, RA-2 (1986), 14-23.
2. S.W. Wilson, State of {XCS} Classifier System Research, *Lecture Notes in Computer Science*, 1813 (2000), 63-81.
3. L. N. de Castro, J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Ed. Springer (2002)
4. V. Kuzmin, Connectionist Q-learning in Robot Control Task, *Proceedings of Riga Technical University* (2002), 112-121.
5. J.H. Holland, *Induction, Processes of Inference, Learning and Discovery*, Mich:Addison-Wesley (1953).
6. Farahmand, Hybrid Behavior Co-evolution and Structure Learning in Behavior-based Systems, *IEEE Congress on EC*, Vancouver (2006), 979-986.
7. Ferreira, Gene Expression Programming: A new adaptive algorithm for solving problems, *Complex Systems*, forthcoming, (2001).
8. P.Stone, Layered Learning in Multiagent Systems, Thesis CS-98-187 (1998)
9. D. Romero, L. Niño, An Immune-based Multilayered Cognitive Model for Autonomous Navigation, *IEEE Congress on EC*, Vancouver (2006), 1115-1122.

Using Semantic Causality Graphs to Validate MAS Models

Guillermo Viguera, Jorge J. Gómez, Juan A. Botía, and Juan Pavón

Facultad de Informática
Universidad de Murcia
Spain
guivigon@dif.um.es

Abstract. Multi-agent systems methodologies define tasks and tools covering all the development cycle. Modelling and implementation activities are well supported and experimented. However, there has been few effort invested in debugging and validation techniques for MAS compared to modelling and implementation stages. This paper addresses the lack of tools to debug and validate a MAS. We propose a visual tool to debug and validate some multi-agent system specification, capturing causality among messages sent by agents and showing semantic information related to interaction protocols used by agents.

Keywords: Intelligent agent-based systems, MAS debug, MAS validation.

1 Introduction

Modelling multi-agent systems is a delicate task prone to errors. Most of the times, it is a problem of lack of specialized verification and validation tools. One of design parts that developers should specially take care of, is interaction protocols specification. By means of these protocols, agents communicate to achieve their individual or common goals. So analyzing interactions provides valuable information to determine agent behavior. Agents methodologies offer to developers tasks and activities, along MAS development cycle, to model multi-agent systems. However the number of debug and validation tools offered by these methodologies is not comparable as the number of development tools offered. Visual tools that represent an abstract view of the system can help developers to perform debug and validation tasks. In this way causality is an important concept from distributed systems that can be adapted to multi-agent systems. Causality of a concrete event (i.e. a message exchanged, an undesired result of the interaction studied, an unexpected conversation, etc.), in the context of multi-agent interactions, may be loosely defined as the cause which directly or indirectly led to generate the event. What we propose in this paper is adding, to causality graphs proposed in [6], semantic information specified inside some interaction protocol. Semantic information defines how received and sent messages affect to agents, i.e. what actions are executed and how agents mental state are modified by those actions. In order to add semantic information, we propose to use the meta-models defined in INGENIAS methodology [5] and the information provided by those meta-models about the MAS. Using this information, developer

will be able to perform a better diagnose and find the reason of possible error in the agents behavior more quickly.

The rest of the paper is organized as follows. Section 2 introduces the notion of causality in the context of a MAS interaction. Section 3 describes INGENIAS methodology. Section 4 describes a MAS example modelled using INGENIAS and how semantic causality graphs can help developer to debug and verify it. Section 5 puts our causality graph in the appropriate context and finally, section 6 enumerates most important conclusions and open issues.

2 Representing Causality Through MAS Interactions

In order to integrate debug, validation and verification in the MAS development cycle, it is important to offer tools performing these tasks in an easy way. This kind of tools should offer an abstract view of a MAS where errors can be detected in an easy and fast way. The paper proposes a tool that uses causality to guide the developer towards the root of the problem. To illustrate the role of causality, a brief example follows. There is a MAS in which an agent (denoted by Initiator) wants to search bibliographic references about some subject. To perform this task the Initiator communicate with another agent (denoted by Broker) using the FIPA Brokering interaction protocol, and delegate on the Broker to do the search task. The *proxy* message sent by Initiator agent, contains the following information: a referential expression denoting the target agents to which the Broker should forward the requested task and the communicative act to forward. After notify to Initiator that the petition is accepted, the broker agent forward again the search request, using the communicative act (*request*) and the target agent list, received both from Initiator. The target agent list contains two agents, denoted by Searcher1 and Searcher2, respectively, so Broker communicate with Searcher1 and Searcher2 using FIPA Request interaction protocol. Searcher1, communicate with agents BookSeller1, BookSeller2, and BookSeller3 representing each one some e-library, to perform the search, using the FIPA Request interaction protocol. Searcher2, is not able to understand the *request*, so it doesn't send some message. After, some interactions, Searcher1 receives only failure messages, which are forwarded to Broker, which forward again the failure to Initiator. The example described above is shown in figure 1.

After MAS execution, the developer realizes that Initiator's search returns failure, but the source of the error is not evident. Looking at the graph shown in figure 1, the developer discovers that Initiator's search has failed because Broker returns failure too. Looking at the bottom of the graph, it seems Broker behavior is correct, because all messages received are failures. Tracking back agents interaction, developer realize that all responses received by Searcher1, are failures too, and Searcher2 does not do anything. The MAS programmer, reviews agents code and finds which databases are been accessed by agents. Once databases are found, the programmer checks whether the bibliographic reference searched by Initiator agent, exists or not in e-libraries that are accessed by BookSeller1, BookSeller2 and BookSeller3. He discovers that the searched reference does not exist and realizes that the problem is in the target agent list elaborated by Initiator. Probably, it was caused by wrong information about the

environment, maybe provided or modified by some information source. Related with Searcher2, developer, after reviewing agent's code, realizes that this agent is not able to process a *request* message, so the other cause error is found too. In this example, causality graph is useful for developer to find error causes, but he has to inspect agents code, inspection that implies to consume more development time.

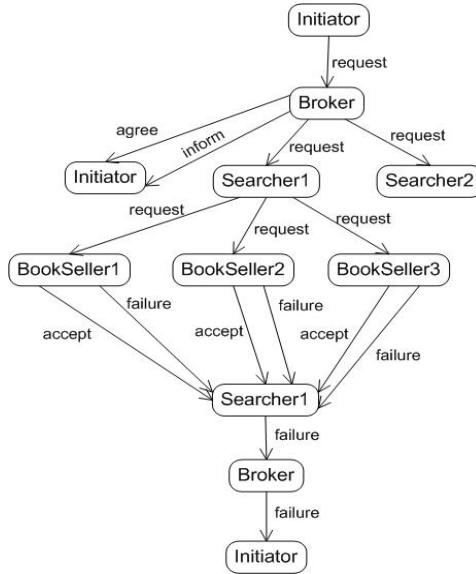


Fig. 1. Example in which agents search bibliographic references

The graph shown in figure 1, can help to developer to discover which agent has a wrong behaviour, respect to what is expected. But once the agent is found the developer can't know which is the internal agent state that cause the wrong behavior. We propose in this paper to add semantic information, extracted from agents interaction protocols, to the graph shown in figure 1 to increase MAS execution information and reduce, in this way, debug and validation time.

3 The INGENIAS Methodology

INGENIAS defines process, methods and tools for developing MAS, from the more abstract analysis, with use cases and system goals, to implementation of code. In order to cope with the complexity of MAS specification, each MAS is considered from five complementary viewpoints [5]:

- Agent viewpoint. This viewpoint is concerned with the functionality of each agent (i.e. responsibilities and capabilities). The behaviour of the agent is defined through three components: (1) the mental state, (2) the mental state manager, which provides for operations on mental entities, and (3) the mental state processor

(which determines how the mental state evolves and it is described in terms of rules, planning, etc.).

- **Organisation viewpoint.** The organisation describes the framework where agents, resources, tasks and goals coexist. This viewpoint is defined by the organisation structure (with group decompositions), social relationships (power relationships mainly), and functionality. The functionality of the organization is defined by its purpose and tasks.
- **Task/Goal viewpoint.** It considers the relationships among goals and tasks or plans, and describes the consequences of performing a task, and why it should be performed (i.e. it justifies the execution of tasks as a way to satisfy goals). For each task, it determines which elements are required and what outputs are expected. To identify which goals are influenced by a task execution, the methodology defines satisfaction and failure relationships.
- **Interaction viewpoint.** The interaction viewpoint addresses the exchange of information between agents, or between agents and human users. The definition of an interaction requires the identification of: actors in the interaction, interaction specification, context of the interaction, and nature of the interaction.
- **Environment viewpoint.** It defines the entities with which the MAS interacts. These entities are resources (elements required by tasks that do not provide a concrete API, like CPU, file descriptors, or memory), applications (they offer some local or remote API), or other agents (from other organisations).

In the following section is explained how the INGENIAS methodology can be used to build the semantic causality graph.

4 Debugging an Example INGENIAS MAS Model

This section shows an example of how semantic causality graphs can aid developer to get more information about a running MAS under development. For this purpose, this section presents an example MAS specification modelled with INGENIAS. Later, it is shown how debug and validation tasks are performed using semantic causality graphs. Taking account that the goal is extending causality graphs that appear in figure 1, to show how agents interactions affect to agent internal state, different INGENIAS viewpoints has to be browsed. In the next section it is described the MAS specification used and which INGENIAS meta-models are used.

4.1 MAS Example Specification Description

The system intends to provide a ticket selling service where a user representative can contact each cinema and proceed with the transaction. The case study considers a user connecting to the system and instructing a representative (User assistant) to obtain a cinema ticket. The assistant contacts another representative (buyer) so that this representative obtains the ticket. The representative contacts different cinemas representatives (seller) until an adequate cinema is located (adequate means there are free seats and the price is good enough). Once obtained, the ticket is delivered, through the different representatives, to the user.

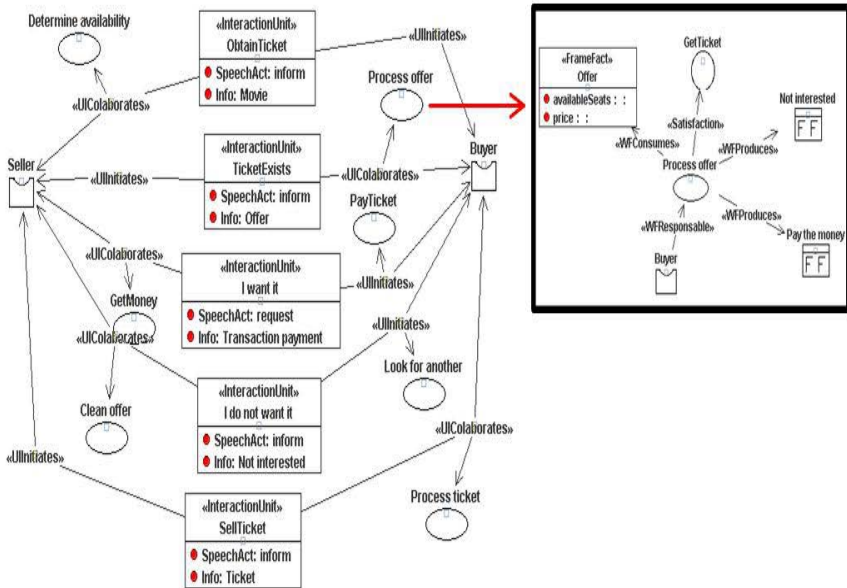


Fig. 2. Definition of interaction protocol to buy a cinema ticket, in which interaction semantic information is added. On the right the task *Process offer* is defined too.

Taking into account that semantic information associated to agents interactions is the main concern, the INGENIAS Interaction viewpoint and Task/Goal viewpoint is consulted. The first one gives information about which tasks are executed by one agent when receives a message. Focusing on these tasks is necessary because modifications performed by some task when a message is received will influence the next message. Related to the interaction viewpoint, the Task/Goal viewpoint informs of the modifications to perform over an agent mental state, those executed by enacted tasks. In the system described above there are two interactions. The first one between User assistant agent and Buyer agent, in which User assistant instructs Buyer to get a cinema ticket. The second one between Buyer and Seller agent negotiates the buy of the cinema ticket required. In figure 2, it is shown a diagram from Interaction viewpoint for specifying the interaction protocol used by buyer agent and seller agent, to negotiate cinema ticket required. On the right, it appears a detailed specification of the task *Process offer* from Task/Goal viewpoint. In the interaction protocol specification, messages sent by agents are represented through *Interaction units*, each one containing one speech act or performative. When a message is sent by one agent (or role) this is specified by means of relation *UIInitiates*. When a message is received, it is represented by means of relation *UIColaborates*. Semantic information is defined associating one task to a sent or received message event. For each task into Task/Goal viewpoint it is identified entities from agent mental state being consumed/produced, and environment entities being used. Due to space limitations, only one task is shown. As the reader can see, the task defined in figure 2, satisfies the goal *GetTicket* (relation *Satisfaction*) and the role *Buyer* is responsible of execute the task (relation *WFResponsible*). In addition, the task will produce the fact *Pay the*

money if the buyer agent is interested in the ticket offered, and will produce the fact *Not interested* if isn't interested. Taking account this semantic information, in the next section it is shown an example of semantic causality graph.

4.2 Validating the Example Using Semantic Causality Graphs

There is a system in which there is one user and his user assistant agent, one Buyer, and two sellers (Seller1 and Seller2) representing one cinema each one. Let's suppose that the user instructs the User assistant to obtain a cinema ticket. The User assistant starts an interaction with Buyer agent, to get the cinema ticket. Later on, Buyer agent starts another interaction with Seller1 to buy the required ticket. The ticket offered is refused by Buyer so it starts a new interaction with Seller2 and again the offer is refused. The developer, looking at the resultant execution, begins to think about the reasons for the refused offers. The graph shown in figure 3 allows the developer to know the reasons that lead to Buyer to refuse tickets offered, getting semantic information from the running MAS (shown inside black rectangles in figure 3), and without necessity to inspect agents code. By means of semantic information, the developer can realize that the tickets are refused due to the offer sent by each seller to Buyer agent (see facts *Offer* produced by Seller1 and Seller2 in figure 3). In both cases the price of the offer is very high, as is shown in the attribute *Price* of offers from Seller1 and Seller2. To solve the problem, by means of semantic information added, the developer realizes that the code of the task *Determine availability*, that is in charge of elaborate the refused offer, should be reviewed. Please, notice that compared with tools provided by other platforms to develop BDI agents, our proposal constitute a more intuitive tool. In this way, in the JASON platform [7], agent's code could be debugged by means of information logged to the user console. Keep the trace of events occurred in the MAS, by means of this tool, is more difficult than using our semantic graph.

5 Related Works

There are previous works on using causality and visual debugging for MAS, and specially the later ones. Causality graphs are used extending them to capture causality among different MAS events in [3]. In that work, a developer defines agent concepts and causal relation among those pre-defined concepts is shown (e.g. a believe created as a consequence of a received message). The work [3] is complementary to the one presented, because the causality graph is at intra-agent level and not at inter-agent level.

An example of visualization tool to assist in testing and debugging MAS is the one described in [2]. Simple communication graphs are used there, i.e. nodes are agents and arcs exits between nodes if they exchanged any message. However, when these communication graphs get so complex to be visualized (i.e. many agents and many arcs), a simplification process based on clustering is used to group agents. Different criteria are used to decide on the grouping.

Keeping the attention on visualization tools for MAS, other interesting example is the one presented in [4]. In this case, the developer can use different kinds of visual

diagrams. For example, Gantt based diagrams show decomposition, dependency and ordering among agents tasks. Using 3D graphical representation, in [1] is proposed a tool to debug some MAS. This example follows a sequence diagram model and allow to filter events occurred in the MAS, by means of special queries, depending on user interests.

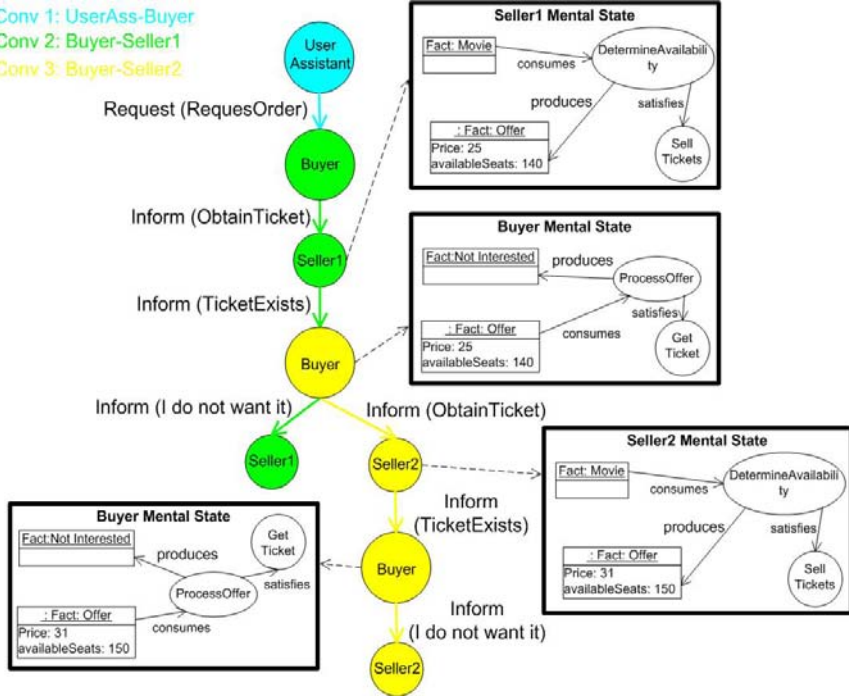


Fig. 3. Example in which agents search bibliographic references

6 Conclusions

This paper has proposed a visual representation of MAS behavior aimed to aid the developer in MAS debugging, verification and validation tasks. This visual representation allows a deeper and quicker understanding of the system behavior. The deeper understanding is provided by means of causality concept, which allows to know the cause message of a given message, and through semantic information, which shows agents mental state that leads to send a message to each agent.

We keep working on a generic algorithm for producing semantic graphs using as input INGENIAS models. Moreover semantic information can lead to program tools to perform automatic reasoning about MAS behaviour. Error causes could be automatically extracted as much as some suggestions to solve the problem. In addition we plan to develop different methods to simplify the graph, for example to show only both interaction and semantic information about one agent. This can be helpful in complex graphs with a high number of agents.

Another interesting line of work is to apply semantic graphs proposed in this paper to hybrid systems modeled using the INGENIAS methodology.

Acknowledgements. This work has been supported by the Spanish Council for Science and Education under grant TIN-2005-08501-C03-02 and the project "Methods and tools for multi-agent systems modelling" supported by Spanish Council for Science and Technology with grant TIN-2005-08501-C03-01.

References

1. S. Ilarri, J.L. Serrano, E. Mena, and R. Trillo. 3D monitoring of distributed multiagent systems. In International Conference on Web Information Systems and Technologies (WEBIST 07), Barcelona (Spain), pages 439_442. INSTICC Press, ISBN 978-972-8865-77-1, March 2007.
2. Juan M. Hernansaez Juan A. Botia and Antonio F. Gomez-Skarmeta. On the application of clustering techniques to support debugging large-scale multi-agent systems. In Programming Multi-Agent Systems Workshop at the AAMAS, Hakodate, Japan, 2006.
3. D. N. Lam and K. S. Barber. Comprehending agent software. In AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, pages 586_593, New York, NY, USA, 2005. ACM Press.
4. Divine T. Ndumu, Hyacinth S. Nwana, Lyndon C. Lee, and Jaron C. Collis. Visualising and debugging distributed multi-agent systems. In ACM Press, editor, AGENTS '99: Proceedings of the third annual conference on Autonomous Agents, pages 326_333, 1999.
5. J. Pavon, J. J. Gomez-Sanz, and R. Fuentes. Agent-Oriented Methodologies, chapter The INGENIAS Methodology and Tools, pages 236_276. Idea Group Publishing, 2005.
6. Guillermo Vigueras and Juan A. Botia. Tracking causality by visualization of multiagent interactions using causality graphs. In Programming Multi-Agent Systems Workshop at the AAMAS (to appear), Honolulu, Hawaii, 2007.
7. Bordini, R. H.; Dastani, M.; Dix, J. & Fallah-Seghrouchni, A. E. (ed.) Multi-Agent Programming: Languages, Platforms and Applications Multi-Agent Programming, Springer, 2005, 15

A Multiagent Framework to Animate Socially Intelligent Agents

Francisco Grimaldo, Miguel Lozano, and Fernando Barber

Computer Science Dept., University of Valencia
Dr. Moliner 50, 46100 Burjassot, Valencia
{francisco.grimaldo,miguel.lozano,.barber}@uv.es

Abstract. This paper presents a multiagent framework designed to animate groups of synthetic humans that properly balance task oriented and social behaviors. The work presented in this paper focuses on the BDI agents and the social model integrated to provide socially acceptable decisions. The social model provides rationality, to control the global coordination of the group, and sociability, to simulate relations (e.g. friends) and reciprocity between members. The multiagent based framework has been tested successfully in dynamic environments while simulating a virtual university bar, where several types of agents (groups of waiters and customers) can interact and finally display complex social behaviors (e.g. task passing, reciprocity, planned meetings).

Keywords: Artificial societies, social multiagent systems.

1 Introduction and Related Work

Socially intelligent agents are autonomous problem solvers that have to achieve their goals by interacting with other similarly autonomous entities [7]. Bearing this in mind, multiagent systems can be referred to as *societies of agents*. Within the field of behavioral animation for autonomous synthetic characters, this points to the design of an adequate framework to produce good quality animations for groups of virtual humans. When designing such agents, the main concern has normally been with the decision-making mechanism, as it is the responsible for the actions that will be finally animated. Virtual humans normally play a role (e.g. a virtual guide, seller, client...), thus, it is very important not to avoid the social aspects involved. The notion of socially acceptable decisions has long been of interest in human societies and also in the multiagent community. Traditionally, designers have sought to make their agents rational so that, they can “do the right thing” efficiently (i.e. the shorter the plan the better). Since agents operate in dynamic resource bounded contexts, obstructions rapidly appear when characters act and compete for the use of shared resources (i.e. objects in a virtual environment). Therefore, self interested agents (i.e. agents devoted to accomplish a set of goals) easily come into conflicts even though their goals are compatible, producing low quality animations. Many researches have tried to achieve social skills through multiagent coordination. For example in Generalized Partial Global Planning (GPGP) [4], agents merge the meta-plans describing their operational procedures and figure out the better action in order to maximize the global utility.

Another example is Multiagent Planning Language (MAPL) [3], which assigns the control over each resource to a unique agent and uses speech-acts to synchronize planned tasks. Collaboration is supported in the RETSINA system [5] thanks to the use of communicative acts that synchronize tasks and occasionally manage conflicts. Team formation and task coordination were applied to heuristic planning based characters in [6] to adapt better to the dynamism of shared environments. MAS-SOC [1] is a similar system in the literature that aims at creating a platform for multiagent based social simulations. In this context, work is ongoing in order to incorporate social-reasoning mechanisms based on *exchange values* [9]. All these approaches focus on improving the global efficiency of a multiagent society, however, an excess of coordination can produce unethical behaviors which are less interesting to animate. Instead, egalitarian societies of agents are more interesting when simulating groups of virtual humans. In these simulations, autonomous characters should display social behaviors (e.g. interchange information with their partners or grouping and chatting with their friends) and task oriented behaviors, in accordance with the role defined (e.g. a waiter serving at a virtual bar). This kind of socially intelligent animation agents is required in many complex simulation environments: military/civil simulations, virtual cities (e.g. social pedestrians), games, etc. The multiagent framework presented here is directed at designing socially intelligent agents able to balance properly their rationality and sociability, as it is necessary to finally display high quality behavioral animations.

2 Social Multiagent Framework

The social multiagent framework presented in figure 1 has been developed over Jason [2], which allows the definition of BDI agents using an extended version of AgentSpeak(L) [8]. The animation system (virtual characters, motion tables, etc) is located at the 3D engine, which can run separately. The agent decision-making is defined in the *Agent Specification File*. This file contains the initial beliefs as well as the set of plans that make up the agent's finite state machine. The *Task Library* contains the set of plans that sequence the actions needed to animate a task and achieve a particular goal. Here, modularity is guaranteed since the *Task library* can be changed depending on the environment and the roles being simulated.

As stated above, only rational behaviors are not enough to simulate agent societies, therefore, we have added to the architecture a *Social library* to manage different types of social situations. This library is based on an auction model that uses the social welfare concepts to avoid conflicts and allow the agents to behave in a coordinated way. The auction model is a numerical model, in contrast with the logical model of the task library. Both libraries do not necessarily propose the same behaviour for the agent. In those cases, the conflict is resolved by means of a *sociability factor*, which allows choosing one of the behaviours. The *Social library* also incorporates a reciprocity model in order to promote social interactions between the members of a society. The *Conversational library* contains the set of plans that handle the animation of the interactions between agents (e.g. planned meetings, chats between friends...). Finally, the *Semantic Layer* is in charge of perceiving the state of the world and executing the actions requested by the agents, while ensuring the consistency of the *World Model*.

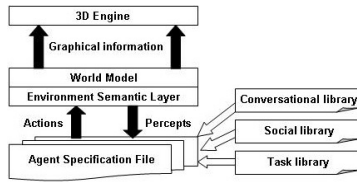


Fig. 1. Social multiagent framework

3 Social Library

The simulation of worlds inhabited by interactive virtual actors normally involves facing a set of problems related to the use of shared limited resources and the need to animate pure social behaviors. Both types of problems are managed by the *Social library* by using a Multiagent Resource Allocation approach [7]. This library allows the auctioning of tasks by any agent in order to reallocate them so that the global social welfare can be increased. Tasks are exchanged between agents using a first-price sealed-bid (FPSB) auction model where the agents express their preferences using *performance* and *social utility functions*.

The performance utility function $U_{perf}^i(<i \leftarrow t>)$ of a bidder agent i reflects the efficiency achieved when he performs the task t . There can be many reasons for an agent to be more efficient: he might perform the task faster than others because of his know-how or it might be using a resource that allows several tasks to be performed simultaneously (e.g. a coffee machine in a virtual bar can be used by a waiter to make more than one coffee at the same time). The utility function has to favor the performance of the agents, but high performances can also be unrealistic for the animation of artificial human societies. For example, if all agents work as much as they can, they will display unethical or robotic behaviors. Furthermore, agents should also show pure social behaviors to animate the normal relations between the members of a society.

Whereas the performance utility function modeled the interest of an agent to exchange a task from an efficiency point of view, we introduce two additional social utilities to represent the social interest in exchanging a task. The aim of social utilities is to promote task allocations that lead the agents to perform social interactions with other agents (e.g. planned meetings with their friends). Negotiation of long sequences of actions is not very interesting for interactive characters, as plans will probably be broken due to the dynamism of the environment and to other unpredictable events. Thus, we define the following social utility functions:

- Internal social utility ($U_{int}^i(<i \leftarrow t, j \leftarrow t_{next}>)$): is the utility that a bidder agent i assigns to a situation where i commits to do the auctioned task t so that the auctioneer agent j can execute his next task t_{next} .
- External social utility ($U_{ext}^i(<j \leftarrow t>)$): is the utility that a bidder agent i assigns to a situation where the auctioneer agent j executes the auctioned task t while i continues his current action.

The winner determination problem has two possible candidates coming from performance and sociability. In equation 1 the welfare of a society is related to

performance, hence, the winner of an auction will be the agent that bid the maximum performance utility. On the other hand, equation 2 defines the social winner based on the maximum social utility received to pass the task to a bidder ($U_{int}^*(t)$) and the maximum social utility given by all bidders on the situation where the task is not exchanged but performed by the auctioneer j ($U_{ext}^*(t)$).

$$winner_{perf}(t) = \left\{ k \in Agents \mid U_{perf}^k(t) = \max_{i \in Agents} \left\{ U_{perf}^i(< i \leftarrow t >) \right\} \right\} \quad (1)$$

$$winner_{soc}(t) = \begin{cases} j & U_{ext}^*(t) \geq U_{int}^*(t) \\ i & U_{ext}^*(t) < U_{int}^*(t) \text{ and } U_{int}^i(t) = U_{int}^*(t) \end{cases} \quad (2)$$

To balance task exchange, social utilities are weighted with a reciprocity matrix (see equations 3 and 4). We define the reciprocity factor w_{ij} for two agents i and j , as the ratio between the number of favors (i.e. tasks) that j has made to i (see equation 5).

$$U_{int}^*(t) = \max_{i \in Agents} \left\{ U_{int}^i(< i \leftarrow t, j \leftarrow t_{next} >) * w_{ji} \right\} \quad (3)$$

$$U_{ext}^*(t) = \max_{i \in Agents} \left\{ U_{ext}^i(< j \leftarrow t >) * w_{ij} \right\} \quad (4)$$

$$w_{ij} = Favors_{ji} / Favors_{ij} \quad (5)$$

At this point, agents can decide whether to adopt this kind of social allocations or to be only rational as explained previously. They choose between them in accordance with their *Sociability* factor, which is the probability to select the social winner instead of the rational winner. *Sociability* can be adjusted in the range [0,1] to model intermediate behaviors between efficiency and total reciprocity. This can provide great flexibility when animating characters, since *Sociability* can be dynamically changed thus producing different behaviors depending on the world state.

4 Application Example

In order to test the presented social multiagent framework, we have created a virtual university bar where waiters take orders placed by customers (see figure 2a). The typical locations in a bar (e.g. a juice machine) behave like resources that have an associated time of use to supply their products (e.g. 7 seconds to make an orange juice) and they can only be occupied by one agent at a time.

The waiters are governed by the finite state machine¹ shown in figure 2b, where orders are served basically in two steps: first, using the corresponding resource (e.g. the grill to produce a sandwich) and second, giving the product to the customer. Tasks are always auctioned before their execution in order to find good social allocations. Equations 6 and 7 define the utility values returned by the performance utility function for these tasks. This function aims at maximizing the number of parallel tasks being performed and represents the waiters' willingness to serve orders as fast

¹ Specified by means of plans in Jason's extended version of AgentSpeak(L).

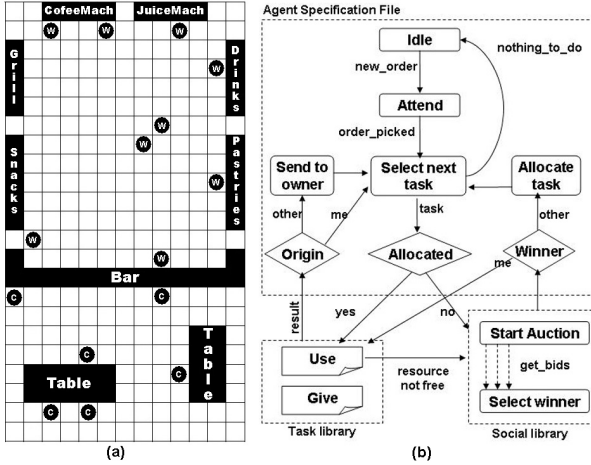


Fig. 2. (a) Virtual university bar environment, (b) Waiter specification

as possible. Social behaviors defined for a waiter are oriented to animate chats between his partners. Therefore, waiters implement the internal and external social utility functions detailed in equations 8 and 9, where $Near$ computes the distance between the agents while they are executing a pair of tasks. These functions evaluate social interest as the chance to meet a partner in the near future (i.e. a planned meeting).

$$U_{perf}^i(i \leftarrow 'Use') = \begin{cases} 1 & \text{if } [(i = \text{Auctioneer}) \text{ and } (\text{IsFree}(\text{Resource})) \text{ or} \\ & [\text{IsUsing}(i, \text{Resource}) \text{ and not}(\text{IsComplete}(\text{Resource}))]. \\ 0 & \text{Otherwise} \end{cases} \quad (6)$$

$$U_{perf}^i(i \leftarrow 'Give') = \begin{cases} 1 & \text{if } [(i = \text{Auctioneer}) \text{ and } (\text{nextAction} = \text{NULL})] \text{ or} \\ & [\text{currentTask} = 'Give' \text{ and not}(\text{handsBusy} < 2)]. \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

$$U_{int}^i(< i \leftarrow t, j \leftarrow t_{next} >) = \begin{cases} 1 & \text{if } \text{IsFriend}(i, j) \text{ and } \text{Near}(t, t_{next}) \text{ and} \\ & \text{ExecTime}(t_{next}) > \text{RemainTime}(\text{currentTask}). \\ 0 & \text{Otherwise} \end{cases} \quad (8)$$

$$U_{ext}^i(j \leftarrow t) = \begin{cases} 1 & \text{if } \text{IsFriend}(i, j) \text{ and } \text{Near}(\text{currentTask}, t) \\ 0 & \text{Otherwise} \end{cases} \quad (9)$$

On the other hand, customers place orders and consume them when served. Now, we are not interested in improving customer performance but in animating interactions between the members of a social class. Thus, we have implemented three classes of customers that use auctions to solve the problem of *where to sit*. A finite

state machine similar to that in figure 2b governs the actuation of customers. Depending on his or her sociability factor, a customer can randomly choose a chair or start an auction to decide where to sit and consume. This auction is received by all customers in the bar, which use the external social utility function defined in equation 10 to promote meetings with others of the same class. We define the performance and the internal social utility functions as 0 since task passing is not possible in this case (i.e. no-one can sit instead of another customer). Finally, when a social meeting emerges, both waiters and customers use the plans in the *Conversational Library* to sequence the speech-acts needed to animate commitments, greetings or simple conversations.

$$U_{ext}^i(j \leftarrow 'Sit') = \begin{cases} 1 & \text{if IsSameClass}(i, j) \text{ and IsConsuming}(i, \text{auctionedTable}) \\ 0 & \text{Otherwise} \end{cases}. \quad (10)$$

5 Results

To illustrate the effects of the social techniques previously applied we have animated the virtual university bar example with 10 waiters serving 100 customers both with different sociability factors (visit <http://www.uv.es/~agentes> to download some animation videos). To measure the efficiency of a group of waiters we use the ratio between the optimal simulation time and the real simulation time (see equation 11). *Throughput* is an indicator in the range $[0,1]$ that estimates how close a simulation is to the ideal situation in which the workload can be distributed among the agents and no collisions arise.

$$Throughput = \frac{N_{tasks} * \overline{T_{task}} / N_{agents}}{T_{sim}}. \quad (11)$$

Figure 3a shows the *Throughput* obtained by different types of waiters versus self-interested agents (i.e. agents with no social mechanisms included). Self-interested agents collide as they compete for the use of the shared resources and these collisions produce high waiting times as the number of agents grows. We can enhance this low performance with elitist agents (*Sociability* = 0) which exchange tasks with others that can carry them out in parallel thus reducing the waiting times for resources. Nevertheless, they produce unrealistic outcomes since they are continuously working if they have the chance, leaving aside their social relationships (e.g. chats between friends). The *Sociability* factor can be used to balance rationality and sociability. Therefore, the *Throughput* for the sort of animations we are pursuing should be placed somewhere in between elitist and fully reciprocal social agents (*Sociability*=1). On the other hand, figure 3b demonstrates that the higher the *Sociability* factor is, the larger the number of social meetings that will be performed by the customers when they sit at a table.

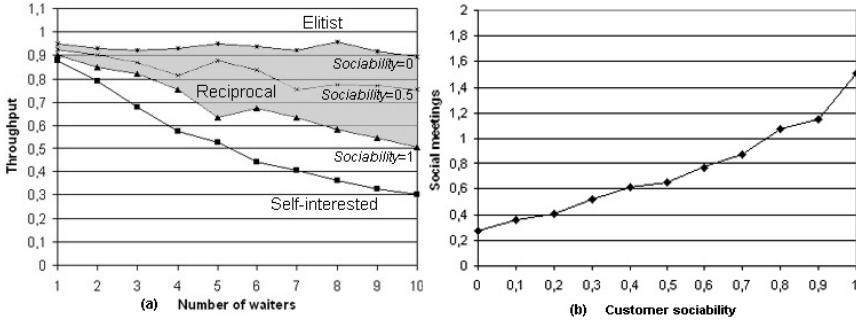


Fig. 3. (a) Waiter *Throughput*, (b) Customer social meetings

Finally, table 1 compares the amount of time devoted to execute each type of task in executions with 10 elitist waiters (*Sociability*=0) and 10 fully reciprocal social waiters (*Sociability*=1). The irregular values in the columns T_{use} and T_{give} on the left side of the table demonstrate how some agents have specialized in certain tasks. For instance, agents 2, 5, 9 and 10 spend most of their time giving products to the customers while agents 3 and 7 are mainly devoted to using the resources of the bar (e.g. coffee machine, etc). Although specialization is a desirable outcome in many multiagent systems, egalitarian human societies need also to balance the workload assigned to each agent. On the right side of the table, fully reciprocal social waiters achieve equilibrium between the time they are giving products and the time they are using the resources of the environment (see *Balance* column). A collateral effect of this equilibrium is the increase in the waiting times, since social agents will sometimes prefer to meet his friends in a resource than to reallocate the task (compare columns T_{wait}).

Table 1. Time distribution for 10 waiters in the bar (time values are in seconds)

Agent	<i>Sociability</i> = 0				<i>Sociability</i> = 1			
	T_{wait}	T_{use}	T_{give}	<i>Balance</i>	T_{wait}	T_{use}	T_{give}	<i>Balance</i>
1	0	32	19	-6	16	69	34	-2
2	3	4	26	-3	18	58	24	-2
3	14	52	1	28	41	45	16	0
4	3	16	28	-3	48	60	27	3
5	0	7	30	-16	34	58	12	-1
6	3	37	17	-1	48	64	14	-2
7	0	67	4	21	18	48	24	1
8	0	45	17	1	33	45	24	4
9	7	5	23	-11	46	36	21	0
10	1	6	41	-10	27	56	20	-1

6 Conclusions and Future Work

The animation of groups of intelligent characters is a current research topic with a great number of behavioral problems to be tackled. We aim at incorporating human

style social reasoning in character animation. Therefore, this paper presents a technique to properly balance social with task-oriented plans in order to produce realistic social animations. The multiagent animation framework presented allows for the definition of different types of social agents: from elitist agents (that only use their interactions to increase the global performance of the group) to fully reciprocal agents. These latter agents extend the theory of social welfare with a reciprocity model that allow the agents to control the emergence of unethical behaviors and promote social interactions among the members of a group.

Work is ongoing to provide the agents with mechanisms to self-regulate their *Sociability* factor depending on their social relations and on their previous intervention. Thus, agents will be able to dynamically adjust to the situation in order to stay within the boundaries of good quality animations at all times.

Acknowledgments. This work has been jointly supported by the Spanish MEC and European Commission FEDER funds under grants Consolider Ingenio-2010 CSD2006-00046 and TIN2006-15516-C04-04.

References

1. R.H. Bordini, A.C. da Rocha, J.F. Hübner, A.F. Moreira, F.Y. Okuyama and R. Vieira. A Social Simulation Platform Based on Agent-Oriented Programming. *Journal of Artificial Societies and Social Simulation*, vol.8, no.3, 2005.
2. R.H. Bordini and J.F. Hübner. Jason, 6th of March 2007. <http://jason.sourceforge.net/>.
3. M.Brenner. A multiagent planning language. In Proc. of ICAPS'03: Workshop on PDDL, 2003.
4. K.S. Decker and V.R. Lesser. Designing a family of coordination algorithms. *Readings in Agents*. Huhns and Singh editors, 1997.
5. J. A. Giampapa and K. Sycara. Team-Oriented Agent Coordination in the RETSINA Multi-Agent System. On Tech. Report CMU-RI-TR-02-34, Robotics Institute-Carnegie Mellon University, 2002.
6. F.Grimaldo, M.Lozano, and F.Barber. Integrating social skills in task-oriented 3D IVA. In Proc. of IVA'05: International Conference on Intelligent Virtual Agents. Springer, 2005.
7. L.M. Hogg and N.Jennings. Socially intelligent reasoning for autonomous agents. *IEEE Transactions on System Man and Cybernetics*, 31(5), 2001.
8. A.S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In Proc. of MAAMAW'96, LNAI 1038, pages 42-55, 1996.
9. M.Ribeiro, A.C. da Rocha and R.H. Bordini. A System of Exchange Values to Support Social Interactions in Artificial Societies. In Proc. Of AAMAS'03: Autonomous Agents and Multiagent Systems. ACM, 2003.

Context Aware Hybrid Agents on Automated Dynamic Environments

Juan F. de Paz, Sara Rodríguez, Juan M. Sánchez, Ana de Luis,
and Juan M. Corchado

Departamento Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
{fcofds,srg,adeluis,corchado}@usal.es, darksnaider@hotmail.com

Abstract. This paper presents an Ambient Intelligence based architecture model that uses intelligent hybrid agents with the ability to obtain automatic and real time information about the context using a set of technologies, such as radio frequency identification, wireless networks and wireless control devices, which can be implemented on a wide diversity of dynamic environments.

Keywords: Case-based planning; Context Aware; Hybrid systems.

1 Introduction

Agents and multi-agent systems (MAS) have become increasingly relevant for developing distributed and dynamic open systems, as well as the use of context aware technologies that allow those systems to obtain information about the environment. This paper is focused on describing the main characteristics of an Ambient Intelligence based architecture which integrates deliberative BDI (Believe, Desire, Intention) agents that employ Radio Frequency Identification, wireless networks, and automation devices to provide automatic and real time information about the environment, and allow the users to interact with their surroundings, controlling and managing physical services (i.e. heating, lights, switches, etc.). These context aware agents collaborate with hybrid agents that use Case-Based Reasoning (CBR) and Case-Based Planning (CBP) as reasoning mechanisms as a way to implement adaptive systems on automated dynamic environments.

A hybrid CBR-BDI agent [5] uses Case-Based Reasoning as a reasoning mechanism, which allows it to learn from initial knowledge, to interact autonomously with the environment as well as with users and other agents within the system, and to have a large capacity for adaptation to the needs of its surroundings. We shall refer to the hybrid CBR-BDI agents specialized in generating plans as hybrid CBP-BDI agents. BDI agents can be implemented by using different tools, such as Jadex [14]. Jadex agents deal with the concepts of beliefs, goals and plans; they are java objects that can be created and handled within the agent at execution time.

The architecture is founded on Ambient Intelligence (AmI) environments, characterized by their ubiquity, transparency and intelligence. Ambient Intelligence proposes a new way to interact between people and technology, where this last one is

adapted to individuals and their context, showing a vision where people are surrounded by intelligent interfaces merged in daily life objects [8], creating a computing-capable environment with intelligent communication and processing to the service of people by means of a simple, natural, and effortless human-system interaction for users [16], reason why to develop intelligent and intuitive systems and interfaces, capable to recognize and respond to the users necessities in a ubiquitous way [7], considering people in the centre of the development [17], and creating technologically complex environments in medical, domestic, academic, etc. fields. [20]. Agents in this context must be able to respond to events, take the initiative according to their goals, communicate with other agents, interact with users, and make use of past experiences to find the best plans to achieve goals.

Next, the main characteristics of the architecture are explained, focusing on the context aware agents and the technologies integrated to them.

2 Hybrid Reasoning and Planning Agents

All agents in this development are based on the BDI (Belief, Desire, Intention) deliberative architecture model [3], where the internal structure and capabilities of the agents are based on mental aptitudes, using beliefs, desires and intentions. We have implemented hybrid agents which integrates CBR systems [1] as a deliberative mechanism within BDI agents, facilitating learning and adaptation and providing a greater degree of autonomy than pure BDI architecture. CBR is a type of reasoning based on the use of past experiences [12] to solve new problems by adapting solutions that have been used to solve similar problems in the past, and learn from each new experience. To introduce a CBR motor into a deliberative BDI agent it is necessary to represent the cases used in a CBR system by means of beliefs, desires and intentions, and then implement a CBR cycle to process them.

The primary concept when working with CBR systems is the concept of case, which is described as a past experience composed of three elements: an initial state or problem description that is represented as a belief; a solution, that provides the sequence of actions carried out in order to solve the problem; and a final state that is represented as a set of goals. CBR manages cases (past experiences) to solve new problems. The way cases are managed is known as the CBR cycle, and consists of four sequential phases: retrieve, reuse, revise and retain. The retrieve phase starts when a new problem description is received. Similarity algorithms are applied in order to retrieve from the cases memory the cases with a problem description more similar to the current one. Once the most similar cases have been retrieved, the reuse phase begins, adapting the solutions for the retrieved cases to obtain the best solution for the current case. The revise phase consists of an expert revision of the solution proposed. Finally, the retain phase allows the system to learn from the experiences obtained in the three previous phases and consequently updates the cases memory. The retrieve and reuse phases are implemented through FYDPS [11] neural networks which allow the agent to recover similar cases from the cases memory and to adapt their solutions using supervised learning, in order to obtain a new optimal solution. The incorporation of these neural networks in the reasoning/planning mechanism reinforces the hybrid characteristics of the agent.

In a hybrid planning agent, the reasoning motor generates plans using past experiences and planning strategies, so the concept of Case Base Planning is obtained [9]. CBP consists of four sequential stages: retrieve stage to recover the most similar past experiences to the current one; reuse stage to combine the retrieved solutions in order to obtain a new optimal solution; revise stage to evaluate the obtained solution; and retain stage to learn from the new experience. Case-based planning (CBP) is the idea of planning as remembering [10]. CBP is a specialization of case-based reasoning (CBR) which is a problem solving methodology based on using a library of solutions for similar problems [10]. In CBP, the solution proposed to solve a given problem is a plan, so this solution is generated taking into account the plans applied to solve similar problems in the past. The problems and their corresponding plans are stored in a plans memory. Problem description (initial state) and solution (situation when final state is achieved) are represented as beliefs, the final state as a goal (or set of goals), and the sequences of actions as plans. The CBP cycle is implemented through goals and plans. When the goal corresponding to one of the stages is triggered, different plans (algorithms) can be executed concurrently to achieve the goal or objective. Each plan can trigger new sub-goals and, consequently, cause the execution of new plans.

Hybrid CBR-BDI and CBP-BDI agents are supported by BDI agents that manage a set of technologies to obtain all the context information required by the reasoning and planning mechanisms implemented, creating AmI-based systems that automatically adapt themselves to the changes in the environment.

3 Technologies for Context Awareness

The essential aspect in this work is the development of an AmI-based architecture as the core of multi-agent systems over automated and dynamic environments. Thus the use of technologies that provide the agents automatic and real time information of the context, and allow them to react upon it, is also important. Ambient Intelligence (AmI) provides an effective way to create systems with the ability to adapt themselves to the context and users necessities. The vision of AmI assumes seamless, unobtrusive, and often invisible but also controllable interactions between humans and technology. AmI provides new possibilities for resolving a wide range of problems and proposes a new way to interact between people and technology, where this last one is adapted to individuals and their context, showing a vision where people are surrounded by intelligent interfaces merged in daily life objects [8], creating a computing-capable environment with intelligent communication and processing to the service of people by means of a simple, natural, and effortless human-system interaction for users [16]. One of the most benefited segments of population with the appearance of AmI-based systems will be the elderly and people with disabilities, improving important aspects of their life, especially health care [8].

RFID technology is a wireless communications technology used to identify and receive information about humans, animals and objects on the move. An RFID system contains basically four components: tags, readers, antennas and software. Tags with no power system (i.e. batteries) integrated are called passive tags or “transponders”, these are much smaller and cheaper than active tags (power system included), but

have shorter read range. Figure 1 shows how these four elements combined enable the translation of information to a user-friendly format. The transponder is placed on the object itself (i.e. bracelet). As this object moves into the reader's capture area, the reader is activated and begins signalling via electromagnetic waves (radio frequency). The transponder subsequently transmits its unique ID information number to the reader, which transmit it to a device or a central computer where the information is processed and showed. This information is not restricted to the location of the object, and can include specific detailed information concerning the object itself. It is used in various sectors including health care [18]. The configuration presented in this paper consists of a transponder mounted on a bracelet worn on the users' wrist or ankle, and several sensors installed over protected zones, with an adjustable capture range up to 2 meters, and a central computer where all the information is processed and stored.

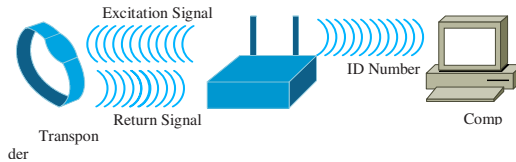


Fig. 1. Functioning of RFID technology

Wireless LAN's (Local Area Network) also known as Wi-Fi (Wireless Fidelity) networks, increase the mobility, flexibility and efficiency of the users, allowing programs, data and resources to be available no matter the physical location [19]. These networks can be used to replace or as an extension of wired LANs. They provide reduced infrastructure and low installation cost, and also give more mobility and flexibility by allowing people to stay connected to the network as they roam among covered areas, increasing efficiency by allowing data to be entered and accessed on site. New handheld devices facilitate the use of new interaction techniques, for instance, some systems focus on facilitating users with guidance or location systems [15] by means of their wireless devices. The architecture presented in this paper incorporates "lightweight" agents that can reside in mobile devices, such as cellular phones, PDA's, etc. [2], and therefore support wireless communication, which facilitates the portability to a wide range of devices.

Automation devices are successfully applied on schools, hospitals, homes, etc. [13]. There is a wide diversity of technologies that provide automation services, one of them is ZigBee, a low cost, low power consumption, two-way, wireless communication standard, developed by the ZigBee Alliance [21]. It is based on IEEE 802.15.4 protocol, and operates at 868/915MHz & 2.4GHz spectrum. ZigBee is designed to be embedded in consumer electronics, , PC peripherals, medical sensor applications, toys and games, and is intended for home, building and industrial automation purposes, addressing the needs of monitoring, control and sensory network applications [21]. ZigBee allows star, tree or mesh topologies, and devices can be configured to act as (Figure 2): network coordinator (control all devices); router/repeater (send/receive/resend data to/from coordinator or end devices); and end device (send/receive data to/from coordinator).

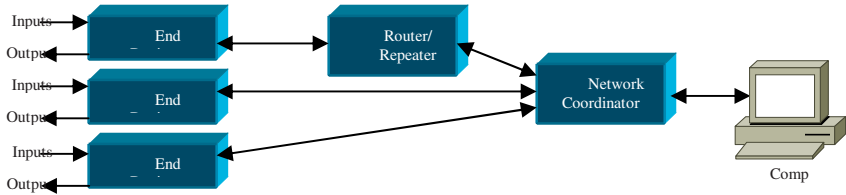


Fig. 2. ZigBee devices' configuration

4 Architecture Model

Reasoning and planning mechanism, and the technology described before are integrated into a generic multi-agent system prototype that can be implemented on different scenarios (Figure 3), for example in geriatric residences [6], with some changes according the users and project necessities. The architecture implements five different agents, which have specific roles (Figure 4):

- User Agent is a BDI agent that runs on mobile devices (PDA). It manages the users' personal data and behaviour (monitoring, location, daily tasks, and anomalies). The beliefs and goals used for every user depend on the plan or plans defined by the super-users. User Agent maintains continuous communication with the rest of the system agents, especially with the ScheduleUser Agent (through which the scheduled-users can communicate the result of their assigned tasks) and with the SuperUser Agent. The User Agent must ensure that all the actions indicated by the SuperUser are taken out, sending a copy of its memory base (goals and plans) to the Manager Agent in order to maintain backups.
- SuperUser Agent is a BDI agent that runs on mobile devices (PDA). It inserts new tasks into the Manager Agent to be processed by the CBR mechanism. It also needs to interact with the User Agents to impose new tasks and receive periodic reports, and with the ScheduleUser Agents to ascertain plans' evolution.
- ScheduleUser Agent is a hybrid CBP-BDI planner agent that runs on mobile devices (PDA). It schedules the users' daily activities obtaining dynamic plans depending on the tasks needed for each user. It manages scheduled-users profiles (preferences, habits, holidays, etc.), tasks, available time and resources. Every agent generates personalized plans depending on the scheduled-user profile.
- Manager Agent is a hybrid CBR-BDI Agent that runs on a Workstation. It plays two roles: the security role that monitors the users' location and physical building status (temperature, lights, alarms, etc.) trough a continuous communication with the Devices Agent; and the manager role that handle the databases and the tasks assignation. It must provide security for the users and ensure the tasks assignments are efficient. This assignation is carried out through a CBR reasoning engine, which is incorporated within the Manager Agent. When a new assignation of tasks

needs to be carried out, both past experiences and the needs of the current situation are recalled, allocating the respective and adequate task.

- Devices Agent is a BDI agent that runs on a Workstation. This agent controls all the hardware devices. It monitors the users' location (continuously obtaining/updating data from the RFID readers), interact with the ZigBee devices to receive information and control physical services (lights, door locks, etc.), and also check the status of the wireless devices connected to the system (PDA's). The information obtained is sent to the Manager Agent to be processed.

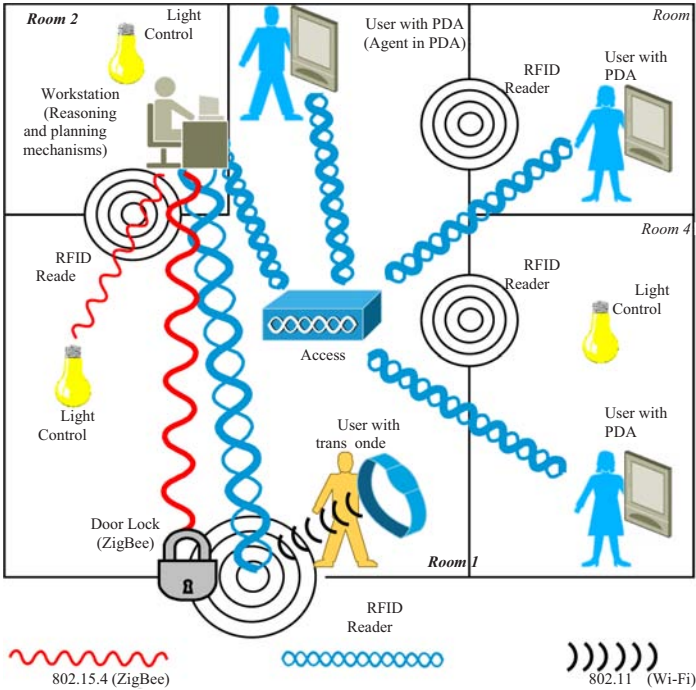


Fig. 3. Architecture applied on an automated environment

The essential hardware used is: Sokymat's Q5 chip 125KHz RFID wrist bands and computer interface readers for people identification and location monitoring; Silicon Laboratories' C8051 chip-based 2.4GHz development boards for physical services automation (heating, lights, door locks, alarms, etc.); mobile devices (PDA's) for interfaces and users' interaction; a Workstation where all the high demanding CPU tasks (planning and reasoning) are processed; and a basic Wi-Fi network for wireless communication between agents (in PDA's and Workstation). All the hardware is some way integrated to agents, providing them automatic and real time information about the environment that is processed by the reasoning and planning mechanisms to automate tasks and manage physical services.

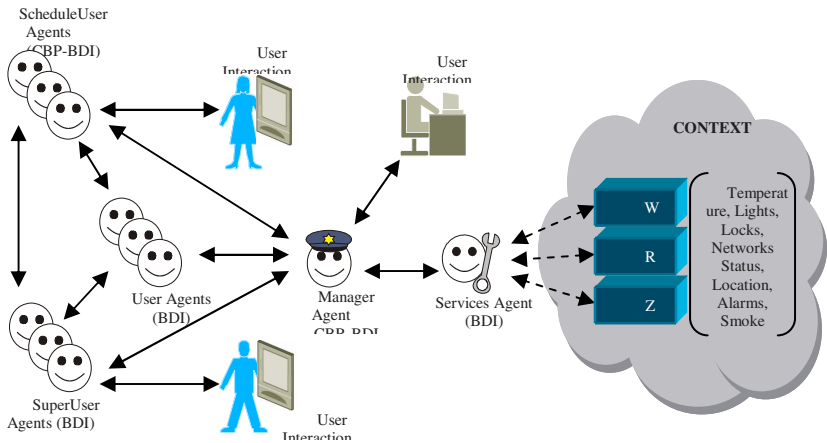


Fig. 4. Agents and technology in the architecture

5 Conclusions and Future Work

Deliberative BDI agents with reasoning and planning mechanisms and the use of technology to perceive the context, provide a robust, intelligent and flexible architecture that can be implemented in wide variety scenarios, such as hospitals, geriatric residences, schools, homes or any dynamic environment where is a need to manage tasks and automate services. In fact, a prototype system, based on this architecture has been successfully implemented into a geriatric residence [6], improving the security and the health care efficiency through monitoring and automating medical staff's work and patients' activities, facilitating the assignation of working shifts and reducing time spent on routine tasks.

The use of CBR systems helps the agents to solve problems, adapt to changes in the context, and identify new possible solutions. These new hybrid agent models supply better learning and adaptation than pure BDI model. In addition, RFID, Wi-Fi and ZigBee devices supply the agents valuable information about the environment, processed trough reasoning and planning mechanisms, to create a ubiquitous, non-invasive, high level interaction among users and the system.

However, it is necessary to continue developing and improving the AmI-based architecture presented, adding new capabilities and integrating more technologies to build more efficient and robust systems to automate services and daily tasks.

Acknowledgements. This work has been partially supported by the MCYT TIC2003-07369-C02-02 and the JCYL-2002-05 project SA104A05. Special thanks to Sokymat by the RFID technology provided and to Telefónica Móviles (Movistar) for the wireless devices donated.

References

1. Allen, J.F.: Towards a general theory of action and time. *Artificial Intelligence* Vol. 23 pp. 123-154. (1984)
2. Bohnenberger, T., Jacobs, O., Jameson, A.: DTP meets user requirements: Enhancements and studies of an intelligent shopping guide. *Proceedings of the Third International Conference on Pervasive Computing (PERVASIVE-05)*, Munich, Germany. (2005)
3. Bratman, M.E.: *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, M.A. (1987)
4. Corchado J.M., Bajo J., de Paz Y. and Tapia D. I.: Intelligent Environment for Monitoring Alzheimer Patients, *Agent Technology for Health Care. Decision Support Systems*. Elsevier Science. DOI 10.1016/j.dss.2007.04.008. In Press (2007)
5. Corchado, J.M., Laza, R.: Constructing Deliberative Agents with Case-based Reasoning Technology. *International Journal of Intelligent Systems*. Vol. 18 No.12 1227-1241 (2003)
6. Corchado J.M., Bajo J., de Paz Y. and Tapia D. I.: Intelligent Environment for Monitoring Alzheimer Patients, *Agent Technology for Health Care. Decision Support Systems*. Elsevier Science. DOI 10.1016/j.dss.2007.04.008. In Press (2007)
7. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: That's what friends are for. *Ambient Intelligence (AmI) and the IS in 2010. Innovations for an e-Society. Challenges for Technology Assessment*. Berlin, Germany. (2001)
8. Emiliani P.L., Stephanidis, C.: Universal access to ambient intelligence environments: opportunities and challenges for people with disabilities. *IBM Systems Journal*. (2005)
9. Glez-Bedia, M., Corchado, J.M.: A planning strategy based on variational calculus for deliberative agents. *Computing and Information Systems Journal*. Vol.10(1) 2-14. (2002)
10. Hammond, K.: *Case-Base Planning: Viewing Planning as a Memory Task*. Academic Press, New York. (1989)
11. Leung K.S., Jin H.D., Xu Z.B.: An expanding Self-organizing Neural Network for the Traveling Salesman Problem. *Neurocomputing*, vol. 62. (2004). pp 267-292.
12. Kolodner J.: *Case-based reasoning*. Morgan Kaufmann (1993).
13. Mainardi, E., Banzi, S., Bonfè, M. & Beghelli, S. (2005). A low-cost Home Automation System based on Power-Line Communication Links. *22nd International Symposium on Automation and Robotics in Construction ISARC 2005*. September 2005. Ferrara, Italy.
14. Pokahr, A., Braubach L., Lamersdorf, W.: Jadex: Implementing a BDI-Infrastructure for JADE Agents, in: *EXP - In Search of Innovation (Special Issue on JADE)*, Vol. 3, 76-85. Telecom Italia Lab, Turin, Italy, September (2003)
15. Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P., Zipf, A.: Crumpet: Creation of user-friendly mobile services personalised for tourism. In *Proceedings of 3G*. (2001)
16. Richter, K., Hellenschmidt, M.: Interacting with the Ambience: Multimodal Interaction and Ambient Intelligence. Position Paper to the W3C Workshop on Multimodal Interaction, 19-20 July. (2004)
17. Schmidt, A.: Interactive Context-Aware Systems Interacting with Ambient Intelligence. In G. Riva, F. Vatalaro, F. Davide & M. Alcañiz, *Ambient Intelligence*, IOS Press pp. 159-178. (2005)
18. Sokymat.: Sokymat. <http://www.sokymat.com>. (2006)
19. Sun Microsystems. (2000). *Applications for Mobile Information Devices. Helpful Hints for Application Developers and User Interface Designers using the Mobile Information Device Profile*. Sun Microsystems, Inc.
20. Susperregi, L., Maurtua, I., Tubío, C., Pérez M.A., Segovia, I., Sierra, B.: Una arquitectura multiagente para un Laboratorio de Inteligencia Ambiental en Fabricación. 1er. Taller de Desarrollo de Sistemas Multiagente (DESMA). Málaga, España. (2004)
21. ZigBee Standards Organization.: ZigBee Specification Document 053474r13. ZigBee Alliance. (2006)

Sensitive Stigmergic Agent Systems – A Hybrid Approach to Combinatorial Optimization

Camelia Chira, Camelia-M. Pinteau, and D. Dumitrescu

Department of Computer Science, Babes-Bolyai University, M. Kogalniceanu 1,
400084 Cluj-Napoca, Romania
{cchira, cmpinteau, ddumitr}@cs.ubbcluj.ro

Abstract. Systems composed of several interacting autonomous agents have a huge potential to efficiently address complex real-world problems. Agents communicate by directly exchanging information and knowledge about the environment. To cope with complex combinatorial problems, agents of the proposed model are endowed with stigmergic behaviour. Agents are able to indirectly communicate by producing and being influenced by pheromone trails. Each stigmergic agent has a certain level of sensitivity to the pheromone allowing various types of reactions to a changing environment. Resulting computational metaheuristic combines sensitive stigmergic behaviour and direct agent communication with the aim of better addressing combinatorial optimization NP-hard problems. The proposed model is tested for solving various instances of the Generalized Traveling Salesman Problem. Numerical experiments indicate the robustness and potential of the new metaheuristic.

Keywords: agent communication, stigmergy, sensitivity, multi-agent system, ant colony optimization.

1 Introduction

Combinatorial optimization problems arise in many and diverse domains including network design, scheduling, mathematical programming, algebra, games, language technology, computational linguistics and program optimization. Metaheuristics are powerful strategies to efficiently find high-quality near optimal solutions within reasonable running time for problems of realistic size and complexity [1].

A metaheuristic combining stigmergic behavior and agent direct communication is proposed. The *Sensitive Stigmergic Agent System (SSAS)* model involves several two-way interacting agents. Agents are endowed with a stigmergic behavior [2, 3] similar to that of Ant Colony Systems [4, 5]. This means that each agent is able to produce pheromone trails that can influence future decisions of other agents. SSAS agents are characterized by a certain level of sensitivity to the pheromone trail allowing various types of reactions to a changing environment [1].

SSAS agents can also communicate by directly exchanging messages using an Agent Communication Language – a behavior similar to that of multi-agent systems [6, 7, 8]. The information directly obtained from other agents is very important in the search process and can become critical in a dynamic environment (where the latest changes in the environment can be transmitted to other agents).

The SSAS model is tested for solving various instances of the (NP-hard) Generalized Traveling Salesman Problem. Numerical results indicate the potential of the proposed system.

The paper is organized as follows: Section 2 presents the two communication mechanisms (direct and stigmergic) of agents within SSAS; Section 3 introduces the concept of pheromone sensitivity for stigmergic agents; Section 4 presents the SSAS model using the concepts introduced in the previous two sections; the SSAS testing results are presented in the next two sections of the paper: Section 5 presents the SSAS hybrid metaheuristic for solving the Generalized Traveling Salesman Problem (GTSP) while Section 6 contains the numerical results and comparisons with other methods. Section 7 gives the conclusions of the paper and directions for future research.

2 Agent Communication in SSAS

Within the proposed model agents are able to communicate both directly and in a stigmergic manner using pheromone trails produced by agents.

2.1 Direct Communication

Communication in multi-agent systems [7] is necessary because agents need to exchange information or to request the performance of a task as they only have a partial view over their environment. Considering the complexity of the information resources exchanged, agents should communicate through an Agent Communication Language (ACL).

SSAS agents are able to exchange different types of messages in order to share knowledge and support direct interoperation. The content of the messages exchanged refers to environment characteristics and partial solutions obtained. The information about dynamic changes in the environment is of significant importance in the search process.

Furthermore, the introduced model inherits agent properties such as autonomy, reactivity, learning, mobility and pro-activeness used in multi-agent systems [6, 7, 8]. The agents that form the system have the ability to operate without human intervention, can cooperate to exchange information and can learn while acting and reacting in their environment.

2.2 Stigmergic Communication

Agents are endowed with the ability to produce pheromone trails that can influence future decisions of other agents within the system.

The idea of stigmergic agents was introduced in [1] where a system composed of stigmergic agents is outlined and illustrated by an example. Biology studies emphasize the remarkable solutions that many species managed to develop after millions of years of evolution. Self-organization [2] and indirect interactions between individuals make possible the identification of intelligent solutions to complex problems. These indirect interactions occur when one individual modifies the

environment and other individuals respond to the change at a later time. This process refers to the idea of *stigmergy* [3].

The stigmergic behavior of the SSAS agents is similar to that of the ants in the bio-inspired Ant Colony Optimization (ACO) metaheuristic [4, 5]. ACO simulates real ant behavior to find the minimum length path – associated to a problem solution – between the ant nest and the food source. Each ant deposits a substance called pheromone on the followed path. The decisions of the ants regarding the path to follow when arriving at an intersection are influenced by the amount of pheromone on the path. Stronger pheromone trails are preferred and the most promising paths receive a greater pheromone trail after some time.

3 Pheromone Sensitivity of SSAS Agents

A robust and flexible system can be obtained by considering that not all agents react in the same way to pheromone trails. Within the proposed model each agent is characterized by a *pheromone sensitivity level* denoted by PSL which is expressed by a real number in the unit interval [0, 1]. Extreme situations are:

- If $PSL = 0$ the agent completely ignores stigmergic information (the agent is ‘pheromone blind’);
- If $PSL = 1$ the agent has maximum pheromone sensitivity.

Small PSL values indicate that the agent will normally choose very high pheromone levels moves (as the agent has reduced pheromone sensitivity). These agents are more independent and can be considered environment explorers. They have the potential to autonomously discover new promising regions of the solution space. Therefore, search diversification can be sustained.

Agents with high PSL values will choose any pheromone marked move. Agents of this category are able to intensively exploit the promising search regions already identified. In this case the agent’s behavior emphasizes search intensification.

During their lifetime agents may improve their performance by learning. This process translates to modifications of the pheromone sensitivity. The PSL value can increase or decrease according to the search space topology encoded in the agent’s experience.

The current implementation of the proposed SSA model generates initial low PSL levels. Low sensitivity of agents to pheromone trails encourages a good initial exploration of the search space. The PSL value increases each generation allowing the exploitation of previous search results towards the end of the search process.

4 Sensitive Stigmergic Agent System Model

The SSAS model is initialized with a population of agents that have no knowledge of the environment characteristics. Each path followed by an agent is associated with a possible solution for a given problem. Each agent deposits pheromone on the

followed path and is able to communicate to the other agents in the system the knowledge it has about the environment after a full path is created or an intermediary solution is built.

The infrastructure evolves as the current agent that has to determine the shortest path is able to make decisions about which route to take at each point in a sensitive stigmergic manner. Agents with small PSL values will normally choose only paths with very high pheromone intensity or alternatively use the knowledge base of the system to make a decision. These agents can easily take into account ACL messages received from other agents. The information contained in the ACL message refers to environment characteristics and is specific to the problem that is being solved. On the other hand, agents with high PSL values are more sensitive to pheromone trails and easily influenced by stronger pheromone trails. However, this does not exclude the possibility of additionally using the information about the environment received from other agents.

After a set of agents determines a set of problem solutions, the proposed model allows the activation of another set of agents with the same objective but having some knowledge about the environment. The initial knowledge base of each agent refers to the information about the path previously discovered by each agent.

5 SSAS Model for Solving GTSP

The SSAS model is used for solving the *Generalized Traveling Salesman Problem (GTSP)*. GTSP is one of the possible generalizations of the well-known NP-hard problem TSP. Within GTSP the nodes of a complete undirected graph are partitioned into clusters. The objective is to find a minimum-cost tour passing through exactly one node from each cluster.

Agents deposit pheromone on the followed path. Unit evaporation takes place each cycle. This prevents unbounded intensity trail increasing. In order to stop agents visiting the same node in the same tour a *tabu* list is maintained.

The SSAS system is implemented using sensitive stigmergic agents with low initial PSL values. Initially, these sensitive-explorer agents autonomously discover new promising regions of the solution space to sustain search diversification. Each generation the PSL value increases with a constant value. Towards the end of the search agents intensively exploit the promising search regions already identified. The changing value of the PSL from low to high sensitivity ensures a meaningful balance between search exploration and exploitation in the problem solving process.

The SSAS model for solving GTSP works as follows:

Step 1

Initially the agents are placed randomly in the nodes of the graph. The PSL value of each agent is set to zero.

Step 2

Each SSAS agent moves to a new node with a probability based on the distance to that node and the amount of trail intensity on the connecting edge. The agent can

send an ACL message to the other agents containing the latter edge formed and its cost.

Step 3

The trail intensity is updated.

Step 4

The PSL value for each agent is increased using the expression:

$$PSL_t = PSL_1 * (1 + \ln t),$$

where t is the number of generations and PSL_1 represents the initial sensitivity level.

Step 5

Only agents that generate the best tour are allowed to globally update the virtual pheromone and the knowledge base. The global update rule is applied to the edges belonging to the best tour.

A run of the algorithm returns the shortest tour found. The next section presents the numerical results of this algorithm for a set of GTSP instances.

6 Numerical Experiments

The SSAS model for solving GTSP is tested for several instances of the problem considered.

The performance of the proposed SSAS model in solving GTSP is compared to the results of classical Ant Colony System (ACS) technique [9], the Nearest Neighbor (NN) algorithm, the GI^3 composite heuristic [10] and Random Key Genetic Algorithm [11].

The algorithm Ant Colony System for GTSP [9] is based on the ACS [4, 5] idea of simulating the behavior of a set of agents that cooperate to solve a problem by means of simple communications.

In Nearest Neighbor algorithm the rule is always to go next to the nearest as-yet-unvisited location. The corresponding tour traverses the nodes in the constructed order.

The composite heuristic GI^3 is composed of three phases: the construction of an initial partial solution, the insertion of a node from each non-visited node-subset, and a solution improvement phase [10].

The Random Key Genetic Algorithm combines a genetic algorithm with a local tour improvement heuristic. Solutions are encoded using random keys, which circumvent the feasibility problems encountered when using traditional GA encodings [11].

The data set of Padberg-Rinaldi city problems (TSP library [12]) is considered for numerical experiments. TSPLIB provides the optimal objective values (representing the length of the tour) for each problem. Comparative results obtained are presented in Table 1.

Table 1. Experimental results for solving GTSP using NN, GI³, ACS, Random Key GA and SSA

Problem	Optimum Value	NN	GI ³	ACS	Random Key GA	SSAS
16PR76	64925	76554	64925	64925	64925	64925
22PR107	27898	28017	27898	27904.4	27898	27898
22PR124	36605	38432	36762	36635.4	36605	36605
28PR136	42570	47216	43117	42593.4	42570	42570
29PR144	45886	46746	45886	46033	45886	45886
31PR152	51576	53369	51820	51683.2	51576	51576
46PR226	64007	68045	64007	64289.4	64007	64007
53PR264	29549	33552	29655	29825	29549	29549.2
60PR299	22615	27229	23119	23039.6	22631	22628.4
88PR439	60099	67428	62215	64017.6	60258	60188.4

The proposed SSAS model gives the optimal solution for 7 out of the 10 problems engaged in the numerical experiments. For two other problems, the solutions reported by SSAS are very close to the optimal value and better than those supplied by the other methods considered.

The test results emphasize that the proposed SSAS model gives better results than the standard ACS model and the NN algorithm. Furthermore, the results of SSAS are comparable and – for some of the considered problems better – than the GI³ algorithm and the Random Key Genetic Algorithm.

The numerical experiments and comparisons emphasize the potential of the proposed hybrid approach to address complex problems and facilitate further connections between multi-agent systems and nature inspired computing.

7 Conclusions and Further Work

Solving large complex problems – particularly those with a dynamic character – represents a challenging task. The idea explored by the paper refers to combining two different complementary approaches in order to address different facets of the problem.

A hybridization of Ant Colony Systems and Multi-Agent Systems at the system conceptual level is considered. The components of a multi-agent system are endowed with a supplementary capacity – the ability of communication by environmental changes. Agents adopt a stigmergic behavior (being able to produce pheromone trails) to identify problem solutions and use direct communication to share knowledge about the environment. This approach results in a new metaheuristic called SSAS (Sensitive Stigmergic Agent System) able to address problems that involve very complex search spaces (e.g. a graph, a tree, a complex network) for which solutions are incrementally built by agents. Numerical experiments indicate the effectiveness and the potential of the proposed SSAS technique.

Intelligent problem solutions can naturally emerge due to agent communication, autonomy and different levels of sensitivity to pheromone trails. The proposed SSAS

metaheuristic can be viewed as an approach to use multi-agent systems for solving NP-hard combinatorial optimization problems.

Future research directions concern the use of agents with sensitive stigmergy for solving real-world problems in non-stationary environments. Stigmergic agents can share information concerning dynamic changes in the environment (e.g. node or edge removing in a dynamic graph, cost modification of an edge, introduction of new nodes or new edges) improving the quality of the search process. The SSAS approach can be useful for addressing large problems concerning vehicle routing, communication in mobile systems or dynamic location.

References

1. Chira, C., Pintea, C.-M., Dumitrescu, D.: "Stigmergic Agents for Solving NP-difficult Problems", Proceedings of Bio-Inspired Computing: Theory and Applications Conference, Evolutionary Computing volume, Wuhan, China (2006) 63-69
2. Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: "Self organization in biological systems"; Princeton Univ. Press (2001)
3. Grasse, P.-P.: "La Reconstruction du Nid et Les Coordinations Interindividuelles Chez *Bellicositermes Natalensis* et *Cubitermes* sp. La Thorie de la Stigmergie: Essai d'interpretation du Comportement des Termites Constructeurs", *Insect Soc.*, 6 (1959) 41-80
4. Dorigo, M., Di Caro, G., Gambardella, L.M.: "Ant algorithms for discrete optimization", *Artificial Life*, 5 (1999) 137-172
5. Dorigo M., Blum, C.: "Ant Colony Optimization Theory: A Survey", *Theoretical Computer Science*, 344 2-3 (2005) 243-278
6. Nwana, H.S.: "Software Agents: An Overview", *Knowledge Engineering Review*, 11 (1996) 1-40
7. Jennings, N. R.: "An agent-based approach for building complex software systems", *Comms. of the ACM*, 44 4 (2001) 35-41
8. Wooldridge, M., Dunne, P. E.: "The Complexity of Agent Design Problems: Determinism and History Dependence", *Annals of Mathematics and Artificial Intelligence*, 45 3-4 (2005) 343-371
9. Pintea, C.-M., Pop, C. P., Chira, C.: "The Generalized Traveling Salesman Problem solved with Ant Algorithms", *Journal of Universal Computer Science*, Graz, Springer-Verlag, in press (2007).
10. Renaud, J., Boctor, F.F.: "An efficient composite heuristic for the Symmetric Generalized Traveling Salesman Problem", *European Journal of Operational Research*, 108 (1998) 571-584
11. Snyder, L.V., Daskin, M.S.: "A Random-Key Genetic Algorithm for the Generalized Traveling Salesman Problem", *European Journal of Operational Research* (2006), 38-53
12. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

Agent-Based Social Modeling and Simulation with Fuzzy Sets

Samer Hassan, Luis Garmendia, and Juan Pavó

Universidad Complutense Madrid, Facultad de Informática, Ciudad Universitaria s/n,
28040 Madrid
{samer, lgarmend, jpavon}@fdi.ucm.es

Abstract. Simple models of agents have been used to study emergent behaviour in social systems. However, when considering the evolution of values in human societies, there is a need for the consideration of more complex agent models, which take into account uncertainty in human thinking. This characteristic can be addressed by using fuzzy logic in the specification of the attributes that describe agents representing individuals, the functions that model the evolution of individual change of mind, the relationships among individuals in a social network, the inheritance, and the similarity between individuals. The introduction of fuzzy logic in the models has improved the results of the simulation.

Keywords: Agent-Based Social Simulation, Agent Modeling, Fuzzy Agents, Fuzzy Logic, Intelligent Agents.

1 Introduction

Agent based social simulation (ABSS) is applied as a tool for the study of the evolution of social systems. Although this has proved successful for a considerable number of cases, simple agent models as those normally used with existing tools are not sufficient to deal with the analysis of the evolution of values in human societies. Taking, for instance, sociological surveys, such as the European Value Survey, which is run periodically, we can see that, although there is an effort to categorize the possible answers to questions such as “Do you trust political parties?”, there is always a certain degree of uncertainty that has to be considered. This is even more evident when considering the evolution of values in individuals. This evolution is a consequence of multiple factors, which are strongly intertwined.

Fuzzy logic can be helpful to handle approximate or uncertainty knowledge. Here we propose to study how fuzzy logic can be useful for ABSS modeling. We analyze the different aspects that can be *fuzzified* from an ABSS model, so it gets better adapted to reality, and more specifically to facilitate the study of the evolution of values in human societies. In concrete, this work has been applied to model and simulate the evolution of religiosity in European societies, based on a Sociology work [1]. Although this has been implemented on RePast [7], a well known ABSS tool, some additions have been made to the existing library in order to be able to model and operate with fuzzy logic attributes, and relationships.

Section 2 reviews concepts about social systems simulation, and introduces the system under study, which is taken for experimentation ideas. Section 3 introduces fuzzy logic concepts related to this environment. Section 4 presents each part of the model that is fuzzified. Finally, Section 5 summarizes the main results and contributions of this work and issues for improving this framework.

2 Modeling and Simulation of Social Systems

Social phenomena are extremely complicated and unpredictable, since they involve complex interaction and mutual interdependence networks. A social system consists of a collection of individuals that interact among them, evolving autonomously and motivated by their own beliefs and personal goals, and the circumstances of their social environment.

A multi-agent system (MAS) consists of a set of autonomous software entities (the agents) that interact among them and with their environment. Autonomy means that agents are active entities that can take their own decisions. In this sense, the agent paradigm assimilates quite well to the individual in a social system, so it can be used to simulate them, exploring the complexity of social dynamics. With this perspective, agent-based simulation tools have been developed in the last years to explore the complexity of social dynamics. An agent-based simulation executes several agents, which can be of different types, in an observable environment where agents' behaviour can be monitored. Observations on agents can assist in the analysis of the collective behaviour and trends of system evolution. This provides a platform for empirical studies of social systems. As simulation is performed in a controlled environment, on one or several processors, this kind of tools allows the implementation of experiments and studies that would not be feasible otherwise.

There are, however, some limitations when trying to simulate real social systems. The main issue is that the individual, with regard to a software agent, is by itself a complex system, whose behaviour is unpredictable and less determined than for an agent, whose behaviour and perception capabilities can be designed with relative simplicity. Moreover, it is not possible in practice to consider the simulation of countless nuances that can be found in a real social system with respect to agent interaction, characterization of the environment, etc. For this reason, it is impractical to intend the simulation of a social system in all dimensions. On the other hand, we should and can limit to simulate concrete social processes in a systemic and interactive context. Therefore, the simulation of social systems should be considered in terms of focus on a concrete process.

In spite of these limitations, the agent paradigm offers many advantages to express the nature and peculiarities of social phenomena. However, existing agent based simulation tools promote the use of rather simple agent models, basically as a kind of cellular automata [11]. This may be valid to study emergent behaviour that results from deterministic behaviour of agents. But, when considering the evolution of complex mental entities, such as human beliefs and values, that approach is rather limited.

As an example of a system that requires further considerations on agent modeling, consider some sociological analysis derived from the European Value Survey and of the World Value Survey (for instance, [4]). In these surveys there are many questions

about the degree of happiness, satisfaction in different aspects of life, or trust in several institutions. Although there is some kind of categorization for the possible answers, such as “Very much” “Partially”, etc., there is always some degree of imprecision hardly to model with discrete categories. Even more, when the individual is evolving to different positions, some of these values get even more undefined. In order to take into account this kind of issues, fuzzy logic has been applied to model different aspects of the multi-agent system: agent attributes, functions for similarity and evolution of agent behaviour, relationships among agents in social networks, and inheritance.

3 Fuzzy Logic in This Context

Fuzzy logic is useful in vague and uncertain environments [12], as it is the case in the study of human societies. For its application in the modelling of ABSS, we have counted with the help of a sociology expert, which has been consulted repeatedly along the process.

Given a universe of discourse U , a fuzzy set $\mu: U \rightarrow [0,1]$ on the set U gives a membership degree to every element of U in the interval $[0, 1]$.

A binary operation $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$ is a *t-norm* [2][9], if it satisfies the following axioms:

1. $T(1, x) = x$
2. $T(x, y) = T(y, x)$
3. $T(x, T(y, z)) = T(T(x, y), z)$
4. If $x \leq x'$ and $y \leq y'$ then $T(x, y) \leq T(x', y')$.

The t-norms can be used to define the intersection of fuzzy sets, as follows: $\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x))$ for all x in U . In a similar way, it can be defined the union and complements of fuzzy sets.

Fuzzy relations $R: U \times U \rightarrow [0, 1]$ have many applications to make fuzzy inference in many branches of Artificial Intelligence with uncertainty, imprecision or lack of knowledge.

Let T be a t-norm [9]. A T -indistinguishability [10] relation is a fuzzy relation on a universe E , $S: E \times E \rightarrow [0,1]$, satisfying reflexivity, symmetry and T -Transitivity (i.e., $T(S(a, b), S(b, c)) \leq S(a, c)$ for all a, b, c in E).

A similarity relation [13] is a Min-indistinguishability.

The classical concept of transitivity is generalised in fuzzy logic by the T -transitivity property of fuzzy relations.

The similarity fuzzy relations generalise the classical equivalence relations, so it is important to define crisp partitions of a set.

The T -transitive closure R^T of a fuzzy relation R is the lowest relation that contains R and is T -transitive. There are many proposed algorithms to compute the T -transitive closure [5].

An algorithm used to compute the transitive closure is the following:

- 1) $R' = R \cup_{\text{Max}} (R \circ_{\text{Sup-T}} R)$
- 2) If $R' \neq R$ then $R := R'$ and go back to 1), otherwise stop and $R^T := R'$.

In this paper, we define fuzzy sets and fuzzy relations on the universe of individuals $U = \{\text{Individual}\}_{i=1..N}$.

4 Fuzzification of Agent Model

The case study presented in [6] has been enriched with the use of fuzzy logic in five aspects: relationships among agents, some variable attributes that determine agent state, and functions of similarity and evolution of agent state, and inheritance. These are discussed below.

The multi-agent system in [6] defines two possible relationships among individuals (agents): *friendship* and *family*. Friendship is naturally predisposed to be fuzzified: defining friendship as a boolean (to be or not to be friend) is unrealistic, because in reality there is a continuous range of grades of friendship. Therefore, friendship is defined as a fuzzy relationship, with real values between 0 and 1.

Let $R_{\text{friend}} : U \times U \rightarrow [0,1]$ be the fuzzy relation on the set of individuals that give a degree of “friendship”. This fuzzy relation gives a degree of friendship for every couple on individuals in the interval $[0, 1]$. And the classic set Friends (Ind) is defined as all the individuals whose $R_{\text{friend}}(\text{Ind})$ is greater than 0.

And so an individual will have a range from true close friends to just “known” people. Of course, some restrictions to this definition could be introduced in order to suit context needs. Equivalent to friendship, we can fuzzificate the family link, although is not as trivial as before: this relationship would be closer to 1 in direct family (parents, brothers), further as the relative relation, and 0 with people whose is not family at all. Thus, R_{family} could be defined in a similar way.

On the other hand, there can be relationships that cannot be fuzzified. There is a “special kind of friendship”: the couple. A stable couple can be perfectly defined as a crisp relation: you have one, or you don’t. There isn’t a specific process for finding a couple, so we decided to ask our sociology expert and build up a general definition for it. An agent will find a couple (always with a random possibility of failure not defined in the mathematical definitions) between the friends that have different sex, are adults and have no couple. The chosen one will be the “most compatible” one, where compatibility is defined as the aggregation of how friends are they and how similar are they. Formally:

Let $R_{\text{has_couple}} : U \rightarrow \{\text{true}, \text{false}\}$ be the crisp relation on the set of individuals that give if an individual has couple or not. This relation is defined by:

$$R_{\text{has_couple}}(\text{Ind}) := (R_{\text{couple}}(\text{Ind}) \neq \text{null})$$

Where $R_{\text{couple}} : U \times U \rightarrow \{\text{Individual}\}$ can be defined as:

$$R_{\text{couple}}(\text{Ind}, \text{Ind2}) := \text{Adult}(\text{Ind}) \text{ AND } \text{Ind2} = \max R_{\text{compatible}}(\text{Ind}, \{\text{Ind}_i \in \text{Friends}(\text{Ind}) \text{ where } (R_{\text{couple}}(\text{Ind}_i) := \text{false AND Sex}(\text{Ind}) \neq \text{Sex}(\text{Ind}_i) \text{ AND Adult}(\text{Ind}_i))\})$$

Where $R_{\text{compatible}} : U \times U \rightarrow \{\text{Individual}\}$ can be defined as:

$$R_{\text{compatible}}(\text{Ind}, \text{Ind2}) := \text{OWA}(R_{\text{friend}}(\text{Ind}, \text{Ind2}), R_{\text{similarity}}(\text{Ind}, \text{Ind2})) = w_1 * R_{\text{friend}}(\text{Ind}, \text{Ind2}) + w_2 * R_{\text{similarity}}(\text{Ind}, \text{Ind2}).$$

An OWA (Ordered Weighted Averaging) [8] is a family of multicriteria combination (aggregation) procedures. By specifying suitable order weights (which sum will result always 1) it is possible to change the form of aggregation: for example, if we want the arithmetic average in the example OWA we can give the value 0.5 to both weights.

But here we have introduced a new concept: similarity. In the original MAS the similarity was modelled and implemented through a not normalized crude gratification algorithm, based on the amount of points gathered from the comparison of the agents' crisp attributes. By defining fuzzy sets over these variables (the attributes) and fuzzifying the similarity operator based on them we will be able to have much more accuracy in the determination of similarity between individuals. Moreover, with those fuzzy set we will be able to make inference based on them.

Therefore, we continue by defining these fuzzy sets based on the variables that define each individual. Even though there are some of them that are not fuzzificable, like the sex, the most part of them will let us define a fuzzy set over them. For example: Let $\mu_{\text{religious}} : U \rightarrow [0,1]$ be the fuzzy set that gives a religious grade based on the religiosity variable of the individual. This set can be defined by segments with different growth (orthodox people, moderated religious ones, agnostics, atheists...) or by a linear function. Thus, $\mu_{\text{religious}}(\text{Ind}) = 0.2$ would mean that this person is mainly not religious. The same way we could continue with other fuzzy sets.

Now, with fuzzy sets over the attributes, we can define the fuzzy similarity. For defining it we will use a T-indistinguishability, which generalizes the classical equivalences. It can be obtained from the negation of a T*-distance, as it is shown in the preliminaries. This way, aggregating the normal distance of each couple of fuzzy sets we can obtain the total similarity between two individuals:

$$R_{\text{similarity}}(\text{Ind}, \text{Ind}2) = \text{OWA} (\forall \mu_i \text{ defined, } N(\mu_i(\text{Ind}) - \mu_i(\text{Ind}2)))$$

We can focus now on the direct interaction between the agents. In this system, we want them to influence each other in some attributes: in sex or age would be impossible, but it's logic that your ideology is influenced by your left-winged friends. This local influence is, by definition, a "fuzzy concept": how much you influence a person can't be easily quantified in any way. It's very difficult to specify this interaction, but after a deep analysis and with the help of our expert, we can dare this mathematical definition:

Let Δ_{Ind}^X be the variation of the attribute "X" of the individual "Ind" by all its environment. We define it as the aggregation of the influence of all its relatives, friends and couple. This influenced is determined by the "proximity" of the person, the distance between the attribute selected, and how young "Ind" is (if the agent is younger, it will be more receptive to be influenced).

$$\Delta_{\text{Ind}_n}^X = \text{OWA}_{i=1 \dots N} (R_{\text{proximity}}(\text{Ind}_n, \text{Ind}_i) * (X_{\text{Ind}_n} - X_{\text{Ind}_i}) * \mu_{\text{young}}(\text{Ind}_n))$$

Let $R_{\text{proximity}}(\text{Ind}_n, \text{Ind}_i) : U \times U \rightarrow [0,1]$ be the fuzzy relation on the set of individuals that give a degree of "proximity". This fuzzy relation is defined by the aggregation (OWA) of the classical relation "couple" with the fuzzy relations Rfriend and Rfamily:

$$R_{\text{proximity}}(\text{Ind}, \text{Ind}2) := \text{OWA}(R_{\text{couple}}(\text{Ind}, \text{Ind}2), R_{\text{friend}}(\text{Ind}, \text{Ind}2), R_{\text{family}}(\text{Ind}, \text{Ind}2)) = w_1 * R_{\text{couple}}(\text{Ind}, \text{Ind}2) + w_2 * R_{\text{friend}}(\text{Ind}, \text{Ind}2) + w_3 * R_{\text{family}}(\text{Ind}, \text{Ind}2)$$

And of course, the evolution of an attribute is determined, by each individual, as $X_{\text{Ind}n} = X_{\text{Ind}n} + \Delta_{\text{Ind}n}^X$.

With continuous influence the global average of the variables will change over time. But there is another source of change: demographic evolution. As time steps go on, agents will find couples and have children. Those children must inherit their parents' characteristics in a way. In the crisp MAS we solve this problem in a rough way: we obtain the new variables from the average of the parents. But now we have defined fuzzy sets based on those variables, so we can use fuzzy operators between them. Thus, we decided to use the fuzzy composition for obtaining the variables of the new individuals (mixed with a random mutation factor not included in the mathematical definition, which was introduced in favour of diversity):

$$\forall X \text{ attribute of Ind, } \mu_x(\text{Ind}) = \mu_x(\text{Father}(\text{Ind})) \circ \mu_x(\text{Mother}(\text{Ind}))$$

Another important side of the agents is their states. The state of an agent is defined by its age, and determines its behaviour. Therefore, an agent in the state of "child" cannot influence adults and cannot find a stable couple... while an "old" agent won't have children and will have more probabilities of dying. But where are the limits between states? In the crisp systems, there are threshold limits that determine that change. For example, if an agent has an age over 27, is in the "adult" state, but if it's under it, is in "young", with a radical different behaviour. This is clearly unrealistic: in reality the changes are gradual, as we are maturing. So we will apply fuzzy logic again, even though this time is quite difficult: it's easy to define how young is an individual, but it's difficult to change gradually its behaviour (anyway, it's an implementation problem, which will not be analyzed here).

The last improvement we propose is for extracting new knowledge from the system, using a fuzzy operation: the T-transitive closure. From the repeatedly application of transitive property of a relationship, it allows us to discover the influence of some variables in other "far" ones (the multiple paths of different length between those variables). This operation suits perfectly with the natural transitivity of the (fuzzy) friendship relation: the friend of my friend is not very friend of me. And so, we can infer inferior friendship grades in others. For example, if A and B are friends in a 0.8, and B and C in 0.6, we will be able to deduce that A and C are friends in a 0.48. And we could continue inferring how a D agent is friend of A, and so on. We are extracting new knowledge from the system, impossible with a crisp relation of friendship (we can't infer that A and C are automatically 100% friends).

We have imposed T-transitivity [3] of the 'friendship' fuzzy relation as a learning process of friendship of individuals with a common friend, and also as a coherence method.

The Zadeh's logic (which uses the t-norm minimum) works well, but we didn't choose it because the friendship of A and C is either the friendship of A and B or the friendship of B and C, and there is no reason for that. A and C are probably not as friend as with the common friend. This logic loses information in this context.

The Lukasiewicz logic uses the Lukasiewicz t-norm [9] defined by $W(a, b) = \max(0, a+b-1)$. It works well when the common friend is a good friend of both, but it

infers no friendship when the common friend is not a very good friend of both individuals.

So, the logic we use to infer friendship is the Product logic (using the t-norm product). It works well for high and low friendship with a common friend, and it does not loose information. For example, if we have a common friend with degree $\frac{1}{2}$, the product logic would decide that we are friends with degree $\frac{1}{4}$, which makes more sense that to infer a friendship degree of $\frac{1}{2}$ (Zadeh's logic) or 0 (Lukasiewicz logic). In the classical world, with only 0 and 1 values, all the logics, and transitivity closure are the same, but when we deal with the uncertainty of friendship, the product logic seems to be the most similar to human reasoning.

5 Results and Conclusions

Part of the proposed fuzzification has been applied to an existing system [6]. More specifically, the *friendship* relation, but not the *family* one. Also, the compatibility concept for matchmaking, with all the restrictions, has implied changes in the similarity algorithm to a fuzzy version. Besides, we extracted new knowledge through the T-transitive closure. There is still the need to implement the fuzzy influence, the fuzzy inheritance, and the fuzzy states. With these changes, similarity operation has improve sharpness; the couples found are much more "logical" (agents know more people than before thanks to the closure, and the similarity measure is better); and the friendship is more realistic (not boolean, with grades, and considers more knowledge thanks to the T-transitive closure).

The proposed fuzzification can be applied to other ABSS systems. For instance, the example has shown how to fuzzify relations that determine agents' interactions, even mixing fuzzy and crisp values (like in the case of "influence" interaction). Also, agents' attributes can be defined in terms of fuzzy sets. Context-dependant functions, like inheritance, were modelled too, as well as a typical fuzzy similarity operation. "Life states" of agents are frequent in systems that evolve over time, especially in task solving environment. Sometimes, it is convenient to fuzzify those states. Finally, a global fuzzy operation over all the agents was defined (the T-transitive closure) on a fuzzy relation (friendship) to make inference with coherent results. Other operations could be needed and applied in a similar way.

In order to adapt these definitions to other systems, new constraints that suit context needs could be introduced. Fuzzy logic facilitates modeling to domain experts, because the linguistic terms can be easily transformed into fuzzy functions. Of course, there will be parts of an ABSS system that cannot be (or just we do not want to) fuzzified, but we think that fuzzy agents in social simulation can be extremely useful.

Future research lines that can be followed include empowering inference and approximate reasoning using fuzzy sets. Fuzzy implications represent a tool with great potential, and have solid foundations because they generalize the classic logical implication. The only special need they have is that the expert has to decide rules from his knowledge, and we must choose the operators that work well in every context. That's not an easy task, more when the field contains huge amounts of uncertainty and imprecision.

Another approach not referred here is to take into account the space and time dimensions. Even though space is implicitly covered when we let an agent to communicate only with its nearby ones, we ignore if an agent is closer than other: geography. This could be seen as another fuzzy relation, where 1 is closest neighbor, and 0 now known at all. About timing, it must be said that all the definitions here should be taken into account, because in our system continuous time does not exist: time is discretised in time steps. This way, all the operations require a step of time. For example, a right way would be: $X_{\text{Indn}}^{s+1} = X_{\text{Indn}}^s + \Delta_{\text{Indn}}^{X, s}$ where “s” is the number of time steps.

Acknowledgements. This work has been performed in the context of the project “Métodos y herramientas para modelado de sistemas multiagente”, supported by Spanish Council for Science and Technology, with grant TIN2005-08501-C03-01.

References

1. Arroyo Menéndez, M., Cambio cultural y cambio religioso, tendencias y formas de religiosidad en la España de fin de siglo. Servicio de Publicaciones de la UCM. Madrid. (2004)
2. Flement, E. P., Mesiar, R., Pap, E.: Triangular norms. Kluwer. Dordrecht. (2000)
3. Garmendia L, Campo C, Cubillo S, Salvador A: A Method to Make Some Fuzzy Relations T-Transitive. International Journal of Intelligent Systems. 14 (9), (1999), 873-882.
4. Inglehart, R., Norris, P. Sacred and Secular. Religion and Politics Worldwide Cambridge University Press. (2004)
5. Naessens, H., De Meyer, H., De Baets, B.: Algorithms for the Computation of T-Transitive Closures, IEEE Trans Fuzzy Systems 10:4 (2002) 541-551.
6. Pavon, J., Arroyo, M., Hassan, S. y Sansores, C.: Simulación de sistemas sociales con agentes software, en Actas del Campus Multidisciplinar en Percepción e Inteligencia, CMPI-2006, volumen I, (2006), 389-400.
7. <http://repast.sourceforge.net/>
8. Ronald R. Yager. Families of OWA operators. Fuzzy Sets and Systems. Volume 57, Issue 3. (1993), 125 - 148
9. Schweizer, B., Sklar, A.: Probabilistic metric spaces. North-Holland, Amsterdam, NL, (1983)
10. Valverde, L.: On the structure of F-indistinguishability operators, Fuzzy Sets and Systems 17, (1985) 313-328.
11. Wolfram, S. A new kind of science. Wolfram Media. (2002)
12. Zadeh, L.A.: Fuzzy sets. Inform. and Control 8, (1965) 338-353.
13. Zadeh, L., A.: Similarity relations and fuzzy orderings, Inform. Sci. 3 (1971) 177-200.

Stage-Dependent Fuzzy-valued Loss Function in Two-Stage Binary Classifier

Robert Burduk

Wroclaw University of Technology, Chair of Systems and Computer Networks,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
robert.burduk@pwr.wroc.pl

Abstract. In this paper, a model to deal with two-stage Bayesian classifier, under the assumption of complete probabilistic information, is introduced. The loss function in our problem is stage-dependent fuzzy-valued. This fuzzy loss function means that the loss depends on the stage at which misclassification is made. The model is firstly based on the notion of fuzzy random variable and secondly on the subjective ranking of fuzzy number defined by Campos and González. The comparison with crisp stage-dependent loss function is given. Finally, an example illustrating this case of Bayesian analysis is considered.

Keywords: Bayes rule, multistage classifier, fuzzy loss function.

1 Introduction

In many decision problems, the utility evaluation can be quite imprecise [2]. The class of the fuzzy-valued loss functions is definitely much wider than the class of the real-valued ones [6], [15]. This fact reflects the richness of the fuzzy expected loss approach to describe the consequences of incorrect classification in contrast with the real-valued approach. For this reason, several studies have been previously described decision problems in which values assessed to the consequences of decision are assumed to be fuzzy [1], [10], [7], [8], [16]. These papers describe only single-stage decision problems.

This paper describes hierarchical classifier problem which is characterized by the fact, that an unknown pattern is classified into a class using several decisions in a successive manner. The mechanics of multistage classification can be described by means of a tree, in which the terminal nodes represent labels of classes and the interior nodes denote the groups of classes. Obviously, the multistage recognition scheme offers much more flexibility than the simple single stage classifier operating with only one set of features and basing its decision on the maximum selection among a set of discriminant values which are calculated independently of each other.

In this paper we proposed two-stage classification model with stage-dependent loss function. This model has been stated so that, on the one hand, the existence of fuzzy loss function representing the preference pattern of the decision maker can be established, on the other hand, a priori probabilities of classes and class-conditional probability density functions can be given.

In the paper [12] the Bayesian hierarchical classifier is presented. The synthesis of multistage classifier is a complex problem. It involves specifying the following components:

- the decision logic, i.e. hierarchical ordering of classes,
- feature used at each stage of decision,
- the decision rules (strategy) for performing the classification.

The present paper is devoted only to the last problem. This means that we shall deal only with the presentation of decision algorithms, assuming that both the tree skeleton and feature used at each non-terminal node are specified. A two-stage classification system with learning was presented in [9], [13].

In the further part, after the introduction of necessary symbols, we will calculate the set of optimal recognition algorithms for internal nodes, minimizing the global quality indicator. As a criterion of optimality we will assume the mean value of the fuzzy loss function (risk), which values depends on the stage of the decision tree, on which an error has occurred.

2 Decision Problem Statement

Let us consider a pattern recognition problem, in which the number of classes is equal to 4. Let us assume that classes were organized in a two-stage binary decision tree. Let us number all the nodes of the constructed decision-tree with consecutive numbers of 0, 1, 2, ..., 6 reserving 0 for the root-node and let us assign numbers of classes from the $\mathcal{M}=\{1,2,\dots,4\}$ set to terminal nodes so that each one of them is labeled with the number of the class which is connected with that node. This allows us to introduce the following notation [11]:

- $\mathcal{M}(n)$ – the set of numbers of nodes, which distance from the root is n , $n=0,1,2,\dots,N$. In particular $\mathcal{M}(0)=\{0\}$, $\mathcal{M}(N)=\mathcal{M}$,
- $\overline{\mathcal{M}} = \bigcup_{n=0}^{N-1} \mathcal{M}(n)$ – the set of interior node numbers (non terminal),
- $\mathcal{M}_i \subseteq \mathcal{M}(N)$ – the set of class numbers attainable from the i -th node ($i \in \overline{\mathcal{M}}$),
- \mathcal{M}^i – the set of numbers of immediate descendant nodes i ($i \in \overline{\mathcal{M}}$),
- m_i – number of immediate predecessor of the i -th node ($i \neq 0$).

We will continue to adopt the probabilistic model of the recognition problem, i.e. we will assume that the class number of the pattern being recognized $j_N \in \mathcal{M}(N)$ and its observed features x are realizations of a couple of random variables \mathbf{J}_N and \mathbf{X} . Complete probabilistic information denotes the knowledge of a priori probabilities of classes:

$$p(j_N) = P(\mathbf{J}_N = j_N), \quad j_N \in \mathcal{M}(N) \quad (1)$$

and class-conditional probability density functions:

$$f_{j_N}(x) = f(x/j_N), \quad x \in X, \quad j_N \in \mathcal{M}(N). \quad (2)$$

Let:

$$x_i \in X_i \subseteq \mathcal{R}^{d_i}, \quad d_i \leq d, \quad i \in \mathcal{M} \quad (3)$$

denote vector of features used at the i -th node, which have been selected from the vector x .

Our aim now is to calculate the so-called multistage recognition strategy $\pi_N = \{\psi_i\}_{i \in \overline{\mathcal{M}}}$, that is the set of recognition algorithms in the form:

$$\Psi_i : X_i \rightarrow \mathcal{M}^i, \quad i \in \overline{\mathcal{M}} \quad (4)$$

Formula (4) is a decision rule (recognition algorithm) used at the i -th node which maps observation subspace to the set of immediate descendant nodes of the i -th node. Equivalently, decision rule (4) partitions observation subspace X_i into disjoint decision regions $D_{x_i}^k$, $k \in \mathcal{M}^i$, such that observation x_i is allocated to the node k if $k_i \in D_{x_i}^k$, namely:

$$D_{x_i}^k = \{x_i \in X_i : \Psi(x_i) = k\}, \quad k \in \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}. \quad (5)$$

The loss function $\tilde{L}(i_N, j_N)$ denote the loss incurred when objects of the class j_N is assigned to the class i_N ($i_N, j_N \in \mathcal{M}(N)$). Our aim is to minimize the mean risk, that is the mean value of the fuzzy loss function:

$$\tilde{R}^*(\pi_N^*) = \min_{\Psi_{i_n}, \Psi_{i_{N-1}}} \tilde{R}(\pi_N) = \min_{\Psi_{i_n}, \Psi_{i_{N-1}}} \tilde{E}[L(I_N, J_N)]. \quad (6)$$

We will refer to the π_N^* strategy as the globally optimal N -stage recognition strategy.

$\tilde{R}^*(\pi_N^*)$ is a fuzzy-valued function on \mathfrak{R} , taking values on the set $F_c(\mathfrak{R})$ (set of normalized convex fuzzy sets on \mathfrak{R} whose level sets and the closed convex hull of the support are compact).

In order to rank fuzzy mean values, we have selected the subjective method defined by Campos and Gonzalez [5]. This method is based on the λ -average valued of a fuzzy number, which is defined for \tilde{A} as the real number given by:

$$V_S^\lambda(\tilde{A}) = \frac{1}{0} \int [\lambda a_{\alpha 2} + (1-\lambda)a_{\alpha 1}] dS(\alpha) \quad (7)$$

where $\tilde{A}_\alpha = [a_{\alpha 1}, a_{\alpha 2}]$, $\lambda \in [0, 1]$ and S being an additive measure on $Y \subset [0, 1]$.

The λ parameter is a subjective degree of optimism-pessimism. In a loss context, $\lambda=0$ reflects the highest optimism, while $\lambda=1$ reflects the highest pessimism. Then, the λ -ranking method used for comparing fuzzy numbers is given by:

$$\tilde{A} \geq \tilde{B} \Leftrightarrow V_S^\lambda(\tilde{A}) \geq V_S^\lambda(\tilde{B}). \tag{8}$$

This λ -average ranking method extends some well-known ranking functions [4], [17]. One of the most relevant characteristics of the ranking method based on the function V_S^λ is its feasibility, which is due to the following reason: when we apply V_S^λ on the fuzzy expected value of an integrably bounded fuzzy random variable the computation of this fuzzy expected value is not required. The λ -average value of a fuzzy expected value is reduced to the expected value of a classical random variable, namely, the composition of V_S^λ and the fuzzy variable [14].

In the next chapter we will calculate globally optimal strategy for stage-dependent fuzzy loss function.

3 The Recognition Algorithm

The fuzzy loss function $\tilde{L}(i_N, j_N)$ denote the loss incurred when objects of the class j_N is assigned to the class i_N ($i_N, j_N \in \mathcal{M}(N)$). Let us assume now:

$$\tilde{L}(i_N, j_N) = \tilde{L}_{d(w)}^s \tag{9}$$

where w is the first common predecessor of the nodes i_N and j_N ($i_N, j_N \in \mathcal{M}(N)$). So defined fuzzy loss function means that the loss depends on the stage at which misclassification is made. Stage-dependent fuzzy loss function for the two-stage binary classifier are presented in Fig. 1.

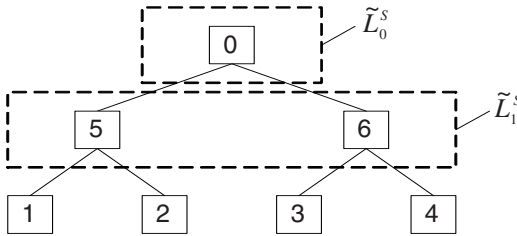


Fig. 1. Interpretation of stage-dependent fuzzy loss function in two-stage classifier

By putting (9) into (6) we obtain the optimal (Bayes) strategy, whose decision rules are as follows:

$$\Psi_{i_n}^*(x_{i_n}) = i_{n+1}$$

$$\begin{aligned}
& (\tilde{L}_0^s - \tilde{L}_1^s) p(i_{n+1}) f_{i_{n+1}}(x_{i_n}) + \tilde{L}_1^s \sum_{J_N \in M^{J_N-1}} [q^*(j_N / i_{n+1}, j_N) p(j_N) f_{j_N}(x_{i_n})] = \\
& = \max_{k \in M^n} \{ (\tilde{L}_0^s - \tilde{L}_1^s) p(k) f_k(x_{i_n}) + \tilde{L}_1^s \sum_{J_N \in M^k} [q^*(j_N / k, j_N) p(j_N) f_{j_N}(x_{i_n})] \}
\end{aligned} \tag{10}$$

where $q^*(j_N / i_{n+1}, j_N)$ denotes the probability of accurate classification of the object of the class j_N in second stages using π_N^* strategy rules on condition that on the first stage the i_{n+1} decision has been made.

4 Illustrative Example

Let us consider the two-stage binary classifier present in Fig 1. Four classes have identical a priori probabilities which are equal 0.25. Class-conditional probability density functions of features X_0 , X_5 and X_6 are normally distributed in each class with the following class-conditional probability density functions: $f_1(x_0) = f_2(x_0) = N(1, 1)$, $f_3(x_0) = f_4(x_0) = N(3, 1)$, $f_1(x_5) = N(1, 1)$, $f_2(x_5) = N(4, 1)$, $f_3(x_6) = N(0, 1)$, $f_4(x_6) = N(2, 1)$.

The stage-dependent fuzzy loss functions are described by a triangular fuzzy numbers and are the following: case a $\tilde{L}_0^s = (2.5, 3, 3.5)_T$, $\tilde{L}_1^s = (0.5, 1, 1.5)_T$, case b $\tilde{L}_0^s = (1, 2, 3)_T$, $\tilde{L}_1^s = (0, 0.5, 1)_T$, case c $\tilde{L}_0^s = (1, 2, 3)_T$, $\tilde{L}_1^s = (0, 1, 2)_T$, case d $\tilde{L}_0^s = (0, 2, 4)_T$, $\tilde{L}_1^s = (0, 1, 2)_T$, case e $\tilde{L}_0^s = (2, 4, 6)_T$, $\tilde{L}_1^s = (0, 0.5, 1)_T$, case f $\tilde{L}_0^s = (1, 2, 2.5)_T$, $\tilde{L}_1^s = (0, 0.5, 1)_T$.

Due to the peculiar distribution of X_5 and X_6 , the decision rules Ψ_5^* and Ψ_6^* , at the second stage of classification, the separation point are following: $x'_5 = 2.5$ and $x'_6 = 1$. Let us now determine the rule Ψ_0^* at the first stage of classification. From (10) we obtain:

$$\Psi_0^*(x_0) = \begin{cases} 5, & \text{if } \begin{cases} (\tilde{L}_0^s - \tilde{L}_1^s) p(5) f_5(x_0) + \\ + \tilde{L}_1^s (q^*(1/5, 1) p(1) f_1(x_0) + \\ + q^*(2/5, 2) p(2) f_2(x_0)) > \\ > (\tilde{L}_0^s - \tilde{L}_1^s) p(6) f_6(x_0) + \\ + \tilde{L}_1^s (q^*(3/6, 3) p(3) f_3(x_0) + \\ + q^*(4/6, 4) p(4) f_4(x_0)), \\ 6 & \text{otherwise,} \end{cases} \\ 6 & \end{cases}$$

Using the data from the example and subjective λ -method for comparison fuzzy risk, we finally obtain results presented in Fig. 2, where value of point x_0^* (separation point for decision regions at the first stage) in function of parameter λ is presented.

For crisp stage-dependent loss function (value equals 2 on first stage and value equals 1 on second stage) we have $x_0^* = 2.025$.

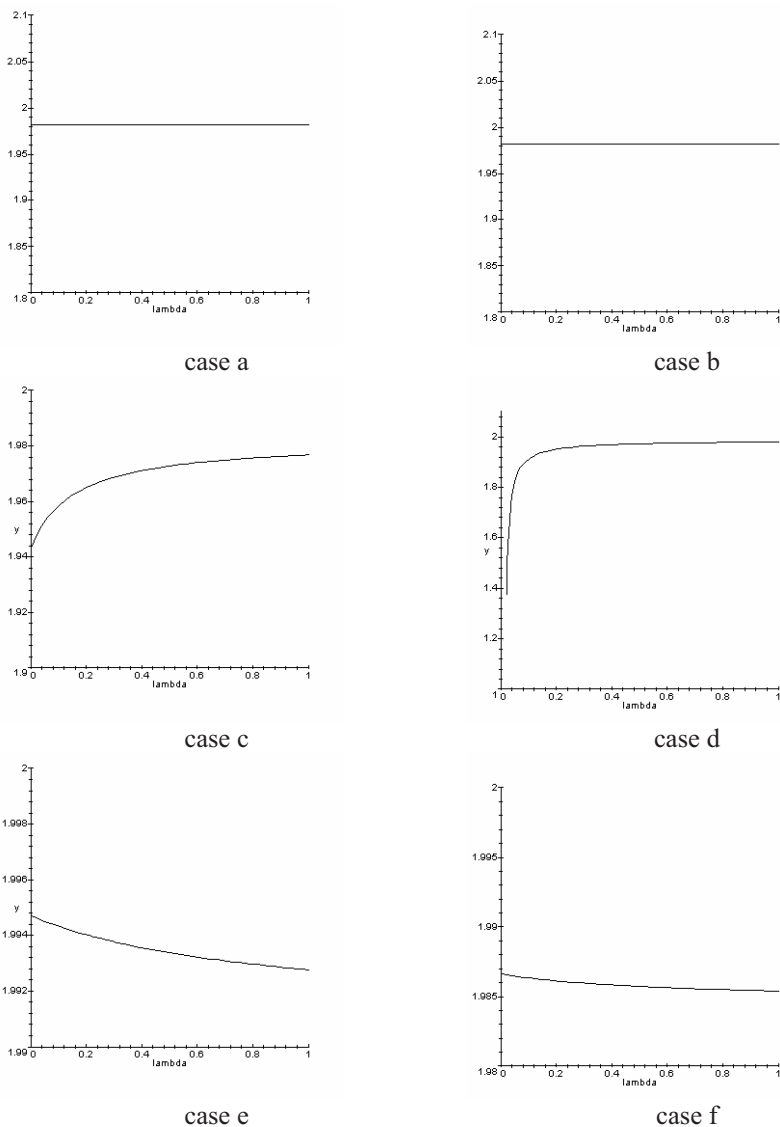


Fig. 2. Separation point for decision regions at the first stage in function of parameter λ - figure caption right to case

Let us denote $\tilde{L}_0^S - \tilde{L}_1^S = (a_1, a_2, a_3)_T$ and $\tilde{L}_1^S = (b_1, b_2, b_3)_T$. Then, we have a constant function for $(a_1 + a_2)(b_2 + b_3) - (a_2 + a_3)(b_1 + b_2) = 0$, the separation

point x_0^* is independent of the choice of the λ parameter. Additionally, we have a descending function for $(a_1 + a_2)(b_2 + b_3) - (a_2 + a_3)(b_1 + b_2) > 0$ and an ascending function for $(a_1 + a_2)(b_2 + b_3) - (a_2 + a_3)(b_1 + b_2) < 0$.

In the case *c* we have $x_0^* = 2.025$ for $\lambda = 0.5$ (the same value as for crisp stage-dependent loss function). This situation is always true for $a_2 = 1$ and $b_2 = 1$ because $V_S^{0.5}(\tilde{L}_0^S - \tilde{L}_1^S) = 1$ and $V_S^{0.5}(\tilde{L}_1^S) = 1$. In the case *d* we do not have value $x_0^* = 2.025$ for each $\lambda \in [0, 1]$. It is because for none of $\lambda \in [0, 1]$ $V_S^{0.5}(\tilde{L}_0^S - \tilde{L}_1^S) = 1$ and $V_S^{0.5}(\tilde{L}_1^S) = 1$. It is not possible to choose of the λ for extreme values (0 or 1). In the case *d* for $\lambda = 0$ we have $x_0^* = 2.91$, for $\lambda = 0.05$ $x_0^* = 1.81$ and $\lambda = 0.1$ $x_0^* = 1.91$. For such cases an analytical results are necessary.

5 Conclusion

In the paper we have presented two-stage Bayes classifier with a full probabilistic information. In this recognition model a priori probabilities of classes and class-conditional probability density functions are given. Additionally, consequences of wrong decision are fuzzy-valued and are represented by triangular fuzzy numbers. For ranking fuzzy numbers we use subjective method with parameter λ . In this work we have considered algorithm for hierarchical classifier with stage-dependent fuzzy loss function when observation of the features are crisp. In [3] the node-dependent loss function is also considered. This paper tries to contribute to a better understanding of the impact a choice of a fuzzy numbers which describe stage-dependent loss function. In future work we can consider another multistage classifier or fuzzy loss function forms. The theoretical results are also necessary for a better understanding of choice of λ parameter.

Acknowledgements. This work is supported by The Polish State Committee for Scientific Research under the grant which is realizing in years 2006-2009.

References

1. Baas, S., Kwakernaak, H.: Rating and Ranking of Multi-Aspect Alternatives Using Fuzzy Sets. *Automatica* 13, 47-58 (1997)
2. Berger, J.: *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York (1993)
3. Burduk, R., Kurzyński, M.: Two-stage binary classifier with fuzzy-valued loss function, *Pattern Analysis and Applications* 9, 353-358, (2006)
4. Bortolan, G., Degani, R.: A Review of Some Methods for Ranking Fuzzy Subsets. *Fuzzy Sets and Systems* 15, 1-19 (1985)
5. Campos, L., González, A.: A Subjective Approach for Ranking Fuzzy Numbers. *Fuzzy Sets and Systems* 29, 145-153 (1989)

6. Domingos, P.: MetaCost: A General Method for Making Classifiers Cost-Sensitive. In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99), 155-164 (1999)
7. Gil, M.A., López-Dáz, M.: Fundamentals and Bayesian Analyses of Decision Problems with Fuzzy-Valued Utilities. *International Journal of Approximate Reasoning* 15, 203-224 (1996)
8. Gil, M.A., López-Dáz, M.: A Model for Bayesian Decision Problems Involving Fuzzy-Valued Consequences. Proc. 6th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge Based Systems, Granada 495-500 (1996)
9. Giusti, N., Sperduti, A.: Theoretical and experimental analysis of a two-stage system for classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 893-904 (2002)
10. Jain, R.: Decision-Making in the Presence of Fuzzy Variables. *IEEE Trans. Systems Man and Cybernetics* 6, 698-703 (1976)
11. Kurzyński, M.: Decision Rules for a Hierarchical Classifier. *Pattern Recognition Letters* 1, 305-310 (1983)
12. Kurzyński, M.: On the Multistage Bayes Classifier. *Pattern Recognition* 21, 355-365 (1988)
13. Li-Wei, K., Bor-Chen K., Chin-Teng L., Der-Jenq, L.: A Two-Stage Classification System with Nearest Neighbor Reject Option for Hyperspectral Image Data, In 18th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP) Taipei, 1258-1263 (2005)
14. López-Dáz, M., Gil, M.A.: The λ -average value of the expected value of a fuzzy random variable. *Fuzzy Sets and Systems* 99, 347-352 (1998)
15. Margineantu, D.D., Dietterich, T.G.: Bootstrap methods for the cost-sensitive evaluation of classifiers, In Proc. 17 Int. Conf. on Machine Learning ICML'2000, Morgan Kaufmann, 583-590, (2000)
16. Viertl, R.: *Statistical Methods for Non-Precise Data*. CRC Press, Boca Raton (1996)
17. Yager, R.: A Procedure for Ordering Fuzzy Subsets of the Unit Interval. *Information Sciences* 22, 143-161 (1981)

A Feature Selection Method Using a Fuzzy Mutual Information Measure

Javier Grande, Mará del Rosario Suárez, and José Ramón Villar

Computer Science Department, University of Oviedo
j.grandegundin@gmail.com, mrsuarez@uniovi.es, villarjose@uniovi.es

Abstract. Attempting to obtain a classifier or a model from datasets could be a cumbersome task, specifically in datasets with a high dimensional datasets. The larger the amount of features the higher the complexity of the problem, and the larger the time expended in generating the outcome -the classifier or the model-. Feature selection has been proved as a good technique for eliminating features that do not add information of the system. There are several different approaches for feature selection, but until our knowledge there are not many different approaches when feature selection is involved with imprecise data and genetic fuzzy systems. In this paper, a feature selection method based on the fuzzy mutual information is proposed. The outlined method is valid either for classifying problems when expertise partitioning is given, and it represents the base of future work including the use of the in case of imprecise data.

Keywords: Mutual Information, classification, feature selection, imprecise data, fuzzy systems.

1 Introduction

When attempting to generate a classifier or a model based from a dataset which is obtained from a real process, there are some facts that must be taken into account [18, 17]. On the one hand, the number of features in the dataset, and the number of examples as well, will surely be high. Furthermore, it is not known which of the features are relevant or not, nor the interdependency relations between them. On the other hand, the data obtained from real processes is vague data due to the precision of the sensors and transducers, the losses in A/D conversions, the sensitivity and sensibility of the sensors, etc.

It is well known that the former fact is alleviated by means of the feature selection techniques. There are several techniques in the literature facing such a problem. This feature selection must be carried out in such a way that the reduced dataset keeps as much information as possible about the original process. In other words, redundant features and features that do not possess information about the process are the ones to be eliminated [24]. However, in the feature selection process it must be taken into account that datasets from real processes are imprecise, so the feature selection decisions must be influenced by such vagueness [22].

It is important to point out that the data impreciseness affects the way in which the behaviour of each feature is managed. Fuzzy logic has been proved as a suitable

technique for managing imprecise data [15, 16]. Whenever imprecise data is present fuzzy logic is going to be used in order to select the main features so the losses in information from real processes could be reduced [17].

This paper intends to evaluate different approaches for feature selection in datasets gathered from real processes. The approaches must be valid to be extended with the fuzzy mutual information (from now on referred as FMI) measure detailed in [19, 22], so the final method would face imprecise data. In this paper it will be shown that using expertise partitioning, and a feature selection method based on the FMI measure, a suitable approach for solving classification problems will be provided. In order to prove that idea, the experiments are to compare the error rate for several classifiers when feature selection is applied. Finally some ideas about future work using the FMI are proposed.

The paper is set out as follows. Firstly, a review of the literature is carried out. Then, a description of the developed algorithms is shown in Sec. 3. Experiments run and results are shown in Sec. 4. Finally, conclusions and future work are commented in Sec. 5.

2 Feature Selection Methods

Real processes generate high dimensionality datasets. In other words, the obtained datasets have an important number of input features, which are supposed to describe the desired output. In practical cases, some input features may be ignored without losing information about the output. This problem is called *feature selection*, and it intends to choose the smaller subset of input features that best describes the desired output [11]. Fuzzy systems are known to be suitable when it is necessary to manage uncertainty and vagueness. The uncertainty in the datasets will influence the feature selection methods, and the fuzzy classifiers and models to be obtained. Feature selection methods related to the problem of managing uncertainty in data will be analyzed below.

There are several feature selection techniques available in the literature. Some authors have proposed a taxonomy of the feature selection algorithms according to how the method must be used and how the method works [9, 25]. According to how the method must be used, feature selection methods are classified as *filters* or as *wrappers*. As filters they are known the feature selection methods that are used as a preprocess method. As wrappers they are known the feature selection methods that are embedded in the whole solution methods, that is, in classification, the feature selection method is included in the optimization method used. The former methods are usually faster than the latter, with lower computation costs. But the wrapper methods performance is usually better than filter methods, and a more suitable feature set is supposed to be selected.

The *Relief* and the *SSGA Integer knn method* are an example of each type of feature selection method. The *Relief* method is a filter method that uses the knn algorithm and the information gain to select the feature subset [8]. The *SSGA Integer knn method* [3], which is a wrapper method, makes use of a filter feature selection method and then a wrapper feature selection method for obtaining a fuzzy rule based classifier.

This wrapper makes use of a genetic algorithm to generate a feature subset which is evaluated by means of a knn classifier. A similar work is presented in [29].

In any case, a wrapper can also be used as a filter, as shown in [13]. In this work, a predefined number of features are given. An optimization algorithm is used to search for the combination of features that give the best classification error rate. Two subsets of features with the same classification error rate are sorted by means of distance measure, which assesses the certainty with which an object is assigned to a class.

According to how the method works there are three possibilities: the *complete search* methods, the *heuristic search* methods and the *random search* methods. The complete search methods are employed when domain knowledge exists to prune the feature search space. Different approaches are known for complete search methods: the *branch & bound* approach, which is assumed to eliminate all the features with evaluation function values lower than a predefined bound, and the *best first search* approach, which searches the feature space until the first combination of features that produces no inconsistencies with the data is obtained.

Heuristic search methods are the feature selection methods that search for a well suited feature set by means of a heuristic search method and an evaluation function. The heuristics used are simple techniques, such hill-climbing could be. Also, the search is known as *Sequential Forward Search* -from now on, SFS- or *Sequential Backward Search* -from now on, SBS-. A heuristic search is called SFS if initially the feature subset is empty, and in each step it is incremented in one feature.

In [1] a SFS Method is detailed. This method makes use of the mutual information between each feature and the class and the mutual information between each pair of features. In each step the best evaluated feature -the one with the highest former mutual information measure- is chosen to be a member of the feature subset if the value of the latter mutual information measure is lower than a predefined bound. A similar feature selection application is the one presented in [28].

Another SFS method is presented in [9], where the fcm clustering algorithm is used to choose the features. Based on the discrimination index of a feature with regard to a prototype of a cluster, the features with higher index value are included in the feature subset. Although it is not feature selection but rather feature weighting, in [26] a gradient based search is used to calculate the weight vector and then a weighted FCM to obtain a cluster from data is used.

The search is *SBS* if at the beginning the feature subset is equal to the feature domain, and in each step the feature subset is reduced in one feature. Finally, the random search methods are those that make use of a random search algorithm in determining the smaller feature subset. Genetic algorithms are typically employed as the random search method.

In [14] a SBS method is shown using the Fisher algorithm. The Fisher algorithm is used for discarding the lowest evaluated feature in each step. The evaluating function is the Fisher interclass separability. Once the feature subset is chosen, then a model is obtained by means of a genetic algorithm. Another SBS contribution is shown in [11]. An interval model for features could be admitted. In this paper, a FCM clustering is run, and each feature is indexed according to its importance. The importance is evaluated as the difference between the Euclidean distances of the examples to the cluster prototype with and without the feature. The larger the difference, the more

important the feature is. Each feature is evaluated with a real value although features are considered interval.

In [25] a boosting of sequential feature selection algorithms is used to obtaining a final feature subset. The evaluation function for the two former is the root mean square error. The third method uses a correlation matrix as feature evaluation function. Finally, the latter uses as feature evaluation function the inconsistency measure.

Random search methods make use of genetic algorithms, simulated annealing, etc. The works detailed above [3, 29] could be considered of this type. Also the work presented in [23] makes use of a genetic algorithm to select the feature subset.

Imprecision and vagueness in data have been included in feature selection for modelling problems. In [20, 21, 6, 27] SBS feature selection methods have been presented taking into account the vagueness of data through the fuzzy-rough sets. In [20] foundations are presented, where in [21] the SBS algorithm is detailed. Finally, an ant colony algorithm is employed in [6, 7]. The same idea has been successfully reported for classification purposes in [27], using the particle swarm optimization algorithm. An important issue concerning the t-norms and t-co norms is analyzed in [2], where non convergence problems due to the use of the max t-co norm is reported. Also, a solution by means of the product t-norm and the sum t-co norm is proposed.

3 The Implemented Feature Selection Algorithms

This paper deals with feature selection for obtaining classifiers with imprecise and vague problems. Mutual information is the tool intended to be used because it helps to choose the features that possess maximum information about the desired output. In order to use such a measure in feature selection for classification problems, the Battiti feature selection algorithm has been shown as a fast and efficient solution. But, to our knowledge, the Battiti approach has not been used in regression problems, so it should be extended. Also, when there is imprecision in the data, the mutual information defined for crisp data is not valid. In such problems, the mutual information measure employed should manage vagueness and imprecision.

Extending the Battiti algorithm to regression problems is not difficult if a discretization schema is taken into account and applied as a dataset preprocess stage. But managing imprecision is a more difficult problem. The mutual information measure must be defined to include the imprecision in calculations. To the best of our knowledge, in the literature there is not approach to feature selection that accomplishes with pure uncertainty data.

In [19, 22] a definition of the Fuzzy Mutual Information (from now on, FMI) measure is done, and an efficient algorithm for computing such measure is presented. It is applied to feature discretization, and results have shown two main ideas. Firstly, the fuzzy mutual information measure defined is a valid measure for both for discrete and imprecise data. Moreover, the result of the FMI measure is the same if discrete data is fed. And secondly, the discretization with such measure outperforms those obtained with similar methods using the classical mutual information definition.

```

Input: D the imprecise dataset, F the feature space,
      d the desired output
      k the number of feature to be chosen
Output: H the feature subset
Let  $f_j$  the  $j$ -th feature in the F,  $j \in [1, |F|]$ 
Set  $H = \{\}$ 
For each  $f \in F$ 
  compute  $I(d, f)$ , I is the fuzzy mutual information
  measure
Find a feature  $f$  that is not dominated over  $I(d, f)$ 
  Let  $F = F - f$ 
  Let  $H = H \cup f$ 
Repeat until  $|H| == k$ 
  Compute  $I(f, h)$  if it is not available, where  $f \in F$ ,  $h \in H$ 
  Select  $f \in F$  that is nondominated over  $I(d, f) - \beta \sum_{h \in H} I(f, h)$ 
    Let  $F = F - f$ 
    Let  $H = H \cup f$ 
The output is H

```

Fig. 1. Feature Selection Algorithm in presence of imprecise data

Concluding, we propose that the feature selection algorithm proposed by Battiti [1] (in following, MIFS) could be extended to regression problems by means of a discretization preprocess stage. But also, we propose the use of the FMI measure instead of the classic mutual information measure used in the referred paper. The whole algorithm, then, is shown in Fig. 1 which will be referred to as FMIFS. It is worth noticing that there are not too many differences with the original algorithm. Specifically, when crisp data is given for a classification problem, the algorithm performs as the Battiti algorithm.

4 Experiments and Results

This section will analyze how the FMI based feature selection method behaves. Two more feature selection methods are used to test the validity of our proposal, both from those implemented in the KEEL project [12]. Specifically, the feature selection methods employed are the Relief and the SSGA Integer Knn methods. The dataset tested is the wine dataset about the chemical analysis of wines grown in a specific area of Italy, with 13 features, 3 class values and 178 examples.

Moreover, thirteen different fuzzy rule learning algorithms have been considered, both heuristic and genetic algorithm based. The heuristic classifiers are described in [5]: no weights (HEU1), same weight as the confidence (HEU2), differences between the confidences (HEU3, HEU4, HEU5), weights tuned by reward-punishment (REWP) and analytical learning (ANAL). The genetic classifiers are: Selection of rules (GENS), Michigan learning (MICH) -with population size 25 and 1000 generations,- Pittsburgh learning (PITT) -with population size 50, 25 rules each individual and 50 generations,- and Hybrid learning (HYBR) -same parameters as PITT, macromutation with probability 0.8- [5]. Lastly, two iterative rule learning

algorithms are studied: Fuzzy Ababoost (ADAB) -25 rules of type I, fuzzy inference by sum of votes- [4] and Fuzzy Logitboost (LOGI) -10 rules of type III, fuzzy inference by sum of votes- [10]. All the experiments have been repeated ten times for different permutations of the datasets (10cv experimental setup), and are shown in Table 1. As can be seen, it can not be stated which of the methods SSGA or the FMIFS is better, and both are better than the Relief and MIFS, as expected.

Table 1. The average classification error after the 10 k fold cross validation of the different fuzzy rule-based classifiers after performing a feature selection, with the original MIFS algorithm and with the modified version proposed in this paper. The number of features selected is 5 features for all of the methods.

Relief	SSGA	MIFS	FMIFS	
HEU1	0.500	0.176	0.323	0.176
HEU2	0.411	0.176	0.323	0.117
HEU3	0.235	0.147	0.264	0.147
HEU4	0.205	0.235	0.205	0.176
HEU5	0.176	0.147	0.176	0.176
REWP	0.088	0.058	0.117	0.088
ANAL	0.235	0.088	0.235	0.117
GENS	0.117	0.147	0.205	0.088
MICH	0.647	0.176	0.617	0.205
PITT	0.117	0.176	0.205	0.088
HYBR	0.176	0.117	0.176	0.058
ADAB	0.058	0.000	0.058	0.029
LOGI	0.058	0.029	0.058	0.088
Best	0	8	0	7

5 Conclusions and Future Work

Experiments show that the FMIFS could be a valid feature selection method. When discrete data is present the selected features are suitable. But more experimentation is needed in order to find the kind of problem for which this method better fits. Also, imprecise datasets must be generated and tested, for which the fuzzy mutual information measure has been developed. Future works also includes analysing who missing data must be processed, and how this measure could be used with different feature selection methods apart from that of Battiti.

Acknowledgments. This work was funded by Spanish M. of Education, under the grant TIN2005-08386-C05.

References

1. Battiti, R. Using Mutual Information For Selecting Features In Supervised Neural Net Learning. *Ieee Transactions On Neural Networks* 5, 4 (1994), 537–550.
2. Bhatt, R. B., And Gopal, M. On Fuzzy-Rough Sets Approach To Feature Selection. *Pattern Recognition Letters*, 26 (2005), 965–975.

3. Casillas, J., Cordón, O., Jesús, M. J. D., And Herrera, F. Genetic Feature Selection In A Fuzzy Rule-Based Classification System Learning Process For High-Dimensional Problems. *Information Sciences*, 136 (2001), 135–157.
4. Del Jesús, M. J., Junco, F. H. L., And Sánchez, L. Induction Of Fuzzy-Rule-Based Classifiers With Evolutionary Boosting Algorithms. *Ieee Transactions On Fuzzy Systems* 12, 3 (2004), 296–308.
5. Ishibuchi, H., Nakashima, T., And Nii, M. *Classification And Modelling With Linguistic Information Granules*. Springer, 2004.
6. Jensen, R., And Shen, Q. Fuzzy-Rough Data Reduction With Ant Colony Optimization. *Fuzzy Sets And Systems*, 149 (2005), 5–20.
7. Jensen, R., And Shen, Q. Fuzzy-Rough Sets Assisted Attribute Selection. *Ieee Transactions On Fuzzy Systems* 15, 1 (2007), 73–89.
8. Kira, K., And Rendell, L. A Practical Approach To Feature Selection. In *Proceedings Of The Ninth International Conference On Machine Learning (Icml- 92)* (1992), Pp. 249–256.
9. Marcelloni, F. Feature Selection Based On A Modified Fuzzy C-Means Algorithm With Supervision. *Information Sciences*, 151 (2003), 201–226.
10. Otero, J., And Sánchez, L. Induction Of Descriptive Fuzzy Classifiers With The Logitboost Algorithm. *Soft Computing* 10, 9 (2005), 825–835.
11. Pedrycz, W., And Vukovich, G. Feature Analysis Through Information Granulation And Fuzzy Sets. *Pattern Recognition*, 35 (2002), 825–834.
12. Project, T. K. [Http://Www.Keel.Es](http://www.Keel.es). Tech. Rep.
13. Ravi, V., And Zimmermann, H.-J. Fuzzy Rule Based Classification With Feature Selector And Modified Threshold Accepting. *European Journal Of Operational Research*, 123 (2000), 16–28.
14. Roubos, J. A., Setnes, M., And Abonyi, J. Learning Fuzzy Classification Rules From Labelled Data. *Information Sciences*, 150 (2003), 77–93.
15. Sánchez, L., And Couso, I. Advocating The Use Of Imprecisely Observed Data In Genetic Fuzzy Systems. In *Proceedings Of I International Workshop On Genetic Fuzzy Systems, Gfs 2005* (2005).
16. Sánchez, L., And Couso, I. Advocating The Use Of Imprecisely Observed Data In Genetic Fuzzy Systems. *Ieee Transactions On Fuzzy Systems, In Press* (2006).
17. Sánchez, L., Otero, J., And Casillas, J. Modelling Vague Data With Genetic Fuzzy Systems Under A Combination Of Crisp And Imprecise Criteria. In *Proceedings Of The First Ieee Symposium On Computational Intelligence In Multi-Criteria Decision-Making, Mcdm2007* (Honolulu, Usa, 2007).
18. Sánchez, L., Otero, J., And Villar, J. R. Boosting Of Fuzzy Models For High Dimensional Imprecise Datasets. In *Proceedings Of The Information Processing And Management Of Uncertainty In Knowledge-Based Systems Ipmu06* (Paris, France, 2006).
19. Sánchez, L., Suárez, M. R., And Couso, I. A Fuzzy Definition Of Mutual Information With Application To The Design Of Genetic Fuzzy Classifiers. In *Proceedings Of The Ieee International Conference On Fuzzy Systems, Fuzzieee 2007* (London, Uk, 2007).
20. Shen, Q., And Chouchoulas, A. A Rough-Fuzzy Approach For Generating Classification Rules. *Pattern Recognition*, 35 (2002), 2425–2438.
21. Shen, Q., And Jensen, R. Selecting Informative Features With Fuzzy-Rough Sets And Its Application For Complex Systems Monitoring. *Pattern Recognition*, 37 (2004), 1351–1363.
22. Suárez, M. R. *Estimación De La Información Mutua En Problemas Con Datos Imprecisos*. Phd Thesis, University Of Oviedo, Gijón, Spain, April 2007.

23. Tolvi, J. Genetic Algorithms For Outlier Detection And Variable Selection In Linear Regression Models. *Soft Computing*, 8 (2004), 527–533.
24. Tourassi, G. D., Frederik, E. D., Markey, M. K., And Carey E. Floyd, J. Application Of The Mutual Information Criterion For Feature Selection In Computer-Aided Diagnosis. *Med. Phys.* 28, 12 (2001), 2394–2402.
25. Uncu, O., And Turksen, I. A Novel Feature Selection Approach: Combining Feature Wrappers And Filters. *Information Sciences*, 177 (2007), 449–466.
26. Wang, X., Wang, Y., And Wang, L. Improving Fuzzy C-Means Clustering Based On Feature-Weight Learning. *Pattern Recognition Letters*, 25 (2004), 1123– 1132.
27. Wang, X., Yang, J., Jensen, R., And Liu, X. Rough Set Feature Selection And Rule Induction For Prediction Of Malignancy Degree In Brain Glioma. *Computer Methods And Programs In Biomedicine*, 83 (2006), 147–156.
28. Yu, D., Hu, Q., And Wu, C. Uncertainty Measures For Fuzzy Relations And Their Applications. *Applied Soft Computing*, 7 (2007), 1135–1143.
29. Yu, S., Backer, S. D., And Scheunders, P. Genetic Feature Selection Combined With Composite Fuzzy Nearest Neighbour Classifiers For Hyper Spectral Satellite Imagery. *Pattern Recognition Letters*, 23 (2002), 183–190.

Interval Type-2 ANFIS

Gerardo M. Mendez¹ and Ma. De Los Angeles Hernandez²

¹ Instituto Tecnológico de Nuevo Leon
Department of Electronics and Electrical Engineering
Av. Eloy Cavazos 2001, Cd. Guadalupe, NL, Mexico, CP 67170
gmm_paper@yahoo.com.mx

² Instituto Tecnológico de Nuevo Leon
Department of Economic Sciences
Av. Eloy Cavazos 2001, Cd. Guadalupe, NL, Mexico, CP 67170
ahernandez@yturria.com.mx

Abstract. This article presents a new learning methodology based on a hybrid algorithm for interval type-1 non-singleton type-2 TSK fuzzy logic systems (FLS). Using input-output data pairs during the forward pass of the training process, the interval type-1 non-singleton type-2 TSK FLS output is calculated and the consequent parameters are estimated by the recursive least-squares (RLS) method. In the backward pass, the error propagates backward, and the antecedent parameters are estimated by the back-propagation (BP) method. The proposed hybrid methodology was used to construct an interval type-1 non-singleton type-2 TSK fuzzy model capable of approximating the behavior of the steel strip temperature as it is being rolled in an industrial Hot Strip Mill (HSM) and used to predict the transfer bar surface temperature at finishing Scale Breaker (SB) entry zone. Comparative results show the performance of the hybrid learning method (RLS-BP) against the only BP learning.

Keywords: Interval type-2 fuzzy logic systems, ANFIS, neuro fuzzy systems, hybrid learning.

1 Introduction

Interval type-2 (IT2) fuzzy logic systems (FLS) constitute an emerging technology. In [1] both, one-pass and back-propagation (BP) methods are presented as IT2 Mamdani FLS learning methods, but only BP is presented for IT2 Takagi-Sugeno-Kang (TSK) FLS systems. One-pass method generates a set of IF-THEN rules by using the given training data one time, and combines the rules to construct the final FLS. When BP method is used in both Mamdani and TSK FLS, none of antecedent and consequent parameters of the IT2 FLS are fixed at starting of training process; they are tuned using exclusively steepest descent method. In [1] recursive least-squares (RLS) and recursive filter (REFIL) algorithms are not presented as IT2 FLS learning methods.

The hybrid algorithm for IT2 Mamdani FLS has been already presented elsewhere [2, 3, 4] with three combinations of learning methods: RLS-BP, REFIL-BP and orthogonal least-squares-BP (OLS-BP), whilst the hybrid algorithm for singleton IT2 TSK FLS (IT2 ANFIS) has been presented elsewhere [5] with two combinations of learning methods: RLS-BP and REFIL-BP.

The aim of this work is to present and discuss a new hybrid learning algorithm for interval type-1 non-singleton type-2 TSK FLS (IT2 NSFLS-1 or IT2 NS-1 ANFIS) using RLS-BP combination in order to estimate the antecedent and consequent parameters during the training process. The proposed IT2 NS-1 ANFIS system is evaluated making transfer bar surface temperature predictions at Hot Strip Mill (HSM) Finishing Scale Breaker (SB) entry zone.

2 Problem Formulation

Most of the industrial processes are highly uncertain, non-linear, time varying and non-stationary [2, 6], having very complex mathematical representations. IT2 NS-1 ANFIS takes easily the random and systematic components of type A or B standard uncertainty [7] of industrial measurements. The non-linearities are handled by FLS as identifiers and universal approximators of nonlinear dynamic systems [8, 9, 10, 11]. Stationary and non-stationary additive noise is modeled as a Gaussian function centred at the measurement value. In stationary additive noise the standard deviation takes a single value, whereas in non-stationary additive noise the standard deviation varies over an interval of values [1]. Such characteristics make IT2 NS-1 ANFIS a powerful inference system to model and control industrial processes.

Only the BP learning method for IT2 TSK SFLS has been proposed in the literature and it is used as a benchmark algorithm for parameter estimation or systems identification [1]. To the best knowledge of the authors, IT2 NS-1 ANFIS has not been reported in the literature [1, 12].

One of the main contributions of this work is to implement an application of the IT2 NS-1 ANFIS using the hybrid REFIL-BP learning algorithm, capable of compensates for uncertain measurements.

3 Problem Solution

3.1 Using Hybrid RLS-BP Method in IT2 NS-1 ANFIS Training

The Table 1 shows the activities of the one pass learning algorithm of BP method. Both, IT2 TSK SFLS (BP) and IT2 TSK NSFLS-1 (BP) outputs are calculated during forward pass. During the backward pass, the error propagates backward and the antecedent and consequent parameters are estimated using only the BP method.

Table 1. One pass in learning procedure for IT2 TSK (BP)

	Forward Pass	Backward Pass
Antecedent Parameters	Fixed	BP
Consequent Parameters	Fixed	BP

The proposed hybrid algorithm (IT2 NS-1 ANFIS) uses RLS during forward pass for tuning of consequent parameters as well as the BP method for tuning of antecedent parameters, as shown in Table 2. It looks like Sugeno type-1 ANFIS [13, 14], which uses RLS-BP hybrid learning rule for type-1 FLS systems.

Table 2. Two passes in hybrid learning procedure for IT2 NS-1 ANFIS

	Forward Pass	Backward Pass
Antecedent Parameters	Fixed	BP
Consequent Parameters	RLS	Fixed

3.2 Adaptive BP Learning Algorithm

The training method is presented as in [1]: Given N input-output training data pairs, the training algorithm for E training epochs, should minimize the error function:

$$e^{(t)} = \frac{1}{2} [f_{IT2-FLS}(\mathbf{x}^{(t)}) - y^{(t)}]^2. \quad (1)$$

where e^t is the error function at time t , $f_{IT2-FLS}(\mathbf{x}^{(t)})$ is the output of the IT2 FLS using the input vector $\mathbf{x}^{(t)}$ from the input-output data pairs, and $y^{(t)}$ is the output from the input-output data pairs.

4 Application to Transfer Bar Surface Temperature Prediction

4.1 Hot Strip Mill

Because of the complexities and uncertainties involved in rolling operations, the development of mathematical theories has been largely restricted to two-dimensional models applicable to heat losing in flat rolling operations.

Fig. 1, shows a simplified diagram of a HSM, from the initial point of the process at the reheat furnace entry to its end at the coilers.

Besides the mechanical, electrical and electronic equipment, a big potential for ensuring good quality lies in the automation systems and the used control techniques. The most critical process in the HSM occurs in the Finishing Mill (FM). There are several mathematical model based systems for setting up the FM. There is a model-based set-up system [15] that calculates the FM working references needed to obtain gauge, width and temperature at the FM exit stands. It takes as inputs: FM exit target gage, target width and target temperature, steel grade, hardness ratio from slab chemistry, load distribution, gauge offset, temperature offset, roll diameters, load distribution, transfer bar gauge, transfer bar width and transfer bar temperature entry.

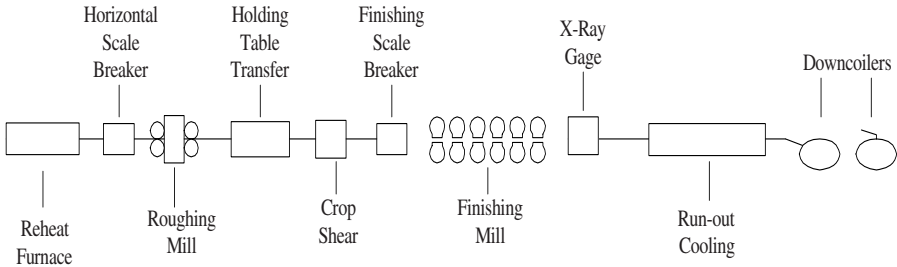


Fig. 1. Typical hot strip mill

The errors in the gauge of the transfer bar are absorbed in the first two FM stands and therefore have a little effect on the target exit gauge. It is very important for the model to know the FM entry temperature accurately. A temperature error will propagate through the entire FM.

4.2 Design of the IT2 NS-1 ANFIS

The architecture of the IT2 NS-1 ANFIS was established in such way that its parameters are continuously optimized. The number of rule-antecedents was fixed to two; one for the Roughing Mill (RM) exit surface temperature and one for transfer bar head traveling time. Each antecedent-input space was divided in three fuzzy sets (FSs), fixing the number of rules to nine. Gaussian primary membership functions (MFs) of uncertain means were chosen for the antecedents. Each rule of the each IT2 NS-1 ANFIS is characterized by six antecedent MFs parameters (two for left-hand and right-hand bounds of the mean and one for standard deviation, for each of the two antecedent Gaussian MFs) and six consequent parameters (one for left-hand and one for right-hand end points of each of the three consequent type-1 FSs), giving a total of twelve parameters per rule. Each input value has one standard deviation parameter, giving two additional parameters.

4.3 Noisy Input-Output Training Data Pairs

From an industrial HSM, noisy input-output pairs of three different product types were collected and used as training and checking data. The inputs are the noisy measured RM exit surface temperature and the measured RM exit to SB entry transfer bar traveling time. The output is the noisy measured SB entry surface temperature.

4.4 Fuzzy Rule Base

The IT2 NS-1 ANFIS fuzzy rule base consists of a set of IF-THEN rules that represents the model of the system. The IT2 ANFIS system has two inputs $x_1 \in X_1$, $x_2 \in X_2$ and one output $y \in Y$. The rule base has $M = 9$ rules of the form:

$$R^i : \text{IF } x_1 \text{ is } \tilde{F}_1^i \text{ and } x_2 \text{ is } \tilde{F}_2^i, \text{ THEN } Y^i = C_0^i + C_1^i x_1 + C_2^i x_2. \quad (2)$$

where Y^i the output of the i th rule is a fuzzy type-1 set, and the parameters C_j^i , with $i = 1, 2, 3, \dots, 9$ and $j = 0, 1, 2$, are the consequent type-1 FSs.

4.5 Input Membership Function

The primary MFs for each input of the IT2 NS-1 ANFIS are Gaussians of the form:

$$\mu_{X_k}(x_k) = \exp\left[-\frac{1}{2}\left[\frac{x_k - x_k'}{\sigma_{X_k}}\right]^2\right]. \quad (3)$$

where: $k = 1, 2$ (the number of type-2 non-singleton inputs), $\mu_{X_k}(x_k)$ is centered at $x_k = x_k'$ and σ_{X_k} is the standard deviation. The standard deviation of the RM exit surface temperature measurement, σ_{X_1} , was initially set to 13.0 °C and the standard deviation head end traveling time measurement, σ_{X_2} , was initially set to 2.41 s. The uncertainty of the input data was modeled as stationary additive noise using type-1 FSs.

4.6 Antecedent Membership Functions

The primary MFs for each antecedent are FSs described by Gaussian with uncertain means:

$$\mu_k^i(x_k) = \exp\left[-\frac{1}{2}\left[\frac{x_k - m_k^i}{\sigma_k^i}\right]^2\right]. \quad (4)$$

where $m_k^i \in [m_{k1}^i, m_{k2}^i]$ is the uncertain mean, with $k = 1, 2$ (the number of antecedents) and $i = 1, 2, \dots, 9$ (the number of M rules), and σ_k^i is the standard deviation. The means of the antecedent fuzzy sets are uniformly distributed over the entire input space.

Table 3 shows the calculated interval values of uncertainty of x_1 input, where $[m_{11}, m_{12}]$ is the uncertain mean and σ_1 is the standard deviation.

Table 4 shows the calculated interval values of uncertainty of x_2 input, where $[m_{21}, m_{22}]$ is the uncertain mean and σ_2 is the standard deviation for all the 9 rules.

Table 3. Intervals of uncertainty x_1 input

	m_{11}	m_{12}	σ_1
	°C	°C	°C
1	950	952	60
2	1016	1018	60
3	1080	1082	60

Table 4. Intervals of uncertainty of x_2 input

Product Type	m_{21}	m_{22}	σ_2
	s	s	s
1	32	34	10
2	42	44	10
3	56	58	10

The standard deviation of temperature noise σ_{n1} was initially set to $1^\circ C$ and the standard deviation of time noise σ_{n2} was set to 1 s.

4.7 Consequent Membership Functions

Each consequent is an interval type-1 FS with $Y^i = [y_l^i, y_r^i]$ where

$$y_l^i = \sum_{j=1}^P c_j^i x_j + c_0^i - \sum_{j=1}^P |x_j| s_j^i - s_0^i \quad (5)$$

and

$$y_r^i = \sum_{j=1}^P c_j^i x_j + c_0^i + \sum_{j=1}^P |x_j| s_j^i + s_0^i \quad (6)$$

where c_j^i denotes the center (mean) of C_j^i and s_j^i denotes the spread of C_j^i , with $i = 1, 2, 3, \dots, 9$ and $j = 0, 1, 2$. Then y_l^i and y_r^i are the consequent parameters. When only the input-output data training pairs $(x^{(1)} : y^{(1)}), \dots, (x^{(N)} : y^{(N)})$ are available and there is not data information about the consequents, the initial values for the centroid parameters c_j^i and s_j^i can be chosen arbitrarily in the output space [11]. In this work the initial values of c_j^i were set equal to 0.001 and the initial values of s_j^i equal to 0.0001, for $i = 1, 2, 3, \dots, 9$ and $j = 0, 1, 2$.

4.8 Results

The IT2 NS-1 ANFIS (RLS-BP) system was trained and used to predict the SB entry temperature, applying the RM exit measured transfer bar surface temperature and RM exit to SB entry zone traveling time as inputs. We ran fifteen epochs of training; one hundred and ten parameters were tuned using eighty seven, sixty-eight and twenty-eight input-output training data pairs per epoch, for type A, type B and type C products respectively.

The performance evaluation for the hybrid IT2 NS-1 ANFIS system was based on root mean-squared error (RMSE) benchmarking criteria as in [1].

Fig. 2 shows the RMSEs of the two IT2 TSK NS-1 systems and the base line IT2 TSK SFLS (BP) for fifty epochs' of training for the case for type C products. Observe

that from epoch 1 to 4 the hybrid IT2 NS-1 ANFIS (RLS-BP) has better performance than both: the IT2 SFLS (BP) and the IT2 NSFLS-1 (BP). From epoch 1 to 4 the RMSE of the IT2 NSFLS-1 has an oscillation, meaning that it is very sensitive to its learning parameters values. At epoch 5, it reaches its minimum RMSE and is stable for the rest of training. The proposed IT2 NS-1 ANFIS system has the best performance and stability after only one epoch of training.

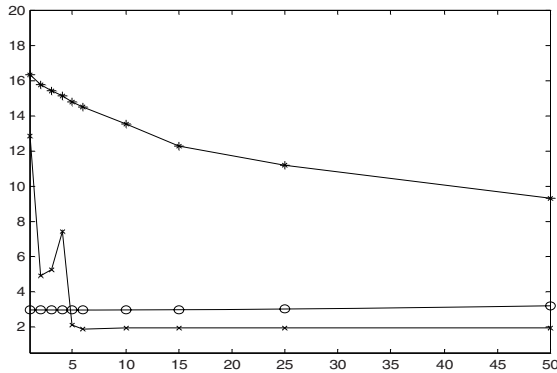


Fig. 2. (*) RMSE_{TSK 2, SFLS (BP)} (+) RMSE_{TSK 2, NSFLS-1 (BP)} (o) RMSE_{TSK 2, NSFLS-1 (REFIL-BP)}

5 Conclusions

An IT2 NS-1 ANFIS using the hybrid RLS-BP training method was tested and compared for predicting the surface temperature of the transfer bar at SB entry. The antecedent MFs and consequent centroids of the IT2 NS-1 ANFIS absorbed the uncertainty introduced by all the factors: the antecedent and consequent initially values, the noisy temperature measurements, and the inaccurate traveling time estimation. The non-singleton type-1 fuzzy inputs are able to compensate the uncertain measurements, expanding the applicability of IT2 NS-1 ANFIS systems.

It has been shown that the proposed IT2 NS-1 ANFIS system can be applied in modeling and control of the steel coil temperature. It has also been envisaged its application in any uncertain and non-linear system prediction and control. Especially in those applications where there is only one chance of training.

References

1. Mendel, J. M. : Uncertain Rule Based Fuzzy Logic Systems: Introduction and New Directions, Upper Saddle River, NJ, Prentice-Hall, (2001)
2. Mendez, G., Cavazos, A., Leduc, L. , Soto, R.: Hot Strip Mill Temperature Prediction Using Hybrid Learning Interval Singleton Type-2 FLS, Proceedings of the IASTED International Conference on Modeling and Simulation, Palm Springs, February (2003), pp. 380-385

3. Mendez, G., Cavazos, A., Leduc, L., Soto, R.: Modeling of a Hot Strip Mill Temperature Using Hybrid Learning for Interval Type-1 and Type-2 Non-Singleton Type-2 FLS, Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Benalmádena, Spain, September (2003), pp. 529-533
4. Mendez, G., Juarez, I.I: Orthogonal-Back Propagation Hybrid Learning Algorithm for Interval Type-1 Non-Singleton Type-2 Fuzzy Logic Systems, WSEAS Transactions on Systems, Issue 3, Vol. 4, March 2005, ISSN 1109-2777
5. Mendez, G., Castillo, O.: Interval Type-2 TSK Fuzzy Logic Systems Using Hybrid Learning Algorithm, FUZZ-IEEE 2005 The international Conference on Fuzzy Systems, Reno Nevada, USA, (2005), pp 230-235
6. Lee, D. Y., Cho, H. S.: Neural Network Approach to the Control of the Plate Width in Hot Plate Mills, International Joint Conference on Neural Networks, (1999), Vol. 5, pp. 3391-3396
7. Taylor, B. N., Kuyatt, C. E.: Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results, September (1994), NIST Technical Note 1297
8. Wang, L-X.: Fuzzy Systems are Universal Approximators, Proceedings of the IEEE Conf. On Fuzzy Systems, San Diego, (1992), pp. 1163-1170
9. Wang, L-X., Mendel, J. M.: Back-Propagation Fuzzy Systems as Nonlinear Dynamic System Identifiers, Proceedings of the IEEE Conf. On Fuzzy Systems, San Diego, CA. March (1992), pp. 1409-1418
10. Wang, L-X.: Fuzzy Systems are Universal Approximators, Proceedings of the IEEE Conf. On Fuzzy Systems, San Diego, (1992), pp. 1163-1170
11. Wang, L-X.: A Course in Fuzzy Systems and Control, Upper Saddle River, NJ: Prentice Hall PTR, (1997)
12. Liang, Q. J., Mendel, J. M.: Interval type-2 fuzzy logic systems: Theory and design, Trans. Fuzzy Syst., Vol. 8, Oct. (2000), pp. 535-550
13. Jang, J. -S. R., Sun, C. -T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Upper Saddle River, NJ: Prentice-Hall, (1997)
14. Jang, J. -S. R., Sun, C. -T.: Neuro-Fuzzy Modeling and Control, The Proceedings of the IEEE, Vol. 3, March (1995), pp. 378-406
15. GE Models, Users reference, Vol. 1, Roanoke VA, (1993)

A Vision-Based Hybrid Classifier for Weeds Detection in Precision Agriculture Through the Bayesian and Fuzzy k-Means Paradigms

Alberto Tellaache¹, Xavier-P. BurgosArtizzu², Gonzalo Pajares³, and Angela Ribeiro²

¹ Dept. Informática y Automática, E.T.S. Informática- UNED, 28040 Madrid, Spain
tellaache@euskalnet.net

² Instituto de Automática Industrial. CSIC. Arganda del Rey. Madrid. Spain
{xpburgos, angela}@iai.csic.es

³ Dept. Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense, 28040 Madrid, Spain
pajares@dacya.ucm.es

Abstract. One objective in Precision Agriculture is to minimize the volume of herbicides that are applied to the fields through the use of site-specific weed management systems. This paper outlines an automatic computer vision system for the detection and differential spraying of *Avena sterilis*, a noxious weed growing in cereal crops. With such purpose we have designed a hybrid decision making system based on the Bayesian and Fuzzy k-Means (FkM) classifiers, where the a priori probability required by the Bayes framework is supplied by the FkM. This makes the main finding of this paper. The method performance is compared against other available strategies.

Keywords: Bayesian classifier, Fuzzy k-Means, precision agriculture, weeds detection.

1 Introduction

Nowadays, there is a clear tendency of reducing the use of chemicals in agriculture. Numerous technologies have been developed trying to obtain safer agricultural products and lower environmental impacts. The concept of Precision Agriculture provides a valuable framework to achieve this goal [1],[2].

Within this general framework, weeds can be managed site-specifically using available geospatial and information technologies [3]. Initial efforts to detect weed seedlings by machine vision were focused on geometrical measurements such as shape factor, aspect ratio, length / area, etc. [4]. Later, color images were successfully used to detect weeds and other types of pests [5]. Weed coverage and weed patchiness, based on digital images, using a fuzzy algorithm for planning site-specific herbicide applications have been also estimated in [6]. Different approaches have used spectral colour indices to distinguish plant material from the background [3], [7], [8]. *Avena sterilis L.*, (“winter wild oat”) is one of the most widely distributed and abundant weeds of cereals in Spain and other regions with Mediterranean climate, causing substantial losses in these crops [9], [10]. The main problem concerning its

detection is that, at the time of herbicide treatment, *A. sterilis* shape, color and texture are undistinguishable from those of the crop (barley or wheat). Due to this similarity, none of the detection methods mentioned previously are applicable to this case.

Our work was based on the hypothesis that a high density of green color in the inter row areas (where the crop is not present) after postemergence herbicides have been applied for broadleaf weed control, indicates that these zones are infested by high densities of *A. sterilis*, which can be differentially sprayed.

Although there are several approaches to compute shapes or areas as attributes [11], [12], [13], the computation of unary attributes describing each isolated patch form is not appropriated in this particular case due to the irregular distribution and shapes of weed patches. Because of this, we decided to define binary relations among the weed patches and the crop rows. In order to decide whether the selected area was to be sprayed or not, the Bayesian and Fuzzy k-Means (FkM) frameworks are combined for making the decision. This makes the main finding of this work.

This paper is organized as follows. In section 2 the image segmentation process is described. In section 3 the combined decision making strategy is proposed. The performance of this approach is described in section 4. Finally in section 5 the conclusions are presented.

2 Image Segmentation Process

Adquisition. The images used for this study were captured in an experimental field of barley on La Poveda Research Station, Arganda del Rey, Madrid. The area of the field was 1.7 ha. Images were taken on two different dates on April 2003. Row spacing was 0.36 m. Digital images were captured with a Sony DCR PC110E camera. The area covered by the piece of each image which is to be processed was approximately $1.9 \times 15 \text{ m}^2$ and the resolution of an image was 1152×864 pixels.

Binarization. In precision agriculture several techniques have been proposed in order to isolate weeds and crops [8,12,13,14]. A thresholding approach applied to the gray level image coming from the RGB original one is applied in [8],

$$T(i, j) = rR(i, j) + gG(i, j) + bB(i, j). \quad (1)$$

where r , g and b are set to: $r = -1$, $g = 2$ and $b = -1$; if $T(i, j) \leq 0$ then $T(i, j) = 0$; if $T(i, j) \geq 255$ then $T(i, j) = 255$, i.e. the grey level output values range in $[0, 255]$.

The resulting binarized image is morphologically opened as in [13] in order to remove spurious white pixels and to smooth white contours.

Crop lines detection and grid cell partition. In the opened binary image, plant material from both weeds and crops is white and the rest, coming from soil, stones and residual is black. The furrows are wide lines with weeds patches. We apply the Hough transform as a robust technique in the normal parameter space (polar coordinates) for describing the furrows [15],[16]. This gives straight-line equations for accumulations greater than 100 pixels. Two lines are fused in a unique line, if the differences between the polar angles and distances are less than the two respective thresholds ε_1 and ε_2 , set to 5 and 10 respectively in this paper.

By drawing horizontal lines vertically spaced in steps of 50 pixels and taking the computed crop lines, the image is split in cells. The basic unit to be analyzed is the cell. Due to the perspective transformation the shape and size of the cells differ among them along the image, Fig. 1(a).

Attribute extraction. In [4], [12] are used topological properties (area, invariant moments, etc.); colour (variance, skewness of the intensity histogram, etc.) for weeds identification. In our experiments they are not applicable because the weeds have irregular forms and similar spectral signature than the crop cereal. Moreover the cereal grows uncontrolled in the field. This means that white patches in soil areas between crops should be weeds and the surrounding crop areas are probably affected by weed seeds. To solve these problems, we use attributes that are independent of the weeds distribution and from the size and shape of the cells. Two area-based attributes are computed and embedded as the components of an area-vector \mathbf{x}_i , for the cell i , this vector is: $\mathbf{x}_i = \{x_{i1}, x_{i2}\}$. Let m the total number of connected regions in the cell i (i.e. the number of white pixels in the cell) and A_{ij} the area of the j -th region. A_{ic} is the total area of the cell. A_{iL} and A_{iR} the areas for the left and right regions respectively. These regions correspond to crop areas and after experimentation we assume that they cover $1/6$ each one of the total area of the cell. Based on the area measurements the following coverage values are computed: crop (C_{ic}), weed (C_{iw}) and soil (C_{is}),

$$C_{ic} = A_{iL} + A_{iR}; C_{iw} = \sum_{j=1}^m A_{ij} - C_{ic} \text{ and } C_{is} = A_{ic} - (C_{ic} + C_{iw}) \quad (2)$$

From equation (2) we compute the components for the area-vector \mathbf{x}_i as follows,

$$x_{i1} = \frac{C_{iw}}{A_{ic}} \quad \text{and} \quad x_{i2} = \frac{C_{iw}}{C_{ic}} \left(1 - \frac{C_{is}}{A_{ic}} \right) \quad (3)$$

The component x_{i1} is defined as the weed coverage rate in [14] and x_{i2} can be the weed pressure defined in [8]; x_{i1} and x_{i2} range in $[0, \frac{4}{5}]$ and $[0, 5.13]$ respectively, obtained by giving limit values in (3). They are linearly mapped to range in $[0, 1]$ so that its contribution is similar.

3 Hybrid Classifier: Bayesian and Fuzzy k-Means

Given \mathbf{x}_i , representing the attributes of the cell i , the problem is to make a decision about if the cell must be or not sprayed. This is carried out through the Bayesian framework. Indeed, the Bayes rule computes the *posterior* probability,

$$P(w_c | \mathbf{x}_i) = \frac{p(\mathbf{x}_i | w_c)P(w_c)}{p(\mathbf{x}_i)}, \quad c = y, n \quad (4)$$

where w_y, w_n represent classes of cells to be and not to be sprayed respectively. $P(w_c)$ is the *a priori* probability that the cell belongs to the class w_c ; $p(\mathbf{x}_i)$ is the twofold mixture density distribution. So, the decision is made through the equation (5),

$$i \in w_y \text{ if } p(\mathbf{x}_i | w_y)P(w_y) > p(\mathbf{x}_i | w_n)P(w_n); \text{ otherwise } i \in w_n \quad (5)$$

The main problems to be solved are the estimations of both the class-conditional probability density functions $p(\mathbf{x}_i | w_y)$, $p(\mathbf{x}_i | w_n)$ and the *a priori* probabilities $P(w_y)$, $P(w_n)$. The first ones are obtained via the well-known parametric Bayesian estimator assuming a Gaussian distribution [17]. The second ones are derived from a decision function provided by the FkM approach [17], [18], [19], [20]. This combination makes the main finding of this paper.

Let $X^y = \{\mathbf{x}_1^y, \mathbf{x}_2^y, \dots, \mathbf{x}_{n_y}^y\}$ and $X^n = \{\mathbf{x}_1^n, \mathbf{x}_2^n, \dots, \mathbf{x}_{n_n}^n\}$ two subsets of attribute vectors representing the cells to be and not to be sprayed respectively, with $X = X^y \cup X^n$; the number of cells belonging to each subset is n_y and n_n respectively, i.e. $N = n_y + n_n$. Initially both sets are selected under the supervision of the technical consultants and farmers.

Prior probability by FkM: given the number of clusters c and following [17], [18], [19], [20] the FkM algorithm is based on the minimization of the objective function J ,

$$J(U; \mathbf{v}) = \sum_{i=1}^N \sum_{j=1}^c \mu_{ij}^m d_{ij}^2. \quad (6)$$

subject to $\mu_{ij} \in [0, 1]; \sum_{j=1}^c \mu_{ij} = 1; \sum_{i=1}^N \mu_{ij} < N; 1 \leq j \leq c, 1 \leq i \leq N.$ (7)

In our approach the clusters are w_y, w_n , i.e. $c = 2$ and $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2\}$. These cluster centers are to be updated. The $N \times c$ matrix $U = [\mu_{ij}]$ contains the membership grade of pattern i with cluster j ; $d_{ij}^2 \equiv d^2(\mathbf{x}_i, \mathbf{v}_j)$ is also the squared Euclidean distance. The number m is called the exponent weight [19]. In order to minimize the objective function in (6), the cluster centers and membership grades are chosen so that high memberships occur for samples close to the corresponding cluster center. The higher the value of m , the less those samples whose memberships are low contribute to the objective function. Consequently, such samples tend to be ignored in determining the cluster centers and membership degrees [19], [20].

The original FkM computes for each \mathbf{x}_i at the iteration k its membership grade and updates the cluster centers according to equation (8),

$$\mu_{ij}(k) = \left(\sum_{r=1}^c (d_{ij}(k)/d_{ir}(k))^{2/(m-1)} \right)^{-1}; \quad \mathbf{v}_j(k+1) = \frac{\sum_{i=1}^N \mu_{ij}^m(k) \mathbf{x}_i}{\sum_{i=1}^N \mu_{ij}^m(k)} \quad (8)$$

The stopping criterion of the iteration process is achieved when $\|\mu_{ij}(k+1) - \mu_{ij}(k)\| < \varepsilon \quad \forall ij$ or a maximum number of iterations is reached.

Once the cluster centers \mathbf{v}_j are estimated, we can compute the membership degrees for \mathbf{x}_i through (8) obtaining μ_{iy} ($\mu_{in} = 1 - \mu_{iy}$).

We identify the *a priori* probabilities as $P(w_y) = \mu_{iy}$ and $P(w_n) = \mu_{in}$. Now, the decision through the equation (5) can be made.

4 Comparative Analysis and Performance Evaluation

4.1 Design of a Test Strategy

In order to assess the validity of the proposed hybrid approach, we have designed a test strategy with the following three goals:

- 1) to compare the performance of the attributes used
- 2) to verify the performance of the proposed combined approach against single strategies
- 3) to compare the performance with respect the number of images processed.

In our approach we use two attributes, x_{i1} and x_{i2} . As described in the section 2, x_{i1} and x_{i2} are used individually in [8] and [14] respectively.

Performance of the attributes. In order to compare this performance we identify three tests: Test 1, which uses both attributes x_{i1} and x_{i2} ; Test 2, which uses only the attribute x_{i1} and Test 3, using only x_{i2} . Hence, this allows the testing of our approach against the two methods given in [8] and [14].

Performance of the combination. Our proposed strategy (Bayes and FkM) is compared against the single FkM and Bayesian methods. When Bayes is applied individually, we ignore prior knowledge. Hence, the prior probability in the equation (5) is fixed to 0.5, which is the intermediate probability value in the range [0,1]. As we will see later, the combined strategy for Test 1 performs better than the single ones, as expected. Hence, we use only the combined methods for Tests 2 and 3.

Performance for the number of images. The probability density function and the membership degrees for the Bayesian and Fuzzy approaches are obtained taking into account the number of cells (patterns), i.e. the number of images processed. In order to compare the performance against the number of patterns, the Tests are applied in four STEPs (0 to 3). At each STEP, a new set of 30 images with 10 cells per image are (300 patterns) are added to the previous ones. So, we estimate the probability density function and compute the membership degrees with the number of cells available at each STEP, i.e. with 300, 600, 900 and 1200 for STEPs 1, 2 3 and 4 respectively.

Initially, the set of 30 images processed in STEP 0 is selected randomly, where the number of cells classified as candidate to be sprayed is 48 (16% of 300). With this set we obtain the initial supervised estimation. Then, for STEPs 1 to 3 the cells classified as belonging to the class w_y / w_n are added to the corresponding subsets X^y / X^n respectively, from which we estimate the new probability density function and membership degrees for Bayes and FkM respectively.

4.2 Measurements for Validation

The results obtained for each strategy are checked by technical consultants and farmers, i.e. under an expert human criterion. The different *Tests* analysed are based on the following values:

True Spraying (TS): i.e. number of cells correctly identified to be sprayed.

True No Spraying (TN): i.e. number of cells that do not require spraying correctly detected.

False Spraying (FS): i.e. number of cells that do not require spraying but identified as cells to be sprayed.

False No Spraying (FN): i.e. number of cells requiring spraying that they are identified by the method as cells that do not require spraying.

Traditionally, from these four quantities several measures have been used for classification. In [21] some of them are analysed. The best ones are those combining the above four values. We have chosen the correct classification percentage (CCP) based on the analysis reported in [21], which is defined as follows,

$$CCP = \frac{TS + TN}{TS + FS + TN + FN} \quad (9)$$

Figure 1(a) displays a representative original image of a cereal crop field to be sprayed. In (b) the segmented image with the set of cells labeled as F identified as cells to be sprayed.

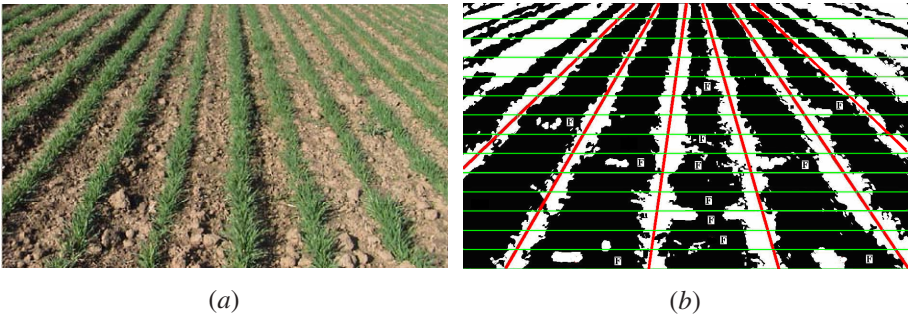


Fig. 1. (a) original image; (b) segmented image

4.3 Analysis of Results

Table 1 displays the results in terms of the correct classification for the three STEPs. Our proposed strategy (Bayes and FkM) is verified for Tests 1, 2 and 3. The Bayes and FkM are individually verified for Test 1. For each STEP the CCP values are displayed under the CCP columns. Larger score values indicate better performance. The percentages of cells classified as cells to be sprayed is also displayed (%).

From results in table 1, one can infer that the best performance is achieved by our proposed approach at each STEP, i.e. through the combined strategy as expected according to the reports found in the literature [19].

Hence, the fusion of classifiers becomes a suitable approach, which achieves better results than the single methods. During the Test 1 we obtain better results than those obtained in Test 2 and Test 3, i.e. the two attributes used in Test 1 perform better than the single attribute used in Test 2 and 3.

Also, the performance improves as the number of pattern samples increases. This is applicable for all strategies. This means that the number of samples is important.

Additionally, we can see that less percentage of cells classified as units to be sprayed is achieved by our approach.

This best percentage and its best performance imply that the amount of herbicide to be applied is reduced by using the proposed hybrid method achieving an important saving in cost and ground pollution.

Table 1. CCP score values for the different strategies in the three Tests and STEPs

	STEP 1		STEP 2		STEP 3	
	CCP	%	CCP	%	CCP	%
Test 1 (Bayes + FkM)	80	23	86	21	91	18
Test 1 (Bayes)	75	25	81	24	87	22
Test 1 (FkM)	72	28	79	26	85	23
Test 2 (Bayes + FkM)	66	32	70	32	70	32
Test 3 (Bayes + FkM)	70	30	70	31	71	31

5 Conclusions

We propose a new automatic process for detecting weeds in cereal crops. The weeds and the crops have similar spectral signatures and textures. This represents an important problem which is addressed under two strategies: segmentation and decision making.

We apply a segmentation process which combines different techniques. This implies that the image is ready for making the decision about its spraying.

The decision is based on the fusion of two well-tested classifiers (Bayes and FkM) under the Bayesian framework making the most important contribution of this paper. Additionally, the strength of the probability allows us to determine the amount of herbicide to be applied, making another important finding against methods where the decision is discrete (yes or not).

The combination of the weed coverage and weed pressure attributes improves the performance of the approach as compared with the use of these attributes separately. An important issue that is to be analysed in future works is the robustness of the proposed approach against illumination variability. This is because the robot-tractor where the system is installed goes in a direction and its opposite, i.e. the illumination coming from the natural environment varies. This could be carried by applying homomorphic filtering, so that the reflectance component is separated from the illumination [15].

Acknowledgments. Part of the work has been performed under project AGL-2005-06180-C03-03. Alberto Tellaeche is with Tekniker foundation in Eibar, Gipuzkoa, Spain, working in Computer Vision tasks and intelligent systems.

References

1. Stafford, J.V.: The role of Technology in the Emergence and Current Status of Precision Agriculture. In Handbook of precision agriculture (Srinivasan, A. ed.). Food Products Press. New York. (2006) 19-56
2. Zhang, A., Wang, M. and Wang, N.: Precision Agriculture-a Worldwide Overview. Computers and Electronics in Agriculture, 36 (2002) 113-132
3. Gerhards, R. and Oebel, H.: Practical Experiences with a System for Site-specific Weed Control in Arable Crops Using Real-time Image Analysis and GPS-controlled Patch Spraying. Weed Research, 46 (2006) 185-193
4. Pérez, A.J., López, F., Benlloch, J.V. and Christensen, S.: Colour and Shape Analysis Techniques for Weed Detection in Cereal Fields. Computers and Electronics in Agriculture, 25 (2000) 197-212
5. Sgaard, H.T. and Olsen, H.J.: Determination of Crop Rows by Image Analysis without Segmentation. Computers and Electronics in Agriculture, 38 (2003) 141-158
6. Yang, C.C., Prasher, S.O. Landry, J.A. and Ramaswamy, H.S.: Development of an Image Processing System and a Fuzzy Algorithm for Site-specific Herbicide Applications. Precision Agriculture, 4 (2003) 5-18
7. Thorp, K.R. and Tian, L.F.: A Review on Remote Sensing of Weeds in Agriculture. Precision Agriculture, 5, (2004) 477-508
8. Ribeiro A., Fernández-Quintanilla, C., Barroso, J., Garcá-Alegre.M.C.: Development of an Image Analysis System for Estimation of Weed. In *Proc. 5th European Conf. On Precision Agriculture (5ECPA)*, 2005 pp. 169-174
9. Barroso, J., Fernández-Quintanilla, C., Ruiz, C., Hernaiz, P. and Rew, L.J.: Spatial Stability of Avena Sterilis ssp. Ludoviciana Populations under Annual Applications of Low Rates of Imazamethbenz. Weed Research, 44 (2004) 178-186
10. Radics, L., Glemnitz, M., Hoffmann, J. and Czimer, G.: Composition of Weed Floras in Different Agricultural Management Systems within the European Climatic Gradient. In *Proc. 6th European Weed Research Society (EWRS). Workshop on Physical and Cultural Weed Control*, Lillehammer, Norway, (2004) 58-64
11. Aitkenhead, M.J., Dalgetty, I.A., Mullins, C.E., McDonald, A.J.S. and Strachan, N.J.C.: Weed and Crop Discrimination Using Image Analysis and Artificial Intelligence Methods. Computers and Electronics in Agriculture, 39 (2003) 157-171
12. Granitto, P.M., Verdes, P.F. and Ceccatto, H.A.: Large-scale Investigation of Weed Seed Identification by Machine Vision. Computers and Electronics in Agriculture, 47 (2005) 15-24
13. Onyango, C.M. and Marchant, J.A.: Segmentation of Row Crop Plants from Weeds Using Colour and Morphology. Computers and Electronics in Agriculture, 39, (2003) 141-155
14. Tian, L.F. and Slaughter, D.C.: Environmentally Adaptive Segmentation Algorithm for Outdoor Image Segmentation. Computers and Electronics in Agriculture, 21 (1998) 153-168
15. Gonzalez, R.C., Woods, R.E. and Eddins, S.L.: Digital Image Processing using Matlab. Prentice Hall, New York (2004)
16. Astrand, B. and Baerveldt, A.J.: An Agricultural Mobile Robot with Vision-Based Perception for Mechanical Weed Control. Autonomous Robots, 13, (2002) 21-35
17. Duda, R.O., Hart, P.E. and Stork, D.G.: Pattern Classification, Jhon Willey and Sons, New York (2001)
18. Zimmermann, H.J.: Fuzzy Set Theory and its Applications, Kluwer Academic Publishers, Norwell (1991)
19. Sneath, P., Sokal, R.: Numerical Taxonomy: the principle and practice of numerical classification, W.H. Freeman, San Francisco (1973).

Development of Multi-output Neural Networks for Data Integration – A Case Study

Paul Trundle¹, Daniel Neagu¹, Marian Craciun², and Qasim Chaudhry³

¹ Department of Computing, School of Informatics, University of Bradford, Richmond Road, Bradford, West Yorkshire, BD7 1DP, United Kingdom
{p.r.trundle, d.neagu}@bradford.ac.uk

² Department of Computer Science and Engineering, University "Dunarea de Jos" of Galati, G309, Stiintei 2, Galati 800146, Romania
marian.craciun@ugal.ro

³ Central Science Laboratory, Sand Hutton, York, YO41 1LZ
q.chaudhry@csl.gov.uk

Abstract. Despite the wide variety of algorithms that exist to build predictive models, it can still be difficult to make accurate predictions for unknown values for certain types of data. New and innovative techniques are needed to overcome the problems underlying these difficulties for poor quality data, or data with a lack of available training cases. In this paper the authors propose a technique for integrating data from related datasets with the aim of improving the accuracy of predictions using Artificial Neural Networks. An overall improvement in the prediction power of models was shown when using the integration algorithm, when compared to models constructed using non-integrated data.

Keywords: Data Integration, Multi-input Multi-output Connectionist Models, Related Data.

1 Introduction

Advances in the fields of Data Mining and Machine Learning have given us a wealth of techniques with which to model and extract knowledge from collections of data. Algorithms exist to perform classification or regression on a wide variety of data from many different sources. There is an extensive choice of methods to use when modelling a particular collection of data, such as Artificial Neural Networks (ANNs) [1], Support Vector Machines [2], Decision Trees [3], Rule Induction [4], Instance-Based Learners [5], Linear and Non-Linear Regression [6] and Bayesian Networks [7], amongst many others. Despite the large amount of diverse modelling algorithms in existence it can still be difficult to make accurate predictions on unusual or poor quality data using these standard techniques. New and innovative approaches are needed to allow high quality predictions to be made on these problematic data collections.

In order to create powerful and robust algorithms for modelling it is sometimes necessary to deviate from standard Machine Learning paradigms and investigate how particular characteristics of a dataset, or collection of datasets, can be exploited to maximise the accuracy of predictions. In this paper the authors discuss the potential

for utilising data from multiple related datasets, i.e. datasets that contain non-identical instances (records) and non-identical attributes (fields) but which have some meaningful relationship and a mathematical correlation between their target attribute (the value that we wish to make predictions for on new instances). Examples of potentially related datasets include: national census data from different years, where some relationship between the responses from individuals from different years is likely to exist; text processing, where pattern of letters and words are likely to be repeated in many different text sources; genomic data, where similarity in the genetic makeup of different species is common; and predictive toxicology data where different datasets contain values describing the relative toxicity for various chemicals on different animal or plant species. A technique for constructing multi-output connectionist models, in the form of ANNs [1], that can integrate data from these related datasets is proposed and explained with examples using real world data.

The usefulness of ANNs for a wide variety of applications is well known, with practical applications in diverse fields such as: protein structure prediction [8]; flight control [9]; economic forecasting [10]; computer system misuse and anomaly detection [11], and face detection [12] amongst innumerable others.

The paper continues in section 2 where the technique described above is formally defined and explained. Section 3 shows the technique in use with real world data and examples. Section 4 draws conclusions from the results and discusses applications of the technique.

2 Methods

This section further describes the technique introduced in the previous section, and explains the methods behind its use. The integration of data from related datasets and the process by which multi-output neural networks can be used for this purpose is described below.

The possibility of using multiple related datasets together to potentially enhance the prediction accuracy and reliability may be of great use to model-builders in fields where related but distinct sets of data exist. The proposed technique attempts to utilise the relationship between the related datasets by discarding the traditional idea of a single model for a single dataset, and instead tries to build combinatorial models that make predictions for multiple dataset output values at the same time. The aim is to allow the model to extract and use knowledge from more than one dataset, so that predictions are based on a maximal amount of information. Increasing the amount of data available during training, as well as providing a more diverse source of information on which a model building algorithm can train, may allow increased learning to take place and a higher level of prediction quality to be achieved. Integrating data in this way should be used cautiously, and reasonable scientific justification that the data is sufficiently related and suitable for use with this technique should be determined before it is used.

Suppose we have a number of related datasets that we wish to integrate, defined as $d_1 \dots d_n$, where n is the total number of related datasets, and each dataset has an associated numerical output value (target for prediction) defined as t_n . Each one also has a set of numerical inputs, defined as $i_{1n} \dots i_{mn}$ where m is the total number of inputs in the given dataset and n is the number of the dataset that the inputs belong to. Traditional modelling techniques relate the input values of a single dataset to the output values, using some process or formula, defined here as f :

$$t_n = f(i_{1n}, i_{2n}, \dots, i_{mn}) \quad (1)$$

The technique proposed by the authors aims to take advantage of the relationship between the input and output values in each of the related datasets by reducing multiple prediction processes into a single, combined process that can make predictions for many or all of the output values for the various datasets. The user must first decide which datasets are to be integrated, and thus which combination of outputs the combined model will make predictions for. Suppose we wish to integrate all of the datasets $d_1 \dots d_n$, we begin by forming a combined set of inputs, defined as I , where every input from all datasets is included but identical sets of inputs are used only once, achieved by taking the union of all possible inputs:

$$I = \{i_{11}, i_{21}, \dots, i_{m1}\} \cup \{i_{12}, i_{22}, \dots, i_{m2}\} \cup \dots \cup \{i_{1n}, i_{2n}, \dots, i_{mn}\} \quad (2)$$

Note that there may be some redundancy between the inputs from all datasets, so the union process may result in a reduction in the total number of inputs as identical sets of values are removed. With this combined set of input values we can build a model to predict the output values for multiple datasets at the same time, hopefully allowing learning to take place in ways that were not possible when using inputs from a single source. The new function that makes these multiple predictions is defined as F :

$$\{t_1, t_2, \dots, t_n\} = F(I) \quad (3)$$

The combinatorial model is then used to make predictions for output values for every dataset that was included in the integration process. An increased computation time is balanced by the ability to use a single model for predicting multiple values simultaneously, as well as the potential for increased prediction power if the model is able to successfully integrate knowledge from the different datasets. The authors suggest the use of ANNs with this technique, as their architecture, and the algorithms used to create and train them, naturally lend themselves to the technique.

Results from the application of this algorithm to a real world dataset are reported in section 3.

3 Results

This section presents the results from the application of the algorithm described in section 2 on a real world dataset. The data comes from the field of Predictive

Toxicology, which is the study of creating models in order to predict the relative magnitude of toxic effects of chemicals on human, animal and environmental health, based on properties of their chemical structure.

The 2 datasets used are *Trout*, which contains 5 attributes (input values called descriptors), and *Daphnia* which contains 11 attributes. Both datasets have an output value which is the relative toxicity of the chemical, and both contain input and output values for 184 chemicals. These 184 are the substances that existed in both of the two larger, original datasets from which *Trout* and *Daphnia* were created. Input values were scaled to the range $-1 \dots 1$. The two datasets were selected due to the existence of positive correlation between their output values as well as a justifiable biological similarity between the two species that the toxicity values were measured on. The two datasets are both subsets of larger datasets and were created by choosing only those chemicals for which input and output values existed in both of the original datasets. *Trout* and *Daphnia* were split 75/25 into a larger training set and a smaller testing set. The chemicals in the dataset were ordered by increasing toxicity and every fourth chemical was selected to be part of the testing set. The software package used to carry out the experiments was MATLAB® from The Mathworks [15].

Predictive Toxicology datasets are often characterised by their unusual size and dimensions, and the difficulties these factors cause when attempting to model the data within them. They often contain vastly more descriptors (attributes) than they do chemical instances (records), which can mean there is little relevant information to learn from, but a lot of irrelevant information for models to make incorrect assumptions on. The integration of additional data, possibly allowing increased learning to occur, and of course the relationships between related datasets within this field, means that Predictive Toxicology may benefit greatly from this technique.

In the experiments each dataset was used to construct a number of feed-forward back-propagation ANNs with varying numbers of hidden neurons (from two to six). Each ANN was trained using the gradient descent algorithm to make predictions for the *single* output value of the dataset. Whilst the relatively large number of hidden neurons used (when compared to the number of inputs) may lead to overfitting on the training data, the authors were interested to see how prone to overfitting the networks were, especially when compared to the larger dual output networks. The featured networks were selected based on their ability to model the training data as well as the unseen testing data. In addition to this, the two datasets were integrated to form a union dataset, *TD*. This new dataset contained 16 inputs and two output values. *TD* was used to train neural networks with *two* outputs, one for each of the output values in the combined dataset. Again a varying number of hidden neurons were used (from two to six, as well as nine and twelve) and ANNs selected based on their accuracy on both training and testing data. Fig. 1 shows the training and testing accuracies for the *Trout* output from both the single output ANNs and the dual output ANNs with varying numbers of hidden neurons, and Fig. 2 shows similar results for the *Daphnia* output.

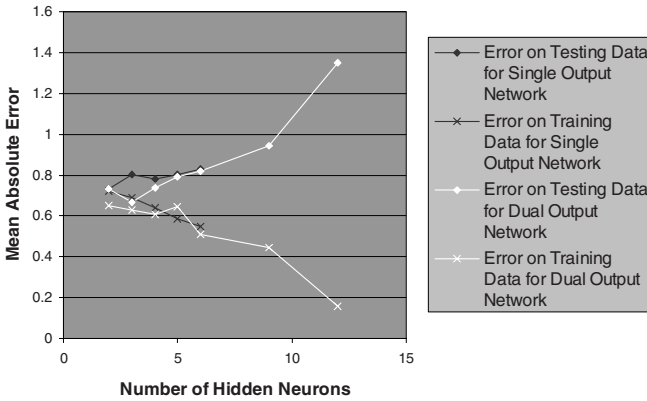


Fig. 1. Errors on testing (*diamonds*) and training (*crosses*) data for *Trout* outputs from single output networks (*black lines*) and from dual output networks (*white lines*) measured against the number of neurons in the hidden layer of the networks

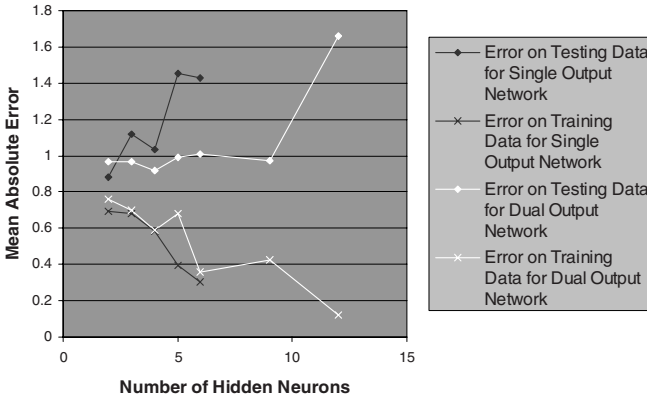


Fig. 2. Errors on testing (*diamonds*) and training (*crosses*) data for *Daphnia* outputs from single output networks (*black lines*) and from dual output networks (*white lines*) measured against the number of neurons in the hidden layer of the networks

As we can see from the figures, both the single and dual output networks were prone to overfitting on the data, characterised by the decreasing error on training data with a corresponding increase in error on the testing data as the number of neurons in the hidden layers increases. For the *Daphnia* data the dual output network seemed more resistant to overfitting, with only the network having twelve hidden neurons showing a significant increase in error on the testing data. Note that the dual output networks generally show a smaller error on testing data than the single output networks with the same number of neurons in the hidden layer.

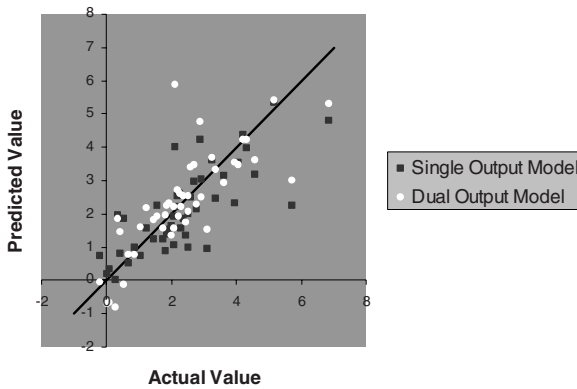


Fig. 3. Predicted output values plotted against actual output values for testing data for the *Trout* dataset for the best single output (*black squares*) and dual output (*white circles*) networks

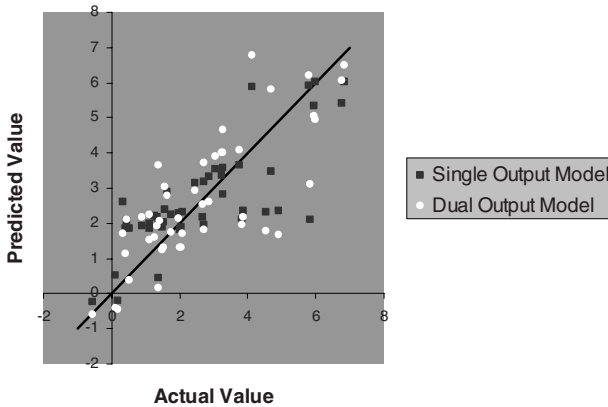


Fig. 4. Predicted output values plotted against actual output values for testing data for the *Daphnia* dataset for the best single output (*black squares*) and dual output (*white circles*) networks

The Mean Absolute Error (MAE) was used to determine the ‘best’ models, where MAE is defined as the average of the absolute numerical differences between a model’s prediction for an output value and the actual value of the output. For the single output models, the best for both *Daphnia* and *Trout* each contained two hidden neurons. For the dual output networks, the best model was found by taking the average of the MAE for the *Trout* output and the MAE for the *Daphnia* output; this was in fact the network with three hidden neurons.

The best single output model for *Trout* data achieved a MAE on testing data of 0.73, whilst the best dual output network achieved a slightly better value of 0.67. Fig. 3 is a scatter graph showing the predicted outputs from the models against the actual outputs, for testing data from the *Trout* dataset. Outputs from both the single

and dual output networks are shown for comparison. The best single output model for *Daphnia* data achieved a MAE on testing data of 0.88, whilst the best dual output network achieved a slightly worse 0.97. Fig. 4 shows the predicted outputs from the models against the actual outputs, for testing data from the *Daphnia* dataset. As we can see from fig. 3, the *Trout* data appears easier to model, with a better distribution of data points nearer to the black line (which represents a perfect prediction). With the exception of a few outliers the majority of predictions made by both the single and dual output networks lie close to this line. However, the lower MAE for the dual output model helps draw attention to the overall better distribution of the white circles on the graph, with many more lying close the line of perfect prediction. Fig. 4 shows the relative difficulty of modelling the *Daphnia* data, with both models showing a more scattered distribution of data points. Whilst the dual output network did perform slightly worse overall, in some areas of the graph it avoids forming rather worrying ‘holes’, where the single model makes no predictions between certain numerical ranges. Note the lack of black squares in the ranges 1.5-2 and 3.5-5.5 on the x-axis.

Overall, the best dual output model shows better accuracies for one output, and arguably better distribution of predictions for the other output compared to the single output models. This shows the excellent potential that the proposed algorithm has for integrating related data. Further testing of the algorithm using benchmark datasets and cross validation techniques is proposed as future work.

4 Conclusions

In this paper the authors have proposed an algorithm to integrate related data by using multi-input, multi-output connectionist models. The algorithm was shown to have good potential for use with real world data when applied to a pair of Predictive Toxicology datasets. The results reported show that these multi-output models can lead to improved accuracies when compared to single output models, and that the distribution of prediction values within the output space can arguably be improved, or at the very least altered in a significant way. The best dual output model from section 3.1 was able to exhibit both of these effects without the need for a proportional increase in the number of neurons compared to the two best single output networks. This behaviour is likely due to the increased opportunity for learning to occur when training the network on the expanded datasets. The entire set of dual output models created showed more resistance to overfitting, particularly on the *Daphnia* data (see fig. 2). This may be due to the inclusion of input values from the other integrated datasets, forcing the model to learn from data that, although related, may not have been directly chosen for its correlation to the output values, thus reducing the potential for models to learn *too* well from training data.

The models built in section 3.1 used a subset of data from the EU project DEMETRA. One of the partner institutions working on this project reported relatively good results when modelling the data on Trout and *Daphnia* toxicity. The average MAE for final models for Trout testing data was 0.66; slightly better than the MAE for the Trout output from the dual model in this paper of 0.67. For *Daphnia* models, the DEMETRA partner was able to create a set of models with an average MAE on

testing data of 0.79; much better when compared to the best single output model from this paper, which had an MAE of 0.88.

Whilst models reported in recent publications in the field of Predictive Toxicology can come with higher accuracy, the authors feel that the results from this paper are extremely promising when we consider that the models developed by the authors were unoptimised. In conclusion, the algorithm proposed in this paper has been shown to be potentially very useful for model builders who find themselves in possession of multiple related datasets, and who wish to create models with possibly higher predictive power than standard model building techniques.

Acknowledgments. The authors acknowledge the support of the EPSRC for their funding of the CASE project: 'Development of artificial intelligence-based in-silico toxicity models for use in pesticide risk assessment', as well as the Central Science Laboratory at York for additional funding. The authors also acknowledge the support of the EU project DEMETRA (EU FP5 Quality of Life DEMETRA QLRT-2001-00691) and the EPSRC project Pythia (EPSRC GR/T02508/01).

References

1. Haykin, S: *Neural Networks: A Comprehensive Foundation* (2nd Edition). Prentice Hall, (1998), 842 pages
2. Cristianini, N., Shawe-Taylor, J: *An Introduction to Support Vector Machines: and Other Kernel-based Learning Methods*. Cambridge University Press, (2000), 204 pages
3. Quinlan, J.R: *Induction of Decision Trees*. Machine Learning, Vol. 1 No.1, (1986), 81-106
4. Cohen, W.W: *Fast Effective Rule Induction*. In *Proceeding of the 12th International Conference on Machine Learning*, (1995)
5. Aha, D.W., Kibler, D., Albert. M.K: *Instance-based Learning Algorithms*. Machine Learning, Vol.6, No.1, (1991), 37-66
6. Bates, D.M., Watts, D.G: *Nonlinear Regression Analysis and Its Applications*. Wiley, (1988), 384 pages
7. Cooper, G. F., Herskovits, E: *A Bayesian Method for the Induction of Probabilistic Networks from Data*. Machine Learning, Vol.9, No. 4, (1992), 309-347
8. Qian & Sejnowski: *Predicting the Secondary Structure of Globular Proteins Using Neural Network Models*. *Journal of Molecular Biology*, Vol.202, (1998) 865-884
9. Kim & Calise: *Nonlinear Flight Control Using Neural Networks*. *Journal of Guidance, Control and Dynamics*, Vol.20, No.1, (1997), 26-33
10. White: *Economic Prediction Using Neural Networks: The Case of IBM Dailystock Returns*. *Proceedings of the 1988 IEEE International Conference on Neural Networks*, Vol.2, (1988), 451-458
11. Ghosh & Schwartzbard: *A Study in Using Neural Networks for Anomaly and Misuse Detection*. *Proceedings of the 8th USENIX Security Symposium*, (1999), 141-152
12. Rowley, Baluja & Kanade: *Neural Network-based Face Detection*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No.1, (1998), 23-38
13. MATLAB, The Mathworks, <http://www.mathworks.com> (accessed March 2007)
14. Development of Environmental Modules for Evaluation of Toxicity of pesticide Residues in Agriculture – DEMETRA, EU FP5 Contract No. QLK5-CT-2002-00691, <http://www.demetra-tox.net/> (accessed March 2007)

Combined Projection and Kernel Basis Functions for Classification in Evolutionary Neural Networks*

P.A. Gutiérrez¹, C. Hervás¹, M. Carbonero², and J.C. Fernández¹

¹ Department of Computing and Numerical Analysis of the
University of Córdoba, Campus de Rabanales, 14071, Córdoba, Spain
zamarck@yahoo.es, chervas@uco.es, i82fecaj@uco.es

² Department of Management and Quantitative Methods, ETEA,
Escritor Castilla Aguayo 4, 14005, Córdoba, Spain
mariano@etea.com

Abstract. This paper describes a methodology for constructing the hidden layer of a feed forward network using a possible combination of different transfer projection functions (sigmoidal, product) and kernel functions (radial basis functions), where the architecture, weights and node typology is learnt using an evolutionary programming algorithm. The methodology proposed is tested using five benchmark classification problems from well-known machine intelligence problems. We conclude that combined functions are better than pure basis functions for the classification task in several datasets and that the combination of basis functions produces the best models in some other datasets.

Keywords: classification, projection basis functions, kernel basis functions, evolutionary neural networks.

1 Introduction

Neural Networks (NNs) have been an important tool for classification since recent research activities have identified them as a promising alternative to conventional classification methods, such as the linear discriminant analysis or decision trees. Different types of NNs are now being used for classification purposes [1]; multilayer perceptron neural networks (MLP) where the transfer functions are Sigmoidal Unit basis functions, SU; Radial Basis Functions (RBF), kernel functions where the transfer function are usually Gaussian [2]; the General Regression Neural Networks (GRNN) proposed by Specht [3]; a class of multiplicative NNs which comprises such types as sigma-pi networks and Product Unit networks, PU, [4], [5]. The RBF network can be considered a local average procedure and the improvement in both its approximation ability as well as in the construction of its architecture has been noteworthy. Bishop [2] concluded that an RBF network can provide a fast, linear

* This work has been partially subsidized by TIN 2005-08386-C05-02 projects of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT) and FEDER funds. The research of P.A. Gutiérrez and J.C. Fernández has been backed respectively by the FPU and the FPI Predoctoral Programs (Spanish Ministry of Education and Science).

algorithm capable of representing complex non-linear mappings. A Product-Unit basis function neural network (PU) is a class of high-order NNs, which have only unidirectional interlayer connections.

Friedman and Stuetzle [6] proposed Projection Pursuit Regression (PPR), a procedure for estimating a smooth function of several variables from scattered, noisy data. PPR algorithms produce a fitted multivariate regression function \hat{f} of the form:

$$\hat{f}_n(x) = \sum_{i=1}^n \hat{g}_i(\hat{\alpha}_i^t \mathbf{x}) \quad (1)$$

where the i th term $\hat{g}_i(\cdot)$ in the sum is constant throughout $\alpha_i^t \mathbf{x} = c$, \mathbf{x} is the input vector, $\{\alpha_i\}$ are unit vectors and each $\alpha_i^t \mathbf{x}$ may be thought of as a projection of \mathbf{x} and so is often called a ridge regression. The estimate at a given point can be considered to be based on averages over certain (adaptively chosen) narrow strips $\{\mathbf{x} : |\alpha_i^t \mathbf{x} - t_i| \leq \varepsilon\}$. This contrasts with kernel and other local averaging procedures, in which the smoothing is done over small balls of the form $\{\mathbf{x} : |\mathbf{x} - \mathbf{x}_0| \leq \varepsilon\}$. In 1989, Donoho and Johnstone [7] prove that, in a certain setting, projection-based and local-averaging based function estimates have complementary properties. But, is it possible in a classification context to combine several projection-based and kernel-based functions to optimize the accuracy of the classifier? Our research aimed to answer this question, comparing the performance of pure basis function models to the performance of combined basis function ones for the purpose of classification.

2 Combined Basis Functions for Classification

In a classification problem, measurements x_i , $i = 1, 2, \dots, k$, of a single individual (or object) are taken, and the individuals are to be classified into one of the J classes based on these measurements. A training sample $D = \{(\mathbf{x}_n, \mathbf{y}_n); n = 1, 2, \dots, N\}$ is available, where $\mathbf{x}_n = (x_{1n}, \dots, x_{kn})$ is the random vector of measurements taking values in $\Omega \subset R^k$, and \mathbf{y}_n is the class level of the n -th individual. We adopt the common technique of representing class levels using a “1-of- J ” encoding vector $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(J)})$. We define the Correctly Classified Rate by

$CCR = \frac{1}{N} \sum_{n=1}^N I(C(\mathbf{x}_n) = \mathbf{y}_n)$, where $I(\cdot)$ is the zero-one loss function. A good classifier tries to achieve the highest possible CCR in a given problem. We consider the softmax activation function [8] given by:

$$g_l(\mathbf{x}, \boldsymbol{\theta}_l) = \frac{\exp f_l(\mathbf{x}, \boldsymbol{\theta}_l)}{\sum_{l=1}^J \exp f_l(\mathbf{x}, \boldsymbol{\theta}_l)}, l = 1, 2, \dots, J \quad (2)$$

where J is the number of classes in the problem, $f_l(\mathbf{x}, \boldsymbol{\theta}_l)$ the output of the l neuron for pattern \mathbf{x} and $g_l(\mathbf{x}, \boldsymbol{\theta}_l)$ the probability of pattern \mathbf{x} belonging to class l . The hybrid model to estimate the function $f_l(\mathbf{x}, \boldsymbol{\theta}_l)$, which we will call $f_l(\mathbf{x})$, for simplicity's sake, is given for a combination of projection and kernel basis functions:

$$f_l(\mathbf{x}) = \sum_{j=1}^{m_1} \beta_j^1 B_j^1(\mathbf{x}, \mathbf{w}_j^1) + \sum_{j=1}^{m_2} \beta_j^2 B_j^2(\mathbf{x}, \mathbf{w}_j^2) \quad (3)$$

We will consider two types of projection basis functions in (3) that can replace B_j^1 or B_j^2 : Sigmoidal Units (SUs) in the form:

$$B_j(\mathbf{x}; \mathbf{w}_j) = \frac{1}{1 + \exp\left(-u_{j0} - \sum_{i=1}^k u_{ji} x_i\right)} \quad (4)$$

and Product Units (PUs) with the following expression:

$$B_j(\mathbf{x}; \mathbf{w}_j) = \prod_{i=1}^k x_i^{w_{ji}} \quad (5)$$

As kernel basis functions replacing B_j^1 or B_j^2 in (3), we consider RBFs, in the form:

$$B_j(\mathbf{x}; (\mathbf{c}_j | r_j)) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{2r_j^2}\right) \quad (6)$$

\mathbf{c}_i and r_i being respectively the center and width of the Gaussian basis function of the i th hidden unit.

In this paper we interpret the outputs of the neurons on the output layer from the point of view of probability which considers the softmax activation function whose expression is found in (2). Taking this consideration into account, it can be seen that the class predicted by the neuron net corresponds to the neuron on the output layer whose output value is the greatest. The optimum rule $C(\mathbf{x})$ is the following:

$$C(\mathbf{x}) = \hat{l}, \text{ where } \hat{l} = \arg \max_l g_l(\mathbf{x}, \hat{\boldsymbol{\theta}}), \text{ for } l = 1, 2, \dots, J \quad (7)$$

The function used in this research to evaluate a classification model is the function of cross-entropy error and is given by the following expression for J classes:

$$l(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^N \sum_{l=1}^J y_n^{(l)} \log g_l(x_n, \boldsymbol{\theta}_l) = \frac{1}{N} \sum_{n=1}^N \left[-\sum_{l=1}^J y_n^{(l)} f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) + \log \sum_{l=1}^J \exp f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) \right] \quad (8)$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J)$. From a statistical point of view, with the soft-max activation function (2) and the cross-entropy error, the neural network model can be seen as a multilogistic regression model. Nevertheless, the non-linearity of the model with respect to parameters $\boldsymbol{\theta}_j$ and the indefinite character of the associated Hessian matrix do not recommend the use of gradient-based methods, (for example, Iteratively

Re-weighted Least Squares, commonly used in the optimization of log-likelihood in linear multinomial logistic regression), to minimize the negative log-likelihood function. Thus we use an evolutionary algorithm to determine both the number of nodes in the hidden layer as well as the weights of the net.

3 The Combined Basis Function Evolutionary Algorithm

In this section, we present the Evolutionary Algorithm (EA) used to estimate the parameters and the structure of the combined basis functions models, including the number of hidden nodes of each transfer/activation function. The objective is to design a neural network with optimal structure and weights for each classification problem tackled. The algorithm proposed is an extension of the Evolutionary Programming (EP) algorithm proposed in [5], [9], that is implemented and available in the non-commercial java tool named KEEL¹.

We consider three kinds of hidden nodes (Sigmoidal Units, SU, Product Unit, PU, and Radial Basis Functions, RBF) and three different combined basis functions models with all the different pairs that can be obtained with the three kinds of hidden nodes (Product-Sigmoidal Units neural networks, PSU, Product-Radial Basis Functions neural networks, PRBF, Sigmoidal-Radial Basis Functions neural networks, SRBF). We consider $p_i=0.5$, $i \in \{1,2\}$ as the probability that a hidden node has of belonging to the type t_i .

The general framework of the *Combined Basic Function Evolutionary Programming* (CBFEP) is the following:

1. Generate a random population of size N_p , where each individual presents a combined basis function structure.
2. Repeat until the stopping criterion is fulfilled
 - 2.a) Apply parametric mutation to the best 10% of individuals. Apply structural mutation to the remaining 90% of individuals, using a modified Add Node mutation in order to preserve the combined basis function structure.
 - 2.b) Calculate the fitness of every individual in the population.
 - 2.c) Add best *fitness* individual and best *CCR* individual of the last generation (*double elitist algorithm*).
 - 2.d) Rank the individuals with respect to their *fitness*.
 - 2.e) The best 10% of population individuals are replicated and substitute the worst 10% of individuals.
3. Select the best *CCR* individual and the best *fitness* individual in the final population and consider both as possible solutions.

As can be observed, the proposed algorithm returns both the best *CCR* and the best *fitness* individuals as solutions because it is difficult to ascertain a priori what the best approach for each problem considered will be.

The severity of mutations depends on the temperature $T(g)$ of the neural network model, defined by $T(g)=1-A(g)$, $0 \leq T(g) \leq 1$. The fitness measure is a strictly

¹ Knowledge Extraction based on Evolutionary Learning, <http://www.keel.es>

decreasing transformation of the entropy error $l(\theta)$ given by $A(g) = \frac{1}{1+l(\theta)}$, where

g is a combined basis function model given by the multivaluated function $g(\mathbf{x}, \theta) = (g_1(\mathbf{x}, \theta_1), \dots, g_J(\mathbf{x}, \theta_J))$. Next, the specific details of the CBFEP algorithm are explained concerning the optimization of the combined basis function structure, and the remaining characteristics can be consulted in [5],[9]. We will refer to the two types of hidden nodes considered in combined basis function topology as t_1 and t_2 .

In order to define the topology of the neural networks generated in the evolution process, we consider three parameters: m , M_E and M_I . They respectively correspond to the minimum and the maximum number of hidden nodes in the whole evolutionary process and the maximum number of hidden nodes in the initialization process. In order to obtain an initial population formed by models simpler than the most complex model possible, parameters must fulfil the condition $m \leq M_I \leq M_E$.

We generate $10N_p$ networks, where $N_p = 1,000$ is the number of networks of the population during the evolutionary process. Then we select the best N_p neural networks. For the generation of a network, the number of nodes in the hidden layer is taken from a uniform distribution in the interval $[m, M_I]$. Once the number of hidden nodes is decided, we generate each hidden node using a probability of 0.5 to decide if the node corresponds to t_1 or t_2 . For PU and SU hidden nodes, the number of connections between each node in the hidden layer and the input nodes is determined from a uniform distribution in the interval $(0, k]$, where k is the number of independent variables. For RBF hidden nodes, the number of connections is always k , since these connections represent the coordinates of the centre of the neuron. The number of connections between each hidden node and the output layer is determined from a uniform distribution in the interval $(0, J - 1]$.

Weights are initialized differently depending on the type of hidden node generated. For PU and SU hidden nodes, weights are assigned using a uniform distribution defined throughout two intervals, $[-5, 5]$ for connections between the input layer and hidden layer and, for all kinds of neurons, $[-10, 10]$ for connections between the hidden layer and the output layer. For RBF hidden nodes, the connections between the input layer and hidden layer are initialized using a clustering algorithm, so the EA can start the evolutionary process with well positioned centres. The main idea is to cluster input data in k groups, k being the number of hidden RBF neurons. In this way, each hidden RBF neuron can be positioned in the centroid of its corresponding cluster. Finally, the radius of the RBF hidden neuron is calculated as the geometric mean of the distance to the two closest centroids. The clustering technique chosen (described in [10]) is a modification of the classic *k-means* algorithm, in which the initial centroids are calculated using a specific initialization algorithm that avoids local minima, increasing the probability that the initial k cluster centres are not from a smaller number of clusters.

For PU and SU hidden nodes, parametric mutation is accomplished for each coefficient, of the model by adding a Gaussian noise represented by a one-dimensional normally-distributed random variable with mean 0 and adaptive variance. Radius r_i of RBF hidden nodes are mutated using a similar procedure. There are different structural mutations, similar to the mutations in the GNRL model [11]. Connection deletion and connection addition have no sense if the node mutated is a RBF. Therefore, they are not taken into consideration with this kind of nodes.

In node fusion, two randomly selected hidden nodes, a and b , are replaced by a new node c , which is a combination of the two. a and b have to be of the same type in order to accomplish this mutation. The connections that are common to both nodes are kept and the connections that are not shared by the nodes are inherited by c at a probability of 0.5, their weight being unchanged. The weight resulting from the fusion of common connections depends again on the type of hidden node chosen. For PU and SU nodes, the resulting weights are given by:

$$\beta_c^l = \beta_a^l + \beta_b^l, \quad w_{jc} = (1/2)(w_{ja} + w_{jb}) \quad (10)$$

while for RBF nodes, the fact that RBFs can be interpreted as circumferences with a perfectly identified centre and radius forces us to consider the resulting weights as:

$$\mathbf{c}_c = \frac{r_a}{r_a + r_b} \mathbf{c}_a + \frac{r_b}{r_a + r_b} \mathbf{c}_b, \quad r_c = \frac{r_a + r_b}{2} \quad (11)$$

\mathbf{c}_j being the center defined by the hidden neuron j , which is $\mathbf{c}_j = (w_{j1}, w_{j2}, \dots, w_{jk})$, and r_j being its radius.

In node addition, once the number of hidden nodes to be added is decided, we generate each hidden node using a probability of 0.5 to decide its type (t_1 or t_2). If the number of hidden nodes in the neural net is $M_E - 1$, hidden node is not added.

4 Experiments

In order to analyze the performance of the combination of basis functions, we have tested five databases in the UCI repository. The experimental design was conducted using a holdout cross-validation procedure (30 runs). All parameters in the CBFEP algorithm are common for all the problems, except the m , M_I and M_E values, which are represented in Table 1, together with the main characteristics of each database. For every dataset, we will perform an analysis of results obtained for each pure or combined basis function used in the neural network model for classification. Table 2 shows the mean value, standard deviation, best and worst result for training and generalization sets of the Correctly Classified Rate (CCR_G) and the mean and standard deviation for the number of net connections in 30 runs of the experiment for the best performing model (pure or combination) in each dataset. It can be seen that the combination of the basis functions produces good results with respect to CCR_G .

To ascertain the statistical significance of the differences between the performance for each dataset, we have carried out an ANOVA [12,13] with the CCR_G of the best models as the test variable (previously evaluating if the CCR_G values follow normal distribution, using a Kolmogorov-Smirnov test). This CCR_G is obtained independently in 30 runs and depends on one fixed factor, the basis functions used in the net. In a post hoc we perform a multiple comparison test of the average fitness obtained at different levels. First, we carry out a Levene test [14], [15], to evaluate the equality of variances. If the hypothesis that the variances are equal is accepted, we perform a Tukey test [14], while in other cases we perform a Tamhane test [16] to rank the mean of each level of the factor. Our aim is to find the level of each factor whose average fitness is significantly better than the average fitness of the rest of the levels. So, in order to determine for each dataset whether one pure basis or combined basis function is better than the others, we have performed an ANOVA I analysis - where the only factor is the typology of the basis function - as well as a multiple comparison test.

Table 3 shows the results obtained following the above methodology, including the test performed (Tukey or Tamhane) and the performance ranking of the different basis functions. For all datasets, Table 3 shows that there are significant differences between the means of CCR_G for different typologies of the basis functions.

Table 1. Main characteristics of each database tested and non-common parameters values

Dataset	#Instances	#Inputs	#Classes	#Gen.	m	M_I	M_E
Balance	625	4	3	500	3	4	5
Card	690	51	2	50	1	2	3
Glass	214	9	6	500	7	8	9
Ionosphere	351	34	2	300	3	4	5
Vote	435	16	2	10	1	1	2

Table 2. Statistical results using the best methodology (pure or combined basis functions, CCR or fitness elitism) for each dataset, of training and generalization CCR for 30 runs of the EA

Dataset	Func.	Elitism	Training				Generalization				# Links	
			Mean	SD	Best	Worst	Mean	SD	Best	Worst	Mean	SD
Balance	PSU	CCR	99.44	0.61	100.00	97.44	98.01	0.96	99.36	96.15	29.33	4.59
Card	SRBF	Fitness	86.16	0.93	88.03	84.56	88.02	1.01	90.70	86.05	62.23	21.04
Glass	SU	Fitness	75.22	2.52	80.12	71.43	67.67	3.49	77.36	62.26	83.83	6.03
Ionos	SRBF	CCR	98.13	0.88	99.62	96.21	93.22	1.61	97.70	89.66	100.27	16.62
Vote	PRBF	CCR	95.88	0.45	96.94	95.41	96.02	0.88	97.22	93.52	13.30	8.26

Table 3. Ranking of the basis function based on a multiple comparison test

Dataset	Test	Ranking
Balance	Tamhane	$\mu_{PSU} \geq \mu_{PRBF}; \mu_{PSU} > \mu_{PU}; \mu_{PRBF} \geq \mu_{PU} > \mu_{SU} > \mu_{SRBF} > \mu_{RBF}$
Card	Tamhane	$\mu_{SRBF} \geq \mu_{SU} \geq \mu_{PU} \geq \mu_{PSU} \geq \mu_{PRBF} > \mu_{RBF}$
Glass	Tukey	$\mu_{SU} \geq \mu_{SRBF} \geq \mu_{PSU} \geq \mu_{PU} \geq \mu_{PRBF} \geq \mu_{RBF}$
Ionos	Tamhane	$\mu_{SRBF} \geq \mu_{SU} \geq \mu_{PSU} \geq \mu_{PRBF} \geq \mu_{PU} \geq \mu_{RBF}; \mu_{SRBF} > \mu_{PRBF}; \mu_{SU} > \mu_{RBF}$
Vote	Tamhane	$\mu_{PRBF} \geq \mu_{PU} \geq \mu_{PSU} \geq \mu_{SRBF} \geq \mu_{SU} > \mu_{RBF}; \mu_{PRBF} > \mu_{PSU}$

5 Conclusions

The models proposed, formed by neural nets with activation/transfer functions of a projection and/or kernel type, are a viable alternative to obtain the best classification models. The nets proposed have been designed with an evolutionary algorithm constructed specifically for the typology of a combination of basis functions. The evaluation of the model and of the algorithm for the five data bases considered showed results that are comparable to those of other classification techniques found in machine learning literature [17]. The combined basis function improves the accuracy of the corresponding pure models in almost all problems evaluated. It can thus be affirmed, as a start, that the combination of basis functions provides a viable procedure for the resolution of classification problems where diverse function typologies can result in different types of behaviour in different areas of the definition domain.

References

1. Lippmann, R.P.: Pattern classification using neural net. *IEEE Com. Mag.*, 27 (1989) 47-64
2. Bishop, C.M.: Improving the generalization properties of radial basis function neural networks. *Neural Computation*, Vol. 3(4) (1991) 579-581
3. Specht, D.F.: A General Regression Neural Network. *IEEE Transactions on Neural Networks*, Vol. 2(6) (1991) 568-576
4. Durbin, R., Rumelhart, D.: Products Units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, 1 (1989) 133-142
5. Hervás, C., Martínez, F.J., Gutiérrez, P.A.: Classification by means of Evolutionary Product-Unit Neural Networks. *International Joint Conference on Neural Networks (2006)*, Canada.
6. Friedman, J.H., Stuetzle, W.: Projection Pursuit Regression. *J. Amer. Statist. Assoc.*, Vol. 76 (1981) 817-823
7. Donoho, D.L., Johnstone, I.M.: Projection based in approximation and a duality with kernel methods. *Ann. Statist.*, Vol. 17 (1989) 58-106
8. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press (1995).
9. Martínez-Estudillo, A.C., Martínez-Estudillo, F.J., Hervás-Martínez, C., et al.: Evolutionary Product Unit based Neural Networks for Regression, *Neural Networks*, 19,4 (2006) 477-486
10. Coen, S. y Intrator, N.: Global Optimization of RBF Networks. Technical report, School of Computer Science, Tel-Aviv University (2000)
11. Angeline, P. J., Saunders, G. M., Pollack, J. B.: An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, Vol. 5(1) (1994) 54-65
12. Miller, R.G.: *Simultaneous Statistical Inference*. 2nd edn. Wiley, New York (1981)
13. Snedecor, G.W., Cochran, W.G.: *Statistical Methods*. Iowa State University Press (1980)
14. Miller, R.G.: *Beyond ANOVA, Basics of App. Statistics*. Chapman & Hall, London (1996)
15. Levene, H.: In *Contributions to Probability and Statistics*. Chapter: Essays in Honor of Harold Hotelling. Stanford University Press (1960) 278-292
16. Tamhane, A.C., Dunlop, D.D.: *Statistics and Data Analysis*. Prentice Hall (2000)
17. Landwehr, N., Hall, M. et al.: Logistic Model Trees. *Machine Learning*, 59 (2005) 161-205

Modeling Heterogeneous Data Sets with Neural Networks

Lluís A. Belanche Muñz

Dept. de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Barcelona, Spain
belanche@lsi.upc.edu

Abstract. This paper introduces a class of neuron models accepting *heterogeneous inputs* and *weights*. The neuron model computes a user-defined *similarity function* between inputs and weights. The specific similarity function used is defined by composition of a Gower-based similarity with a sigmoid function. The resulting neuron model then accepts mixtures of continuous (crisp or fuzzy) numbers, and discrete (either ordinal, integer or nominal) quantities, with explicit provision also for missing information. An artificial neural network using these neuron models is trained using a *breeder genetic algorithm* until convergence. A number of experiments are carried out to illustrate the validity of the approach, using several benchmarking problems. The network is compared to a standard RBF network and shown to learn from non-trivial data sets with superior generalization ability in most cases. A further advantage of the approach is the interpretability of the learned weights.

Keywords: Artificial neural networks; similarity measures; data heterogeneity.

1 Introduction

A marked shortcoming of the neuron models existent in the literature is the difficulty of adding prior knowledge to the model (either of the problem to be solved or of data characteristics). The task of an artificial neural network is to find a *structure* in the data, to transform it from a *raw form* (possibly in a space of high dimension, only here and there covered by example cases) to a new *hidden* space (the space spanned by the hidden units) in such a way that the problem is simpler when projected to this space. This processing is repeated for all the subsequent hidden layers. This scenario is valid at least for the most widespread neuron models: that used in the multilayer perceptron (a scalar product between the input and weight vectors plus an offset, followed by a squashing function) and that used in the radial basis function neural network or RBFNN (a distance metric followed by a localized response function). In essence, the task of the hidden layer(s) is to find a new, more convenient representation for the problem *given* the data representation chosen, a crucial factor for a successful learning process that can have a great impact on generalization ability [1]. In practice, it may well be that for complex transformations several layers are needed or very many neurons per layer if the maximum number of hidden layers is restricted to one. This gives support to a more precise formulation of neuron models

as similarity computing devices. Being able to better capturing the required similarity relations in the input space, the task of computing the required transformations may be simpler to learn.

This work introduces a class of neuron models accepting *heterogeneous inputs* and *weights*. The neuron model computes a user-defined *similarity function* between inputs and weights. The specific similarity function used is defined by composition of a Gower-based similarity with a sigmoid function. The resulting neuron model then accepts mixtures of continuous (crisp or fuzzy) numbers, and discrete (either ordinal, integer or nominal) quantities, with explicit provision also for missing information.

An artificial neural network using these neuron models is trained using a *breeder genetic algorithm*. A number of experiments are carried out to illustrate the validity of the approach, using several benchmarking problems. These are selected as representatives because of the diversity in the kind of problem and richness in data heterogeneity. The network is compared to a standard RBFNN and shown to learn from non-trivial data sets with superior generalization ability in most cases. A further advantage of the approach is the interpretability of the learned weights.

2 Heterogeneous Neural Networks

In many important domains from the real world, objects are described by a mixture of continuous and discrete variables, usually containing missing information and characterized by an underlying uncertainty. For example, in the well-known UCI repository [2] over half of the problems contain *explicitly declared* nominal attributes, let alone other data types, usually unreported. This heterogeneous information has to be encoded in the form of real-valued quantities, although in most cases there is enough domain knowledge to design a more specific measure. Some of the commonly used methods [3,1] are: **Ordinal** variables coded as real-valued or using a *thermometer* scale. **Nominal** variables coded using a 1-out-of- c representation, being c the number of modalities. **Missing** values are either removed (actually the entire case) or “filled in” with the mean, median, nearest neighbor or else a new input is added, equal to one only if the value is absent. Statistical approaches need to make assumptions about or model the input distribution itself [1], or are computationally intensive [4]. **Fuzziness** and other forms of uncertainty are usually not reported.

Heterogeneous neural networks are neural architectures built out of neuron models allowing for heterogeneous and imprecise inputs, defined as a mapping $h: H^n \rightarrow R_{out} \subseteq R$. Here R denotes the set of real numbers and H^n is a cartesian product of an arbitrary number n of *source sets* $H^{(k)}$, $k=1\dots,n$. These source sets may include extended reals $R_k \cup \{X\}$ (where each R_k is a real interval), extended families of fuzzy sets $F_k \cup \{X\}$, and extended finite sets of the form $O_k \cup \{X\}$, $M_k \cup \{X\}$, where the O_k have a full order relation, while the M_k have not. The special symbol X extends the source sets and denotes the unknown element (missing information), behaving as an *incomparable* element with respect to any ordering relation. According to this definition, neuron inputs are vectors composed of n elements among which there might be reals, fuzzy sets, ordinals, categorical and missing data. An

heterogeneous neuron computes a *similarity measure*, or proximity relation, followed by the familiar form of a squashing non-linear function, with domain in $[0,1]$.

Let us represent patterns belonging to a space $S \neq \emptyset$ as a vector of n components, where each component x_{ik} represents the value of a particular feature k . A *similarity measure* in S is a unique number expressing how “like” two patterns are, given only these features. Let us denote by s_{ij} the similarity between x_i and x_j in S , that is, $s_{ij} = s(x_i, x_j)$. A similarity measure may be required to be non-negative, symmetric, bounded and strongly reflexive. Transitivity considerations are usually put aside [5].

A basic but very useful heterogeneous measure can be devised using a Gower-like similarity index [6], which is nothing more than an average of the similarities among the different variables, with provision for missing information. This coefficient has its values in the real interval $[0,1]$ and for any two objects x_i and x_j given by tuples of cardinality n , is given by the expression:

$$s_G(x_i, x_j) = \left(\frac{\sum_{k=1}^n \delta_{ijk} s_{ijk}}{\sum_{k=1}^n \delta_{ijk}} \right) \tag{1}$$

where $\delta_{ijk} = \delta(x_{ik} \neq X \wedge x_{jk} \neq X)$ is a binary function expressing whether the objects are *comparable* or not according to variable k (here δ is 1 if its argument is true and 0 if it is false). This treatment handles what is not known in a way it does not affect the known values. The similarities for objects x_i and x_j according to their value for variable k are denoted $s_{ijk} = s(x_{ik}, x_{jk})$. These values are normalized to a common real interval ($[0,1]$ in this case) and computed according to different formulas for different variables (possibly but not necessarily determined by variable type alone). Given these definitions, a *neuron model* can be devised that is both similarity-based and handles data heterogeneity and missing values, as follows. Let $\Gamma_i(x)$ be the function computed by the i -th neuron, where $x \in H^n$, having a weight vector $\mu_i \in H^n$ and smoothing parameter γ_i belonging to $(0,1)$. Then $\Gamma_i(x) = \mathfrak{g}(\gamma_i s_G(x, \mu_i))$. For the activation function \mathfrak{g} , a modified version of the classical logistic can be used, which is an automorphism (a monotonic bijection) in $[0,1]$. A layered, feed-forward architecture, with a hidden layer composed of heterogeneous neurons and an output layer of classical ones is a straightforward choice, thus conforming a *hybrid* structure. The heterogeneous neural network computes the function:

$$F_{\text{HNN}}(x) = \sum_{i=1}^h \alpha_i \Gamma_i(x) \tag{2}$$

where h is the number of (heterogeneous) hidden neurons and $\{\alpha_i\}$ is the set of mixing coefficients. The general training procedure for the heterogeneous neural network (HNN for short) is based on evolutionary algorithms, due to the presence of

missing information and the likely non-differentiability of the similarity measure. Specifically, the training procedure for the HNN used in this work is based on a *Breeder Genetic Algorithm* (BGA) [7], a method in mid position between Genetic Algorithms (GA) and Evolution Strategies. While in GA selection is stochastic and meant to mimic Darwinian evolution, BGA selection is driven by a deterministic *breeding* mechanism, an artificial method in which only the best individuals (usually a fixed percentage τ of total population size μ) are selected to be recombined and mutated, as the basis to form a new generation. The used BGA does not need any coding scheme, being able to cope with heterogeneous or missing genes [8].

3 Heterogeneous Similarity Measures

In this subsection specific measures defined to belong to the real interval $[0,1]$ are presented for the sake of illustration, but there certainly are many variations that could be superior choices, thereby making better use of available expert knowledge.

Ordinal variables. It is assumed that the values of the variable form a linearly ordered space (O, \leq) . Let $x_{ik}, x_{jk} \in O$, $x_{ik} \leq x_{jk}$, P_{lk} the fraction of values of variable k that take on the value x_{lk} . Then define

$$s_{ijk} = \frac{2 \log(P_{ik} + \dots + P_{jk})}{\log P_{ik} + \dots + \log P_{jk}} \quad (3)$$

where the summations run through all the ordinal values x_{lk} such that $x_{ik} \leq x_{lk}$ and $x_{lk} \leq x_{jk}$ [9].

Nominal variables. It is assumed that no partial order exists among these values and the only possible comparison is equality. The basic similarity measure for these variables is the overlap. Let N be a *nominal* space and $x_{ik}, x_{jk} \in N$. Then $s_{ijk}=1$ if $x_{ik}=x_{jk}$ and 0 otherwise.

Continuous variables. Let $x, y \in \Gamma = (r^-, r^+)$ a non-empty real interval. The standard metric in R is a metric in Γ . Therefore, for any two values $x_{ik}, x_{jk} \in R$:

$$s_{ijk} = \sigma \left(\frac{|x_{ik} - x_{jk}|}{\sup_{x, y \in \Gamma} |x - y|} \right) \quad (4)$$

where $\sigma : [0,1] \rightarrow [0,1]$ is a decreasing continuous function. The family $\sigma(z) = (1-z^\beta)^\alpha$, $0 < \beta \leq 1, \alpha \geq 1$ is used here, with $\alpha=4, \beta=1$.

Integer variables. Given that $N \subset R$, any distance-based similarity in R is also valid in N . A flexible choice does not limit the range of integer values (assumed positive for

convenience). In this case, a self-normalizing distance-based similarity $s_{ijk} = \sigma(|x_{ik} - x_{jk}|)$ is in general indicated, where $\sigma : [0, \infty) \rightarrow (0, 1]$ is a decreasing continuous function. In particular, the function $\sigma(z) = 1/(1+z)$ can be used.

Fuzzy variables. For variables representing fuzzy sets, similarity relations from the point of view of fuzzy theory have been defined elsewhere [10], and different choices are possible. In possibility theory, the possibility expresses the likeliness of co-occurrence of two vague propositions, with a value of one standing for absolute certainty. For two fuzzy sets A,B of a reference set X, it is defined as:

$$\Pi_{(A)}(B) = \sup_{u \in X} (\mu_A \cap \mu_B(u)) = \sup_{u \in X} (\min(\mu_A(u), \mu_B(u))) \tag{5}$$

Notice that this measure is indeed strongly reflexive and also symmetric. In our case, if F_k is an arbitrary family of fuzzy sets, and $x_{ik}, x_{jk} \in F_k$, the following similarity relation can be used $s_{ijk} = \Pi_{(x_{ik})}(x_{jk})$.

4 An Experimental Comparison

A number of experiments are carried out to illustrate the validity of the approach, using several benchmarking problems. These are selected as representatives because of the diversity in the kind of problem and richness in data heterogeneity. A total of eight learning tasks are worked out, taken from [2,3], and altogether representative of the kinds of variables typically found in real problems, while displaying different degrees of missing information (from none to 26%). Their main characteristics are displayed in Table 1. For every data set, the available documentation is analysed in what concerns type and meaning of variables, allowing an assessment on the more appropriate treatment. Missing information is also identified. There is hence an explicit *transfer of knowledge* from the domain to the heterogeneous neuron model. Given that no specific information was available, no fuzzy data were derived.

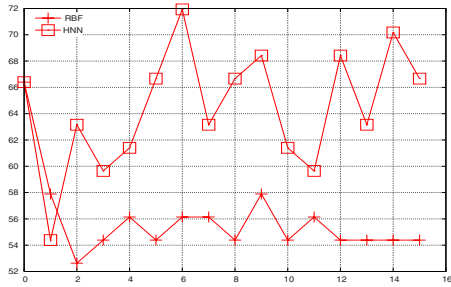
Table 1. Basic features of the data sets. 'Missing' refers to the *percentage* of missing values, p to the number of examples and n to that of variables. Data types are: C (continuous), N (nominal), I (integer), D (ordinal) and F (fuzzy). 'Output' is R for regression or else a number indicating the number of classes.

Name	p	n	Output	Missing	Data Types
Pima Diabetes	768	8	2	10.6%	7C, 2I
Audiology	226	69	4	2.3%	8D, 61N
Boston Housing	506	13	R	none	11C, 1I, 1N
Solar Flares	1066	9	2	none	6N, 3D
Heart Disease	920	13	2	16.2%	3C, 7N, 3I
Horse Colic	364	20	3	26.1%	8D, 5N, 2I, 5C
Credit Screening	690	15	2	0.6%	6C, 9N
Servo Data	167	4	R	none	2C, 2N

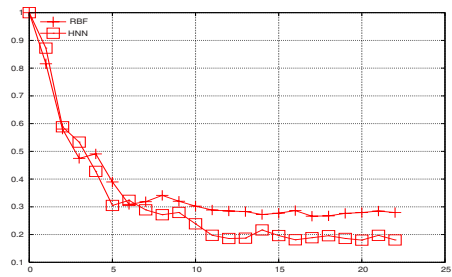
The HNN is compared to a standard RBF network, since it can be naturally seen as a generalization of the RBF. This is so because the response of hidden neuron is localized: centered at a given object (the neuron weight, where response is maximum), falling down as the input is less and less similar to that center. Both networks are trained using exactly the same procedure to exclude this source of variation from the analysis. The RBF network needs a pre-processing of the data, in accordance of Table 1 and following the recommendations in [3]. The RBF neuron model has provision for a smoothing parameter, different for every hidden neuron.

The network architecture is fixed to one single layer of hidden neurons ranging from 1 to \sqrt{p} and as many output neurons as required by the task, logistic for classification problems and linear otherwise. The input variables for the RBF network are standardized to zero mean, unit standard deviation. This is not needed by the heterogeneous neurons because they compute a normalized measure, but is beneficial for the RBF units. The output is presented for both networks in 1-out-of- c mode (that is, c outputs) for classification problems with c classes. The error measure in this case is generalized cross-entropy (GCE) [1]. For regression problems, the mean is subtracted to the continuous output and the *normalized root mean square* (NRMS) error is reported. The weights (including biases and standard deviations) for the RBF network are let to vary in $[-10, 10]$, a sufficiently wide range given the normalization chosen; the same interval is used for the hidden-to-output weights in both networks. The BGA is set to the following parameters: $\mu = 60$, $\tau = 5$.

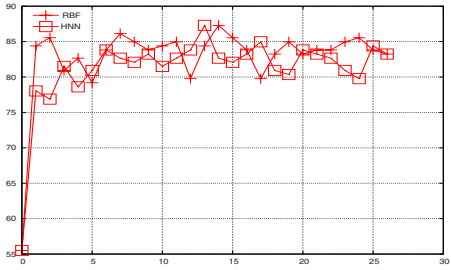
Each data set is split in three parts, for training (TR), validation (VA) and test (TE), as 50%-25%-25%. The BGA task is asked to minimize error (either NRMS or GCE) on the TR part, until 1,000 generations or convergence. Then the network having the lowest error on the VA part is returned. The reported performance is the NRMS or GCE of that network on the TE part. We present performance results in Figures. 1 (a) to 1 (h). In each figure, the abscissa represents the number of neurons in the hidden layer, and the ordinate axis the NRMS or GCE of that network on the TE part, as explained above, both for the HNN and the RBF networks. For classification problems, the abscissa value 0 is used to indicate the percentage of examples in the majority class (that is, the minimum acceptable performance, as a reference). For regression problems, this value is set to 1.0, which corresponds to the performance that the best constant model would achieve (again as a reference). Note that for classification problems, higher values are better, whereas for regression problems, lower values are better. The HNN shows enhanced performance for most of the problems, especially for those displaying higher heterogeneity or missingness, and less so (not conclusive) when this is not the case. This is reasonable since the HNN behaves as a RBF network when there is no heterogeneity or missingness. The interpretability of the learned weights is also enhanced, since every weight is of the same type as its matching input.



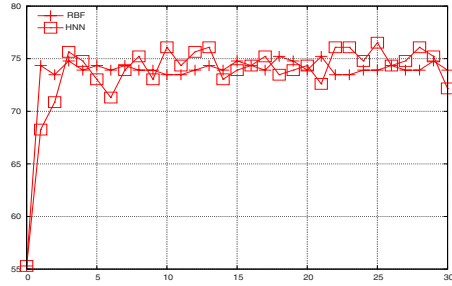
(a) Audiology



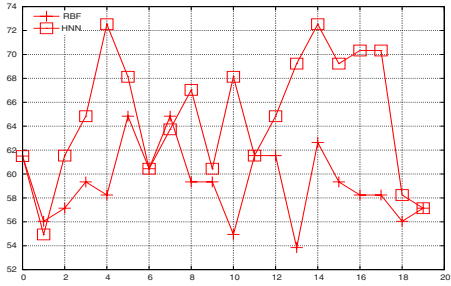
(b) Boston Housing



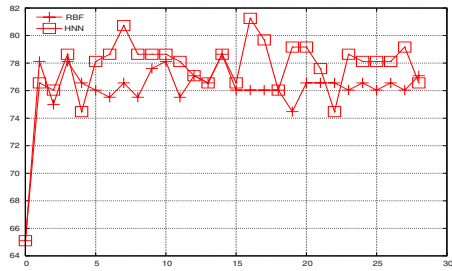
(c) Credit Screening



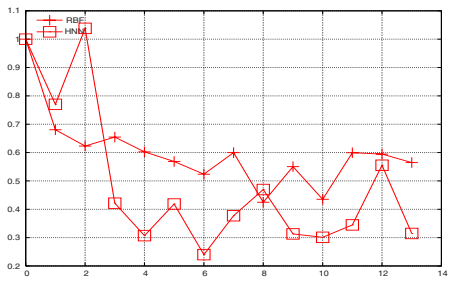
(d) Heart Disease



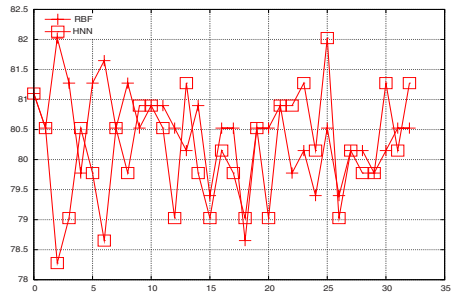
(e) Horse Colic



(f) Pima Diabetes



(g) Servo Data



(h) Solar Flares

Fig. 1. Generalization performance results

Acknowledgements. This work has been supported by the Ministerio de Educación y Ciencia, under project CGL2004-04702-C02-02.

References

1. C. Bishop. *Neural Networks for Pattern Recognition*. Oxford, (1995).
2. P.M. Murphy, D. Aha. *UCI Repository of machine learning databases*. UCI Dept. of Information and Computer Science, (1991).
3. L. Prechelt. *Proben1 - A set of Neural Network Benchmark Problems and Benchmarking Rules*. Fac. für Informatik. Univ. Karlsruhe T.R. 21/94 (1994).
4. V. Tresp, S. Ahmad, R. Neuneier. *Training Neural Networks with Deficient Data*. In *NIPS 6*, Morgan Kaufmann, (1994).
5. J.L. Chandon, S. Pinson. *Analyse Typologique*. Masson, (1981).
6. J.C. Gower. *A General Coefficient of Similarity and some of its Properties*. *Biometrics*, 27: 857-871, (1971).
7. H. Müllenbein, D. Schlierkamp-Voosen. *Predictive Models for the Breeder Genetic Algorithm*. *Evolutionary Computation*, 1(1) (1993).
8. Ll. Belanche, Ll. Evolutionary Optimization of Heterogeneous Problems. In *LNCS 2439*, pp. 475-484 (2002).
9. D. Lin. *An Information-Theoretic Definition of Similarity*. *Proceedings of International Conference on Machine Learning* (1998).
10. L. Zadeh. *Fuzzy Sets as a basis for a theory of possibility*. *Fuzzy Sets and Systems*, 1: 3-28, (1978).

A Computational Model of the Equivalence Class Formation Psychological Phenomenon

José Antonio Martín H.¹, Matilde Santos¹, Andrés García², and Javier de Lope³

¹ Facultad de Informática, Universidad Complutense de Madrid
jamartinh@fdi.ucm.es, msantos@dacya.ucm.es

² Facultad de Psicología, Universidad Nacional de Educación a Distancia
agargar@psi.uned.es

³ Dept. Sistemas Inteligentes Aplicados, Universidad Politécnica de Madrid
javier.delope@upm.es

Abstract. A Computational model of the Equivalence Class formation psychological phenomenon is presented. We develop our model based on the Self-Organized-Feature-Maps (SOFM) Artificial Neural Network (ANN) architecture which is used as a sort of associative memory. By means of a punishment/reward signal we model the induction of Equivalence Classes and we show that the proposed model fulfils the basic set of characteristics of induced Equivalence Classes observed in documented psychological experiments realized in Humans and non-Human Animals.

Keywords: Self-Organization, Reinforcement Learning, Equivalence classes, Neuro-Computing, behavioral simulation.

1 Introduction

Equivalence Classes in Psychology were firstly obtained by Sidman [3, 4] when working with Humans under a classical conditional discrimination experiment. One of the most significant experimental results along this line of research was the generation by means of reinforcement learning of groupings of stimuli. This type of grouping has been denominated Equivalence Classes following a parallelism with Set-Theory in mathematics where an equivalence relation or class is defined when the set verifies the properties: reflexive, symmetry and transitive. It is said that a Subject forms a class of stimuli or category when at the functional level it is able to respond of equivalent form to the elements of the category set. This phenomenon is the base of the Theory of Equivalence Classes and takes its name from Set theory. In an Equivalence Class any 2-subset of elements can be considered equivalent from the functional point of view.

Here we propose a computational model about such phenomenon based on the Self-Organized-Feature-Maps (SOFM) Artificial Neural Network model introduced by Kohonen [1, 2] and the Reinforcement Learning Paradigm [5].

2 The Equivalence Class Phenomenon

It is said that a Subject forms a class of stimuli or category when at the functional level it is able to respond of equivalent form to the elements of the category set. In an Equivalence Class any 2-subset (pair) of elements can be considered equivalent from the functional point of view. Rigorously, it is called an Equivalence Class respect of R defined by the element (a) that belongs to the set C to the subset formed by all the elements (x) of C that are related with (a). Any Equivalence Class will contain at least its representing element (a).

$$C[a] = \{x \in G / xRa\} \quad (1)$$

One of the most significant advantages of the behaviorist framework of the formation of classes of stimuli, concepts and categories is in the experimentation by means of conditional discriminations which gives a great flexibility allowing a huge variability and a procedural power. The main relevant fact about the power of the conditional discriminations is that without a previous training phase there are derived discriminative relations that can be studied following the normative mark of the classical Set theory. The pioneer study of Sidman [3] was the first documented study about the development of Equivalence Classes; nevertheless it took more than a decade of study and research to obtain a rigorous and systematic formulation about the emergent (now called derived) relations following the classical Set theory. The Equivalence class formation phenomenon is robust in Humans and has been applied in many studies with children and young people under very diverse experimental conditions, nevertheless in non-human animals there is a source of controversy about such phenomenon. Sidman claims that such phenomenon is exclusively of Humans [4] indeed he explains that the specific competence for the equivalence class formation is a Primitive of Behavior and hence cannot be explained in terms of other competencies. Also, this is another source of controversy due to the fact that some researchers in the field don't see the need for such competence has to be a Primitive of Behavior and it should be susceptible of analysis and decomposition into low level processes e.g. a natural computational low level mechanism.

2.1 Basic Properties of Equivalence Classes

The three fundamental properties of equivalence classes are: Reflexivity, Symmetry and Transitivity.

1. Reflexivity consists in the interchangeability of the sample with itself: ($A = A$).
2. Symmetry consists in the reciprocal and bidirectional relation between sample and stimulus: (if $A = 1$ then $1 = A$).
3. Transitivity refers to the transference of the association or relation between two or more conditional discriminants guided by some shared element:

$$(if A = B \text{ and } B = C \text{ then } A = C).$$

It is hypothesized that Equivalence Class formation is mainly due to the fulfillment of the Symmetry property since this property could be interpreted as the basis for

the other two properties, for instance, the reflexivity relation could be explained by the symmetry acting over the element with itself. We must recall that at the physical level the equality between a stimulus with itself has no rigorous meaning indeed it could be of no sense. Thus, if at the physical level there is no equality in rigorous sense there is no Reflexivity property but Symmetry indeed. Therefore, and in the psychological framework of Equivalence Classes study with respect to the stimuli perception, when trying to test the Reflexivity property of a physical stimulus with itself we are actually testing symmetry between two different physical stimuli although its differences are imperceptible. In fact, the word “symmetry” comes from the root “equal measurement”, that is, two objects are equals with respect to a determined instrumental capability of measurement but not equals in absolute sense as in mathematics and logic. Being of this way, we can hypothesize that the reflexivity property arises because the reinforcement induces the perception of symmetry between two physical stimuli. Similarly, when reflexivity takes place induced by symmetry between three stimuli the transitivity property would emerge.

3 Computational Model of Equivalence Class Formation

The main motivation for such model is mainly due to the fact that it is a computational model, that is, a constructive operational mechanism that will allow to experiment within a digital computer the same kind of procedural techniques applied in the study of equivalence relations exhibited by animals providing this way a framework in which to test directly the similarities and differences between natural and artificial agents, that is, a kind of Turing-Test between Animals and digital computers as in Fig. 1.

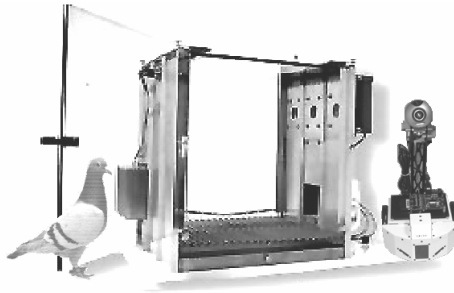


Fig. 1. A Turing test between a Pigeon and a Robot with a vision system

We base our computational model in the SOFM ANN architecture which can be easily related with the Equivalence Class formation phenomenon. We will present next the formal model of the SOFM ANN architecture and later we explain its operation in direct relation with the Equivalence Class formation phenomenon.

3.1 The SOFM Model

The Self-Organizing Map (SOM) model was developed in the middle of the eighty decade by Teuvo Kohonen [1, 2] as a natural extension to the previous work of von der Malsburg. This model is one of the most widely used unsupervised neural network architectures due to its simplicity and huge potential of practical applicability. The SOFM model takes inspiration from true physiological facts tending to form a topology preserving spatial map of features that are “self-organized” by means of lateral inhibition mechanisms, fact which indeed has been validated by neurophysiologic evidence. In its simpler version the SOFM model is represented as an $(n \times m)$ grid of processing units (nodes or neurons) which forms a single layer. Each processing unit is represented by a vector (code-book) $w_{ij}(t)$ where $w_{ij} \in \mathbb{R}^m$ which represents its synaptic weights. In the operative phase, the weights remains fixed. Each neuron (i,j) calculates the similitude between the input vector \mathbf{X} (x_k in $1 \leq k \leq m$) and its own vector of synaptic weights w_{ij} by means of some selected similarity criterion (e.g. Euclidian distance). Once every neuron has calculated its degree of similitude the most similar neuron to the input vector \mathbf{X} is selected as the winner neuron.

$$d(w_g, X) = \min_{ij} [d(w_{ij}, X)] \quad (2)$$

In this way each neuron acts as a specific feature detector and the winner neuron g indicates the specific feature or pattern detected in the input vector \mathbf{X} .

In the training phase (learning) each neuron is initialized randomly with the intention to cover uniformly the different features of the input space. The process for training the network starts with the presentation of an input vector \mathbf{X} and by means of applying the same procedure as in the operative phase, that is to select the winner neuron $d(w_g, x)$. The winner neuron g modifies its synaptic weights in order to make its vector w_g more similar to the input vector \mathbf{X} :

$$w_g = w_g + \alpha [\mathbf{X} - w_g], \quad (3)$$

where α is the learning rate and $(0 < \alpha < 1)$.

Note that this is the classical *moyenne adaptive modifiée* technique [6] which adaptively approximates the average μ of a set $x = (x_1 \dots x_\infty)$ of observations:

$$\mu = \mu + \alpha [x_i - \mu] \quad (4)$$

In this way in presence of the same input stimulus the neuron will respond in the future with more intensity. This process is repeated many times for different input vectors in such a way that at the end, the totality of the neurons will be specialized over different features of the input domain. One of the defining features of the SOFM model is the interrelation between proximal neurons in the MAP. The SOFM model introduces a neighborhood function that defines the surroundings of a winner neuron. The main idea is that the neighbor neurons of a winner neuron must be updated during the learning phase taking a fraction of the learning of the winner neuron; this fraction is determined by the neighborhood function. There are different frequently used

functions as the neighborhood function; the most commons are Gaussian, rectangular and the Mexican hat.

$$w_{ij} = w_{ij} + \alpha \times N_g(i, j) \times [X - w_{ij}], \tag{5}$$

where $N_g(i, j)$ is the neighborhood function and $(0 \leq N_g(i, j) \leq 1)$.

$N_g(i, j) = 1$ when w_{ij} is the winner neuron.

Thus the complete equation that represent the learning procedure of the SOFM model is shown in (6):

$$w_{ij}(t + 1) = w_{ijk}(t) + \alpha \times N_g \times [X_k - w_{ijk}(t)], \tag{6}$$

where w_{ijk} is the k^{th} synaptic weight of the neuron (i, j) , α is the learning rate, N_g is the neighborhood function taken for the winner neuron and x_k is the k^{th} value of the input vector \mathbf{X} .

In this way we can describe in general terms the SOFM model as a system divided in the next principal components:

1. A collection of processing unit w_{ij} called neurons where each neuron has an internal representation.
2. A similarity measure $d(.)$ which is applied to every neuron internal representation and an arbitrary input vector, that determines the level of activation of each neuron.
3. A neighborhood function $N_g(.)$ which represents the lateral interactions between the neurons.
4. A learning rule Δw_{ij} which modifies the internal representation of every neuron taking into account a learning rate parameter and the neighborhood function.

3.2 Adaptation of the SOFM Model for the Study of Equivalence Class Formation

Our model consist in a unique layer of neurons or processing units where each neuron will be represented by a square matrix of N rows and M columns where each cell will contain a real numeric value. A useful metaphor is to think on this structure as a device analogous to the ocular retina. Thus a layer formed by these processing units can be seen as a square mesh composed by neurons organized in rows and columns in an analogous way to the description of the functional organization of the primary visual cortex V1 and V2.

The training procedure is as follows:

For any presented stimulus (A) each neuron calculates its degree of similarity with (A) and the neuron with highest similarity will become active inhibiting the rest of neurons. We will denote by $G(A)$ the activated neuron when perceiving the stimulus (A).

Once a neuron has been activated it is reinforced or punished making that its internal representation be more close or far respectively to the stimulus (A), thus, if we repeat this training process for the same stimulus with reinforcement the behavior of a neuron can be seen as in the sequence shown in Fig. 2.

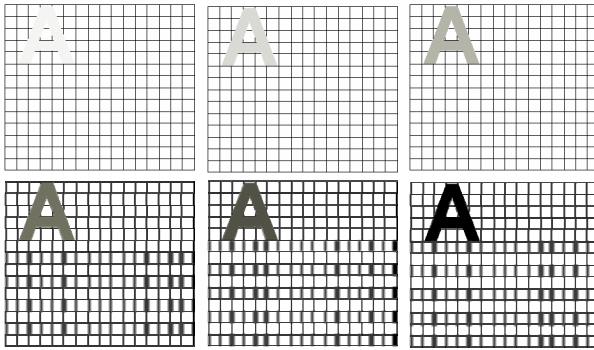


Fig. 2. A training sequence with reinforcement

Thus the general learning rule will follow Eq. 7.

$$w_{t+1} = w_t + \alpha \times \beta \times N_g \times [X - w_t], \quad (7)$$

where w_{t+1} is the internal representation of each neuron at time step $t+1$ (after the stimulus presentation), w_t is the internal representation of each neuron at time step t (before the stimulus presentation), α is a value in $[0,1]$ which modulates the learning rate, N_g is the neighborhood function, $(X - w_g)$ is the difference between the input stimulus $X = (A)$ and the internal representation of the activated neuron w_g and finally β is the reward/punishment signal normalized in the range $[-1, +1]$. Of this form under a repeated reinforcement/punishment process the internal representation of the whole Map will tend to associate positively or negatively the features presented in the set of training stimuli.

Returning to the case of Fig. 2 the neuron will be conditioned to respond to the stimulus (A) and whenever the stimulus be (A) or very similar to (A) the neuron will be activated.

3.3 Equivalence Class Induction

Let us suppose that once we have trained the network in order to recognize and respond to the stimulus (A) the network is trained to adapt and respond to a composite stimulus (A+B). Given the fact that the neuron has been trained to respond when the stimulus (A) is presented it will respond in the same way to the composite stimulus (A+B) since (A) is contained in (A+B) and thus the training will produce an association between the stimuli (A) and (B) as shown in Fig. 3.

Of this form, the neuron has been trained to respond to the composite stimulus (A+B) and in fact to respond to the stimulus (A) and to the stimulus (B). It is of importance to note that now this neuron fulfills two of the properties of an Equivalence class:

1. Reflexivity: $A \rightarrow A, B \rightarrow B$.
2. Symmetry: $A \rightarrow B, B \rightarrow A$.

The fundamental fact that is the base of our computational hypothesis of the Equivalence Class formation phenomenon is that now the neuron cannot differentiate

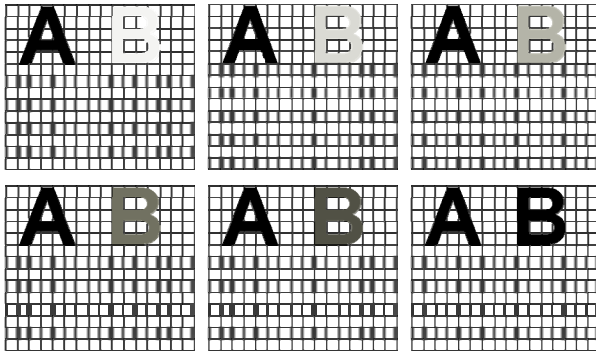


Fig. 3. A training sequence with reinforcement of a composite stimulus (A+B) (simultaneous presentation)

between the stimuli (A) or (B) and thus these stimuli becomes equivalent for the network. It is evident also that following the same procedure if we train a new associative relation (B+C) the stimulus (C) will form part of the internal representation of the neuron and by that reason (C) will be related of emergent form to stimulus (A) fulfilling the transitive property:

3. Transitivity: $A \rightarrow B, B \rightarrow C, A \rightarrow C$.

And also the Symmetry property between (A) and (C) is implicitly derived from this training process.

That is, the neuron will have learned an Equivalence Class:

$$C[A] = \{ A, B, C \} \tag{8}$$

From this time the set of stimuli $\{A,B,C\}$ will be equivalent for the computational model according to the Fig. 4.

Since as many neurons will exist as they are needed in the model it will be able to generate as many Equivalence Classes as needed. In a more realistic model what we call neuron would be formed by a neuronal group which is activated within a temporal frame this way the computational model would be a model of the conditional discrimination experiments.

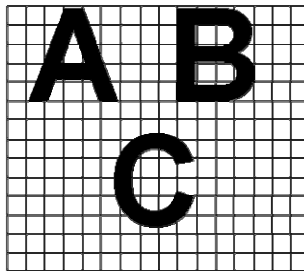


Fig. 4. Equivalence Class $C[A] = \{A,B,C\}$

4 Conclusions and Further Work

A Computational model of the Equivalence Class formation psychological phenomenon has been presented. We developed our model based on the SOFM ANN architecture which is used as a sort of associative memory. We have modelled the induction of Equivalence Classes and we have shown that the proposed model fulfills the basic set of characteristics of induced Equivalence Classes observed in documented psychological experiments realized in Humans and non-human Animals.

We have proposed some improvements to the presented model in order to handle in rigorous sense conditional discrimination experiments by means of a neuronal group which is activated and reinforced or punished within a temporal frame.

Acknowledgements. Authors wish to acknowledge the financial support of the UCM Project: PIMCD 582/2006.

References

1. T. Kohonen. Self-Organization and Associative Memory. Springer-Verlag, 3th ed., 1989.
2. T. Kohonen. The self-organizing map. *Proc. of the IEEE*, 78(9):1464-1480, 1990.
3. M. Sidman. Reading and auditory-visual equivalences. *J.Speech Hear.Res.*,14(1):5-13, 1971.
4. M. Sidman and W. Tailby. Conditional discrimination vs. matching-to-sample: An expansion of the testing paradigm. *J. Exp. An Behavior*, 37(1):5-22, 1982.
5. R. Sutton and A. Barto. *Reinforcement Learning, An Introduction*. MIT Press, 1998.
6. G. Venturini. Adaptation in dynamic environments through a minimal probability of exploration. In *SAB94: Proc. of the third Int. Conf. on Simulation of adaptive behaviour: from animals to animats 3*, pp. 371-379, Cambridge, MA, USA, 1994.

Data Security Analysis Using Unsupervised Learning and Explanations

G. Corral¹, E. Armengol², A. Fornells¹, and E. Golobardes¹

¹ Grup de Recerca en Sistemes Intel·ligents
Enginyeria i Arquitectura La Salle, Universitat Ramon Llull
Quatre Camins, 2, 08022 Barcelona, Spain
{guiomar, afornells, elisabet}@salle.url.edu

² IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Barcelona, Spain
eva@iia.csic.es

Abstract. Vulnerability assessment is an effective security mechanism to identify vulnerabilities in systems or networks before they are exploited. However manual analysis of network test and vulnerability assessment results is time consuming and demands expertise. This paper presents an improvement of *Analia*, which is a security system to process results obtained after a vulnerability assessment using artificial intelligence techniques. The system applies unsupervised clustering techniques to discover hidden patterns and extract abnormal device behaviour by clustering devices in groups that share similar vulnerabilities. The proposed improvement consists in extracting a symbolic explanation for each cluster in order to help security analysts to understand the clustering solution using network security lexicon.

Keywords: Network Security, Unsupervised Learning, Clustering, Explanations.

1 Introduction

The significant growth of networks and Internet has lead to increase security risks. As networks and networked systems become essential in any corporation, also their vulnerabilities become a main concern. Vulnerability assessment is the process of identifying and quantifying vulnerabilities in a system and it pursues two main goals: test everything possible and generate a concise report [1]. However, time and cost can limit the depth of a vulnerability assessment. These limitations justify the automation of the processes involved in a vulnerability assessment, not only those related to the testing phase, but also those related to the analysis of test results, as a thorough network security test generates extensive data quantities that need to be audited.

Logs collection, network traffic capture and potential threat identification are tasks difficult to handle when managing large data sets. Thus it is prohibitively expensive to classify it manually [2]. Artificial Intelligence (AI) techniques can help managing all this information and identifying subjacent patterns of behaviour. More specifically, unsupervised learning is suitable to handle vulnerability results as it does not need previous knowledge of data and can find relationships between tested devices.

In previous works we demonstrated the utility of clustering vulnerability assessment results [3,4], including partition methods [5] and soft computing solutions, like Self-Organization Maps (SOM) [6]. These solutions have been combined in *Consensus*, an integrated computer-aided system to automate network security tests [3,7], and this new system is called *Analia*. Security analysts obtain a configuration of different clusters, and every cluster contains tested devices with similar vulnerabilities and behaviour. When identifying and solving the vulnerabilities of a device, the same process can be applied to all devices in that cluster. Also main efforts can be focused first on the most critical clusters without having to analyze the whole data set. Analysts can evaluate results with clustering validity indexes. However they do not know the reasons of that clustering and have no explanation of the obtained solution.

This paper presents an improvement of *Analia* based on constructing explanations for clusters. This solution focuses on creating generalizations based on the anti-unification concept [8] to characterize each cluster. These generalizations permit describing a cluster with the same representation language used to describe the elements; therefore security analysts can more easily understand results. Analysts will obtain a solution where all tested devices have been grouped in different clusters regarding their vulnerabilities and will know the reason of these clustering results.

This paper is organized as follows: Section 2 surveys related work about the application of AI in network security. Section 3 explains the concept of symbolic description. Section 4 details the *Analia* system. Section 5 describes the experiments. Finally, Section 6 presents conclusions and further work.

2 Related Work

Considerable data quantities are compiled after performing a network security test and then a manual classification becomes an arduous work. AI techniques are useful in the analysis phase to handle vulnerability assessment results. Unsupervised learning techniques are particularly appropriate for this environment, where no previous knowledge of network behaviour and data results is required.

Clustering permits dividing data space into regions based on a similarity metric. Among many approaches, *K*-means [5] and SOM [6] highlight over the rest. *K*-means has been applied to group similar alarm records [9], for intrusion or anomaly detection [2,10], and for network traffic classification [11]. On the other hand, SOM has been used to detect anomalous traffic, intrusions, and classify attacks [12,13,14].

Another important aspect to be considered is the comprehension of the relationship between elements of an obtained cluster. For this reason, extraction of explanations from results is a key point. Symbolic descriptions [8] have been used in Case-based Reasoning systems to produce explanations on their performance and to organize their case memory [15]. In the present paper we adapt this idea to generate explanations in order to justify the clusters produced using an unsupervised technique.

3 Explanations

When clustering in a security environment, not only the distribution of tested devices in groups regarding their vulnerabilities is useful, but also the reason of that organization. This paper proposes a value-added service by including symbolic descriptions that explain why a subset of cases has been grouped together. These descriptions are based on the *anti-unification* concept [8] with some differences. The anti-unification of two objects is defined as the most specific generalization of both objects. It is a description with the attributes shared by both objects whose value is the most specific one. This paper considers the shared attributes among a set of objects, without using the most specific generalization of the values of these attributes.

Let C_i be a cluster and let e_1, \dots, e_n be the set of elements of that cluster after the application of a clustering algorithm included in *Analia*, like *K-means*, *X-means*, *SOM* or *Autoclass*. Each element e_j is described by a set of attributes A , where each attribute $a_k \in A$ takes values in V_k . We propose to describe the cluster C_i using a symbolic description D_i , with the following rules:

- D_i contains the common attributes to all the elements in C_i . Those attributes with unknown value in some element $e_j \in C_i$ are not considered in constructing D_i .
- Let a_k be an attribute common to all the elements in C_i taking values on a set V_k . The attribute a_k is not included in the description of D_i when the union of the values that a_k takes in the elements in C_i is exactly V_k .

Let us illustrate how to build a description D_i with an example in Table 1. Let C_i be a cluster formed by three elements, where e_1 , e_2 and e_3 are network devices whose attributes detail their open ports, possible operating systems and the number of detected security notes. Attributes common to all elements with values different from 0 are included in D_i , due to the fact that $a_k=0$ reflects an insignificant attribute in this domain. Notice that attribute ‘port 25’ is not in D_i because it takes all its possible values. A more detailed description of data representation is described in next section.

Table 1. Description of three elements from *Consensus* dataset and their explanation D_i

	Ports						W2000		XP	W2003	Security Notes
	21	25	53	80	135	445	SP3	SP4	SP2	Server	
e_1	1	0	0	0	1	1	41%	41%	41%	41%	3
e_2	1	2	1	0	1	1	41%	41%	41%	41%	6
e_3	1	1	0	1	1	1	41%	41%	41%	41%	6
D_i	1	--	--	--	1	1	41%	41%	41%	41%	(3,6)

4 Analia

Analia is the Analysis module of *Consensus* that introduces AI techniques to improve the results analysis after a network security test. *Consensus* automates the security testing procedures to reliably verify network security [7]. *Analia* uses *Consensus* to gather data and then applies unsupervised learning to help security analysts.

The main goal of *Analia* is to help finding hidden patterns in tested devices. After a security test, analysts must focus on every device in order to find abnormal behaviours, incorrect configurations or critical vulnerabilities. If the list of devices is extensive certain behaviours could become unnoticeable, some patterns could be masked or maybe the most vulnerable devices could be the last to be checked. *Analia* aids analysts to find similarities within data resulting from security tests. Then unsupervised learning helps analysts extracting conclusions without analyzing the whole data set. Next, best results are selected by using validity indexes when evaluating different executions. Finally, explanations justify clustering results.

This modular architecture offers different advantages. First, there is a separation between data collection and analysis. Network tests can be regularly planned and data is stored after every test. Whenever necessary, an analysis of the tested devices can be performed. *Analia* is a complementary module to refine results. *Analia* works with the same database where *Consensus* stores all data, avoiding data duplication. Its knowledge representation flexibility permits configuring different representations for the same data set as inputs to the clustering algorithms. Also the incorporation of new unsupervised learning techniques can be easily performed. Configuration files are obtained from the parameters selected by the analyst in the *Analia* web interface.

The main goal of clustering is to group elements with similar attribute values into the same class. In the clustering task, classes are initially unknown and they need to be discovered from data. The clustering process usually involves the following steps:

Pattern Representation. Patterns are multidimensional vectors, where each dimension is a feature. In a clustering context, with lacking class labels for patterns, the feature selection process is necessarily ad hoc. In *Analia*, every element is a tested device and features are characteristic data from a security test on that device. The main goal is to obtain groups of devices with similar vulnerabilities. Thus features should be related to data associated to vulnerabilities. Not all data stored in *Consensus* is suitable, as many testing tools return long data strings difficult for parametrization. Port scanning and operating system (OS) fingerprinting have been selected [3], as these processes determine most of the system vulnerabilities. Port scanning looks for multiple listening ports on a device and detects available services associated with those ports. OS fingerprinting remotely determines the OS of devices. Different knowledge representations can be configured in *Analia* combining results from these two processes [3]. An example is shown in Table 2, where OS features depict the probability of having that OS installed, and port features show when an open port (1), a filtered port (2), or no response (0) has been detected. An open port shows an available service; a filtered port means that a filter is preventing the system from determining whether it is open, and no response means that there was no reply from the host. Attributes shown in Table 2 are an example, as *Analia* includes all ports and OS detected in tested devices and they may vary for every vulnerability assessment.

Pattern Proximity Measure. The pattern proximity measure has been defined in every clustering algorithm. All of them implement the Euclidean distance, a special case of the Minkowski metric. It measures the distance between two tested devices.

Table 2. An example of knowledge representation with OS and open ports

Ports					Operating Systems				
23	25	53	80	...	Linux	Solaris	XP SP1	XP SP2	...
1	0	0	1	...	0.67	0.2	0.0	0.0	...

Clustering. Four clustering algorithms have been included in *Analia*: *K*-means [5], *X*-means [3], SOM [6] and AutoClass [16]. The goal is not to perform a system comparison, but to determine how well these methods perform over security data sets. Previous experiments have shown effective results [3,17]. Clustering eases analysis compared to a raw data set. Also analysts do not need previous knowledge about these techniques as they only have to select an algorithm and the desired number of clusters, depending on the technique. Although they can report different clustering results, all let deriving behaviour patterns and discovering modified devices.

Data Abstraction. This process extracts a simple and compact representation of a data set. A summary description of each cluster can be easy to comprehend and intuitively appealing for a security analyst. Generalizations have been obtained based on the anti-unification concept [8] to characterize each cluster, where a cluster is described using the same representation language that depicts data elements. For instance, *K-means* represents a cluster by its centroid and SOM, by its director vector. However, neither centroids nor director vectors is understandable information for security analysts. On the other hand, explanations use the same vocabulary of feature characterization.

Cluster Validation. This process assesses a clustering procedure’s output. It can not rely on given patterns as they do not exist in unsupervised domains. Different cluster validation techniques have been integrated in *Analia* in order to help analysts to check clustering results and compare different executions: Dunn [18], DB [19] and Silhouette [20] indexes. Also Cohesion indexes [17] have been designed ad hoc.

These steps have been analyzed and implemented in *Analia* in order to cluster data from a network security test in groups of devices with similar vulnerabilities.

5 Experiments

The main goal of the experimentation is to corroborate that not only clustering but also explanations let analysts obtain groups of devices with similar vulnerabilities and understand each cluster characterization. *Analia* can also help detecting unauthorized changes. When testing similar devices, clustering should group them. If an element is in another cluster, descriptions explain the changes between the modified and the rest.

Security tests have been executed over the university network to obtain data from real servers, alumni laboratories and staff computers. Alumni lab computers should follow the same pattern as any other software installation is forbidden. Therefore they all should be in the same cluster if nobody has illegally installed new software or changed their configuration. 44 devices have been tested: 21 (14+7) from two different labs, 9 public servers, 11 internal servers and 3 staff computers.

The pattern representation of Table 2 has been selected. *K*-means has been executed with *K* from 3 to 8, where *K* is the number of clusters to obtain. *X*-means has been configured to find the best *K* between 3 and 8. SOM has been configured with size maps of 2×2, 3×3, 4×4, 5×5 and 6×6. Autoclass has been configured to calculate the number of clusters automatically. Also 10 different random seeds have been used. Best executions have been selected using validity indexes (see previous section).

Figure 1(a) shows clustering results related to lab devices, represented by their IP addresses. Other cluster results have been omitted for security purposes as they have public IP addresses. Lab computers should belong to two different groups, as they have two different configurations. However three clusters represent these devices. Analysts can detect at a glance the suspicious device in cluster 5, but explanations in Table 3 will be needed to confirm it. Another perspective is shown in Fig. 1(b). After obtaining clustering results from all executions, a statistic of how many times a device has been clustered with every other device for all executions has been calculated. The arrow in Fig. 1(b) shows that device 28 (with IP address 10.0.14.203) has never been clustered with any other device for all selected executions. On the other hand, devices 1, 2, 3, 4 and 26 have been clustered always together and device 27 has been clustered with them in more than the 80% of the executions. They are the devices of cluster 1. Although analysts can easily detect an untrusting device, this information does not explain why this device is in a different cluster. However symbolic description of every cluster will give analysts this valuable information to quickly narrow the scope, isolate the device and apply the correction measures as soon as possible.

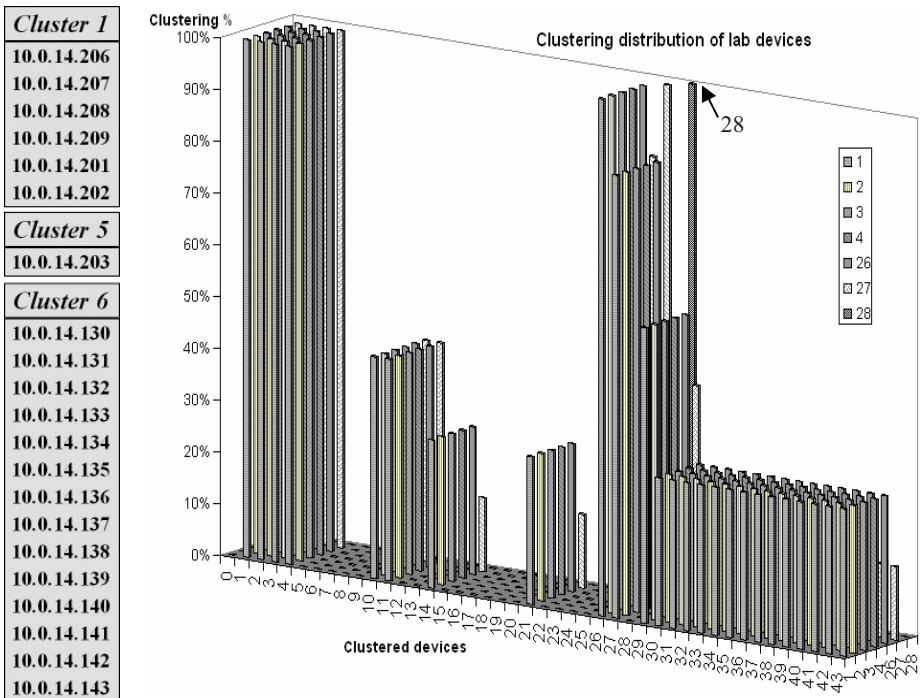


Fig. 1. (a) Clustering results with IP addresses of devices (b) Clustering distribution of devices

Table 3. Symbolic description of clusters containing lab devices

Cluster	Ports					W2000		XP		2003	
	135	139	445	781-807	904-930	WS_SP4	S_SP2	SP1	SP2	Standard	Enterp.
1	---	1	---	---	---	67%	67%	67%	67%	70%	70%
5	1	1	1	2	2	67%	67%	67%	67%	70%	70%

Table 3 shows the explanation of the studied clusters. These descriptions show that the OS of cluster 5 has not been modified with respect to cluster 1. However cluster 5 contains a high number of filtered ports (value=2). A filtered port means that a firewall, filter or some other network obstacle is blocking the port and preventing the system from determining whether it is open. Also ports 135 and 445 are open, which means that MSRPC¹ and Microsoft-DS² services have been altered. These ports should be blocked as they can be easily used to attack Windows computers. Therefore, analysts can promptly act to solve this situation without having to analyze the vulnerabilities of all lab computers.

6 Conclusions

Network vulnerability assessments generate many data and its analysis becomes a laborious work. By contrast, AI techniques are useful when dealing with large data sets. Unsupervised learning helps in the extraction of implicit, previously unknown and potentially useful information from data. Thus they are worthy to automate the classification of security data, thereby reducing the human effort required.

This paper has proposed *Analía*, a new system that includes unsupervised learning to cluster data obtained from a network vulnerability assessment. The system also introduces the concept of symbolic description to represent the main characteristics of the obtained clusters. The clustering process pursues grouping devices with similar vulnerabilities and discovering pattern behaviour regarding these vulnerabilities. Unsupervised learning can be very useful not only to identify similar devices, but also to find devices that unexpectedly appear separated and have new and different vulnerabilities. Explanations permit analysts to understand why some devices are in the same cluster or why they have been separated. This separation can be easily identified when a compact cluster with all the similar elements was expected. Also the relevant attributes of clusters are easily identified in order to detect the common vulnerabilities of all the devices included in a cluster. Therefore, when studying an element of a cluster, the obtained conclusions can be applied to the rest of its neighbours. Special efforts can be primarily focused on the most vulnerable clusters or the clusters where the most critical devices have been included.

Further work considers a deeper study on knowledge representation. Also an analysis of relations among the explanations of clusters will be studied in order to get profit of that information.

¹ MSRPC: Microsoft Remote Procedure Call.

² Microsoft-DS: Port used for file sharing in Microsoft Windows.

Acknowledgments. This work has been supported by the MCYT-FEDER project called MID-CBR (TIN2006-15140-C03-01 and TIN2006-15140-C03-03) and by the *Generalitat de Catalunya* under grants 2005SGR-302 and 2007FIC-00976. We would also like to thank *La Salle - Universitat Ramon Llull* for the support to our research group.

References

1. Peltier TR, Peltier J (2003) Managing a Network Vulnerability Assessment. CRC Press, Inc
2. Eskin E, Arnold A, Prerau M (2002) A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In: Data Mining for Security Applications
3. Corral G, Golobardes E, Andreu O, Serra I (2005) Application of clustering techniques in a network security testing system. AI Research and Development, IOS Press, 131:157–164
4. Fornells A, Golobardes E, Vernet D, Corral G (2006) Unsupervised case memory organization: Analysing computational time and soft computing capabilities. In 8th European Conference on CBR, LNAI Springer-Verlag, 4106:241-255
5. Hartigan J, Wong M (1979) A k-means clustering algorithm. Applied Statistics, 28:100–108
6. Kohonen T (1989) Self-Organization and Associative Memory. In: Springer Series in Information Sciences, Springer, Berlin, vol 8
7. Corral G, Zaballos A, Cadenas X, Grané A (2005) A distributed security system for an intranet. In 39th IEEE Carnahan Conference on Security Technology, pp 291-294
8. Armengol E, Plaza E (2000) Bottom-up induction of feature terms. Machine Learning 41(1):259-294
9. Julisch K (2003) Clustering intrusion detection alarms to support root cause analysis. ACM Transactions on Information and System Security 6:443–471 □
10. Leung K, Leckie C (2005) Unsupervised anomaly detection in network intrusion detection using clusters. In: Proceedings of 28th Australasian CS Conference, vol 38
11. Marchette D (1999) A statistical method for profiling network traffic. In: 1st USENIX Workshop on Intrusion Detection and Network Monitoring, pp 119–128
12. Ramadas M, Ostermann S, Tjaden BC (2003) Detecting anomalous network traffic with SOMs. In: 6th Symposium on Recent Advances in Intrusion Detection, 2820: 36–54
13. Depren M, Topallar M (2004) Network-based anomaly intrusion detection system using SOMs. In: IEEE 12th Signal Processing and Communications Applications, pp 76–79
14. DeLooze L (2004) Classification of computer attacks using a self-organizing map. In: Proceedings of the 2004 IEEE Workshop on Information Assurance, pp 365–369
15. Armengol E, Plaza E (2006) Symbolic Explanation of Similarities in CBR. Computing and Informatics 25:1001-1019
16. Cheeseman P, Stutz J (1996) Bayesian classification (autoclass): Theory and results. Advances in Knowledge Discovery and Data Mining, pp 153–180
17. Corral G, Fornells A, Golobardes E, Abella J (2006) Cohesion factors: improving the clustering capabilities of Consensus. Intelligent Data Engineering and Automated Learning, LNCS Springer, 4224:488–495
18. Dunn J (1974) Well separated clusters and optimal fuzzy partitions. Journal of Cybernetics 4:95–104
19. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Learning 4:224–227
20. Rousseeuw P (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20:53–65

Finding Optimal Model Parameters by Discrete Grid Search

Álvaro Barbero Jiménez, Jorge López Lázaro, and José R. Dorronsoro

Dpto. de Ingeniería Informática and Instituto de Ingeniería del Conocimiento
Universidad Autónoma de Madrid, 28049 Madrid, Spain
alvaro.barbero@iic.uam.es, jorge.lopez@iic.uam.es,
jose.r.dorronsoro@iic.uam.es

Abstract. Finding optimal parameters for a model is usually a crucial task in engineering approaches to classification and modeling tasks. An automated approach is particularly desirable when a hybrid approach combining several distinct methods is to be used. In this work we present an algorithm for finding optimal parameters that works with no specific information about the underlying model and only requires the discretization of the parameter range to be considered. We will illustrate the procedure's performance for multilayer perceptrons and support vector machines, obtaining competitive results with state-of-the-art procedures whose parameters have been tuned by experts. Our procedure is much more efficient than straight parameter search (and probably than other procedures that have appeared in the literature), but it may nevertheless require extensive computations to arrive at the best parameter values, a potential drawback that can be overcome in practice because of its highly parallelizable nature.

1 Introduction

In order to perform well many algorithms and modeling techniques in computer science rely on a careful choice of their parameters. Taking for instance machine learning and data mining, two well known models for classification and regression problems are Support Vector Machines (SVMs; [1, 11]) and Multilayer Perceptrons (MLPs; [2]). In the MLP case, possible parameters are the number of hidden layers and the number of hidden units (both discrete) or the weight decay rate (continuous). As for SVMs, typical parameters of interest are the penalty factor used in nonlinearly separable problems or the Gaussian kernel width.

Even if a given model is well suited to a concrete problem, incorrect choices of its parameters may result in a worse performance than expected. The usual approach to parameter choice is to combine some prior experience on the problem at hand with a limited heuristic search of possibly optimal parameters. However, when one wants to merge several independent models so that their combination presents a smaller variance than that of each individual model (as typically is the case with hybrid systems), one usually can no longer count on a deep experience on the application of each model to the problem at hand. When such a prior knowledge is absent or cannot be directly applied to the technique or algorithm that is being used (which henceforth will be referred to as the model), the optimal parameters have to be found either

through a more or less exhaustive search in parameter space or by using a blind or stochastic search metamodel, such as genetic algorithms [4, 8], genetic programming [5], or some kind of grid-search procedure [6, 12]. However, some of these metatechniques depend also on the choice of parameters for them such as, for instance, the number of individuals in a genetic algorithm, provoking thus a vicious circle.

An alternative is an intelligent search in the metamodel parameter space. A first step is to make a discretization of possible parameter values. In some cases, such as the number of hidden units in a Multilayer Perceptron (MLP), the model itself gives us such a discretization, but in some other cases (as, say, weight decay values for MLPs) it has to be imposed. The result of parameter discretization is the definition of a grid G of possible parameter values, that for the i -th parameter is determined by the range $[a_i, b_i]$ and the parameter resolution limit δ_i ; in other words, the values of parameter i to be explored are $\{a_i, a_i + \delta_i, \dots, a_i + H_i \delta_i\}$, with $H_i = (b_i - a_i) / \delta_i$.

An obvious (and very costly) way of exploring G is a linear search on all parameter dimensions keeping track of the parameter choices that lead to the best model. To alleviate the resulting huge computational effort, some kind of error metamodel may be introduced and used to select possibly better parameter values. For instance, it is possible [7] to obtain parameter-dependent error bounds for SVMs when leave-one-out estimations are used and some kind of gradient descent can be applied on the resulting error model to find optimal SVM parameters. On the other hand, a relatively standard technique (see [9], chapter 14) in the design of experiments field is to test the model performance in a few points of the parameter space, fit a first or second order empirical error model over the results obtained and move over this model towards the possible optimal parameter until convergence (or sometimes lack of fit) is reached. The empirical error models are taken as approximations to the real underlying error surface. As this process is repeated, the number and relative positions of the testing points usually change and may lead to a rather extensive number of models to be evaluated. For square penalty SVMs an intermediate approach has been proposed in [12] for the optimal selection of the Gaussian kernel width and the penalty constant. In this work we will extend this approach to a general metamodel, valid in principle for any parameter-dependent model setting for which M parameters are considered instead of the two used in [12], and that only needs as inputs the parameter ranges $[a_i, b_i]$ and the maximum parameter resolution δ_i to be used.

The paper is organized as follows. Section 2 contains the algorithm description as well as an analysis of its computational cost. In section 3 we shall apply the procedure to four classification problems from the Statlog repository [13], two regression problems from the Proben database [10], and another four classification problems used in [12] for comparison purposes, working in all cases with both MLPs and SVMs. For the Statlog datasets we shall show numerically that the optimal parameter MLP provided by our focused grid search always beats the best Statlog MLP result and either the grid MLP or SVM best model beats the best Statlog model for each problem. As for the Proben datasets, we obtain for the grid MLP and SVM models similar results to those of the Proben's best multilayer network, even improving them for some of the training-test settings. Finally, concerning the datasets used in [12] we will obtain similar numerical results while offering some computational improvements. The paper ends with a short conclusions section.

2 Parameter Vector Grid Search

As already mentioned, our method is based on the grid-search algorithm proposed in [12]. Instead of assuming a two-dimensional grid with two continuous parameters, we generalize the method to M possible discrete parameters. The parameter discretization is performed by fixing minimum and maximum values for each one, and by selecting a resolution level δ_i that determines how many possible values of the parameter will be considered for model construction. Assume that each parameter λ_i , $i = 1, \dots, M$, to be optimally chosen can be discretized to values in a range $\{a_i, a_i + \delta_i, \dots, a_i + H_i\delta_i = b_i\}$ and set $L_i = b_i - a_i$. For simplicity in the following discussion we will assume $H_i = 2^K$, although the algorithm will also work with other H_i values.

The algorithm starts by fixing the initial center c^0 with coordinates $c_i^0 = a_i + L_i/2$ and defining the left and right limits $\alpha_i^L = c_i^0 - L_i/2 = a_i$, $\alpha_i^R = c_i^0 + L_i/2 = b_i$ of the initial outer grid R_1 and, similarly, the left and right limits $\beta_i^L = c_i^0 - L_i/2^2$, $\beta_i^R = c_i^0 + L_i/2^2$ of the initial inner grid R_2 . The outer grid R_1 is then the set of the 3^M points with coordinates (p_1, \dots, p_M) where $p_i \in \{\alpha_i^L, c_i^0, \alpha_i^R\}$, and the inner grid R_2 is then the set of the 2^M points with coordinates (q_1, \dots, q_M) , where $q_i \in \{\beta_i^L, \beta_i^R\}$. This initial grid is depicted in figure 1. Now the model is evaluated at each of the $3^M + 2^M$ grid points, and the point τ_i giving the best model is used to define the new center c^1 as follows: $c_i^1 = \tau_i^1 + L_i/4$ if $\tau_i^1 = \alpha_i^L$ or $c_i^1 = \tau_i^1 - L_i/4$ if $\tau_i^1 = \alpha_i^R$ or $c_i^1 = \tau_i^1$ otherwise.

The new center c^1 is then used to define the new outer grid limits $\alpha_i^L = c_i^1 - L_i/2^2$, $\alpha_i^R = c_i^1 + L_i/2^2$ as well as the new inner grid limits $\beta_i^L = c_i^1 - L_i/2^3$, $\beta_i^R = c_i^1 + L_i/2^3$. A new optimal parameter vector τ^2 is then found and next grid center c^2 is defined as before. Note that the minimum resolution at the first iteration is $\rho_i^1 = L_i/4 = 2^{K-2}\delta_i$. Thus, the procedure will perform $K-1$ iterations until it reaches the limit resolution $\rho_i^{K-1} = 2^{K-1-(K-1)}\delta_i = \delta_i$. The final vector τ^{K-1} will give the optimal parameter values. While at the first iteration $3^M + 2^M$ model evaluations have to be done to arrive at τ^1 , note that they can be suitably cached when selecting the next optimal parameter vector τ^2 . This can be done at successive iterations avoiding thus model re-evaluation at the cached points that belong to the successive outer and inner grids. For instance, assuming the grid center to be the optimal parameter vector, we do not have to re-evaluate the model at the checked points in figure 1.

We analyze next the computational cost of the above grid search. Let us assume for simplicity that $\delta_i = \delta$ and also that $L_i = L = 2^K\delta$ for all $i = 1, \dots, M$. An exhaustive search would perform $1 + 2^K$ evaluations for each one of the M parameters. Its cost in number of evaluations would then be $(1 + 2^K)^M \approx 2^{KM} = (L/\delta)^M$.

On the other hand, grid search performs K iterations. In the first one 3^M outer and 2^M inner grid evaluations must be performed. The number of evaluations in subsequent iterations depends on the location of the best parameter vector τ^k . The best situation takes place, for instance, when τ^k is located in the outer grid center. As shown in figure 1 for two dimensional parameter vectors, the next iteration can reuse all the 2^M outer grid corners and its center, and will then require $3^M - 1$ new model evaluations. If we assume an optimal position for the new grid center at each iteration (for instance, if it is always the outer grid's geometrical center), the same will happen for all iterations except for the last one, in which the 2^M inner grid points are not evaluated since they are rounded to outer grid points (see figure 3). The overall number of model evaluations will thus be $3^M + 2^M + (3^M - 1)(K - 2) + (3^M - 2^M - 1) \approx K3^M = 3^M \log_2(L/\delta)$, which is substantially better than exhaustive search.

The worst case for grid search is depicted in figure 2 for two dimensional parameters, where the next grid iteration will require 5 model evaluations, one more than for the optimal two dimensional case. To simplify the somewhat complicated general analysis, we can assume $3^M + 2^M$ evaluations being done in each iteration (this would also be the case if no previous model evaluations have been cached). The overall number of model evaluations would then be $(3^M + 2^M)(K - 1) + 3^M \approx K(3^M + 2^M) = (3^M + 2^M) \log_2(L/\delta)$. While about twice larger than the best case complexity, this worst case is again much better than exhaustive search.

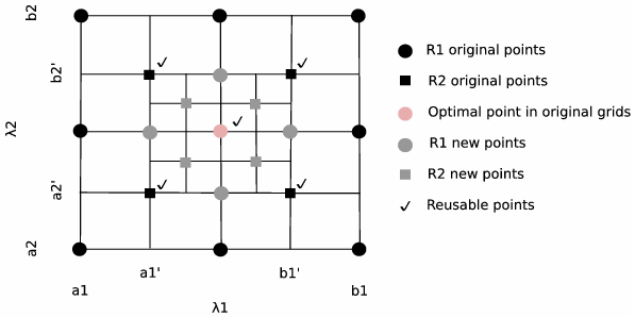


Fig. 1. Example of an iteration of the algorithm in the best case with two parameters. We show the initial configuration of the inner and outer grids R_1 and R_2 (in black) as well as the new grids which are built in the following iteration around the optimum (in gray) and the points that can be reused from the cache. Due to the limitations of the drawing software we used, the notation is different from the one used in the text: $a_i = \alpha_i^L$, $b_i = \alpha_i^R$, $a_i' = \beta_i^L$, $b_i' = \beta_i^R$.

3 Experimental Metamodeling of MLPs and SVMs

We will illustrate the preceding technique for both MLPs and SVMs with a Gaussian kernel applying them to several classification and regression tasks. As testing datasets we have chosen the Statlog's Australian credit, Pima Indian diabetes, DNA sequencing and image segmentation classification problems [13], the flare and building regression problems from the Proben database [10] and the Splice, Ringnorm, Waveform and Twonorm classification problems used in [12]. The MLP parameters to be searched were the number of hidden units (N_H) and weight decay factor (μ); the training method was conjugate gradient descent. For Gaussian kernel SVMs we searched in classification problems for the optimal kernel width σ and the penalty parameter C for slack margin variables; for regression problems we also considered the error insensibility parameter ε . SVM training was done for classification problems by an implementation of the Schlesinger-Kozinec (SK) algorithm [3] (if the number of classes is greater than 2 one SVM is trained for each output using a 1-of-c scheme for the targets); the SVMSeq algorithm [14] was used for regression problems. Discretization choices for the parameters used in both cases are shown in table 1. Notice that even in our proposed fast formulation, grid search is rather heavy computationally, making imperative to simplify as much as possible the iterated model evaluations.

Given the moderate size of our experiments, we have used 10-fold cross-validation on the training set for model evaluation at each grid point. Model quality was

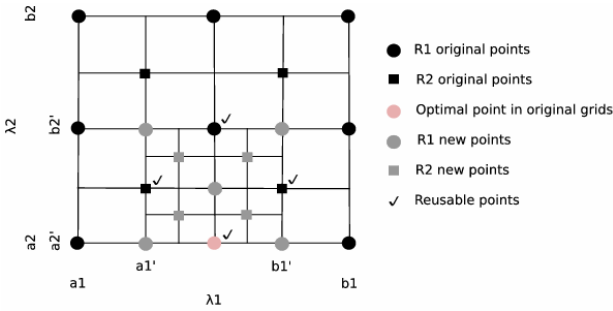


Fig. 2. Example of an iteration of the algorithm in the worst case with two parameters. We show the initial configuration of the inner and outer grids R_1 and R_2 (in black) as well as the new grids which are built in the following iteration around the optimum (in gray) and the points that can be reused from the cache.

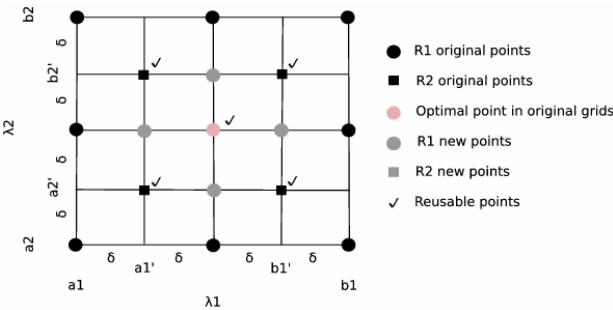


Fig. 3. The two last iterations of the algorithm with two parameters. Since we are in the precision limit δ after the approximation of the grid points to the discrete parameters, more points can be reused. After these two iterations the algorithm will stop, because a hypothetical next iteration would not be able to evaluate new points.

measured as either misclassification error or squared mean error, depending on the type of problem at hand. Once the optimal parameters have been computed, the performance of the resulting model has been tested in a way as close as possible to the one used for the reference results. Table 2 gives the results achieved over the Statlog datasets. It shows the test procedure used, the best misclassification error and the corresponding algorithm, as given in the Statlog web page, the misclassification error reported in that page for MLPs and, finally, the misclassification errors of the optimal parameter MLP and SVM. As it can be seen, the optimal parameter MLP clearly beats its Statlog counterpart and either it or the optimal parameter SVM model are quite close to the best Statlog algorithm.

Turning our attention to the regression problems, tests results for the Proben solar flare and building problems are shown in table 3. Three instances of these datasets appear in the Proben data collection, each one with their corresponding training and

Table 1. Discretization options for MLP and SVM parameters. While the algorithm searches for the best N_H value, in the rest of parameters their \log_{10} value is used, making a search according to their nature.

Parameter	a	b	δ
N_H	2	10	1
$\log_{10}\mu$	-10	-1	0.01
$\log_{10}C$	-3	3	0.1
$\log_{10}2\sigma^2$	-1	6	0.1
$\log_{10}\varepsilon$	-3	0	0.1

Table 2. Results of finding the parameters for a MLP and a SVM in four Statlog classification tasks. Error levels for the best Statlog result and Statlog backpropagation algorithm are presented for comparative purposes. Errors are computed as the ratio of misclassified examples.

Problem	Test m.	St. best m.	St. best err.	St. BP	MLP	SVM
Australian credit	10 CV	Cal5	13.1	15.4	13.4	13.3
Diabetes of Pima Indians	12 CV	LogDisc	22.3	24.8	22.3	23.0
DNA Sequence	Tr./test set	Radial	4.1	8.8	5.2	4.3
Image Segmentation	10 CV	Alloc80	3.0	5.4	5.1	3.5

test sets. In the solar flare problem our two predictors gave similar results, their performance being slightly worse for the flare1 problem, similar to the one reported for the flare3 problem and clearly better for flare2. The other dataset is the so-called building, whose objective is to predict the hourly consumption of electrical energy and hot and cold water in a building. In its original formulation (set 1) the problem is an extrapolation task: four consecutive months of data are used as training set, and the next two months give the test set. Sets 2 and 3 in the Proben collection are random permutations of the patterns, simplifying the problem to an interpolation task. As seen in table 3, our optimal parameter models (particularly the SVM) clearly outperform the best network in the extrapolation task (Proben1 dataset), and obtain competitive results in the other two problems.

Table 3. Comparison of Proben's results with those of optimal parameter MLPs and SVMs. For each task, three possible permutations of test and training set are used. Error levels for the best Proben1 results in a multilayer network are presented for comparative purposes. Errors are computed as squared error percentage (see [10]).

Problem	Proben1	MLP	SVM
flare1	0.5283	0.5472	0.5413
flare2	0.3214	0.2732	0.2606
flare3	0.3568	0.3423	0.3504
building1	0.6450	0.4267	0.4369
building2	0.2509	0.2696	0.2418
building3	0.2475	0.2704	0.2318

Finally we make a comparison with Staelin's results in [12] testing our procedure in four of its problems. A clear cut comparison is not easy, as no details are specified in [12] about the testing methodology. We have used 10-fold cross-validation in each problem to estimate the predictors' accuracy, except in the splice task, where training and test subsets are predefined. Results are presented in table 4. Our two optimal parameter models give similar accuracies in the Twonorm problem and are clearly better in the splice problem. Staelin's result is clearly better for the Waveform problem while it is similar to our optimal SVM accuracy in the Ringnorm problem; the MLP performance is in this case worse. Table 4 also shows the number of model evaluations each procedure does. It is seen that although MLPs perform fewer evaluations, that number is similar for Staelin's SVM and our optimal parameter version.

Table 4. Comparison of Staelin's grid search method and our proposed algorithm. Accuracy measured in ratio of misclassified patterns and number of model evaluations performed is shown for four classification problems used in [12]. Reference results are compared against finding the parameters for a MLP and a SVM by our method.

Problem	St. Acc.	St. #evals	MLP acc.	MLP #evals.	SVM acc.	SVM #evals.
splice	12.20	46	9.84	36	10.03	50
ringnorm	1.25	46	5.44	40	1.46	50
waveform	6.50	47	12.86	42	13.08	44
twonorm	2.00	47	2.22	43	2.19	52

4 Conclusions

Optimal parameter model finding is usually a cumbersome and tedious task that nevertheless becomes crucial from an engineering point of view, as different parameter choices may result in markedly different performances. An automated approach is quite desirable, particularly for a hybrid approach that combines several distinct methods, where it is very difficult, if not impossible, to achieve a good knowledge of each method's ability to handle a given problem.

We have presented an algorithm for finding optimal model parameters so that a minimum error rate is achieved. The algorithm works with no specific information about the underlying model and it only requires the discretization of the parameter range to be considered. We have illustrated the procedure's performance for two well-known and widely used classification and modeling methods, multilayer perceptrons and support vector machines. As shown in the previous section, our approach leads to results competitive with state-of-the-art procedures for which expert-tuned parameters are used.

Although our procedure is much more efficient than straight parameter search (and probably the same would be true when compared to other proposed approaches such as, say, genetic algorithms), it is nevertheless clear that it will also require extensive computations to arrive at the best parameter values. On the other hand, the proposed

method is highly parallelizable because in each iteration the points in the parameter space to be evaluated are independent and model performance in each of them can be computed on parallel threads.

References

1. Cristianini, N.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press (2000)
2. Duda, R., Hart P., Stork D.: Pattern classification. Wiley (2000)
3. Franc V., Hlaváč V.: An iterative algorithm learning the maximal margin classifier. *Pattern Recognition* 36, pp. 1985-1996 (2003)
4. Harpham C., Dawson C.W., Brown M.R.: A review of genetic algorithms applied to training radial basis function networks. *Neural Comput. & Applic.* 13, pp. 193-201 (2004)
5. Howley T., Madden M.: The genetic kernel support vector machine: description and evaluation. *Artificial Intelligence Review* 24, pp. 379-395 (2005)
6. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification, www.csie.ntu.edu.tw/~cjlin/libsvmtools
7. Keerthi, S.S.: Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Trans. Neural Networks* 13, pp. 1225-1229 (2002)
8. Kuo L.E, Melsheimer S.S.: Using genetic algorithms to estimate the optimum width parameter in radial basis function networks. *Proceedings of the American Control Conference, Baltimore* (1994)
9. Montgomery D.C.: Design and analysis of experiments. Wiley (1976)
10. Prechelt L.: Proben1: A set of neural network benchmark problems and benchmarking rules, <http://digbib.ubka.uni-karlsruhe.de/eva/ira/1994/21>
11. Schölkopf B., Smola A.J.: Learning with kernels. MIT Press (2001)
12. Staelin C.: Parameter selection for support vector machines. Technical Report HPL-2002-354. HP Laboratories, Israel (2002)
13. Statlog project datasets and results, <http://www.liacc.up.pt/ML/old/statlog/datasets.html>
14. Vijayakumar S., Wu S.: Sequential support vector classifiers and regression. *Neural Computation* 15, pp. 2643-2681 (2003)

A Hybrid Algorithm for Solving Clustering Problems

Enrique Domínguez and José Muñz

Department of Computer Science, E.T.S.I.Informática, Universidad de Málaga
Campus Teatinos s/n, 29071 Malaga, Spain
{enriqued, munozp}@lcc.uma.es

Abstract. There exist several approaches for solving clustering problems. Hopfield networks and self-organizing maps are the main neural approaches studied for solving clustering problems. Criticism of these approaches includes the tendency of the Hopfield network to produce infeasible solutions and the lack of generalization of the self-organizing approaches. Genetic algorithms are the other most studied bio-inspired approaches for solving optimization problems as the clustering problems. However, the requirement of tuning many internal parameters and operators is the main disadvantage of the genetic algorithms. This paper proposes a new technique which enables feasible solutions, removes the tuning phase, and improves solutions quality of clustering problems. Moreover, several biology inspired approaches are analyzed for solving traditional benchmarks.

1 Introduction

Clustering is a basic and widely applied methodology that has a variety of goals. Application fields include statistics, mathematical programming (such as location, selecting, networks partitioning, routing, scheduling, etc.) and computer science (including pattern recognition, learning theory, image processing and computer graphics, etc.)

Clustering data based on a measure of similarity is a critical step in scientific data analysis and in engineering systems. A common approach is to use data to learn a set of centers (prototypes or exemplars) such that the sum of squared errors between data points and their nearest centers is small. A popular clustering technique is the k-centers algorithm [1]. This approach consists of grouping a collection of objects (patterns) into several mutually exclusive clusters in order to achieve the optimum of an objective function. Clustering problems are rapidly becoming computationally intractable as problem scale increase, due to the combinatorial character of the method.

Solving clustering problems has been a core area in research for many communities in engineering, operations research, statistics and computer science. The interdisciplinary features of most clustering problems have caused a large amount of papers from many researches that have proposed numerous and different algorithms to overcome the many difficulties of such problems. Several researches used algorithms based on the model of organic evolution as an attempt to solve hard optimization problems [2]. Due to their representation scheme for search points, genetic algorithms (GAs) [3] are one of the most easily applicable representatives of evolutionary algorithms.

The idea of using neural networks (NNs) to provide solutions to NP-hard optimization problems have been pursued for over decades. Hopfield and Tank [4] showed that the travelling salesman problem (TSP) could be solved using a Hopfield neural network. This technique requires minimization of an energy function containing several terms and parameters. Due to this technique was shown to often yield infeasible solutions, researchers tried to either modify the energy function or optimally tune the numerous parameters involved so that the neural network would converge to a feasible solution.

Both techniques GAs and NNs are addressed in this work as hybrid paradigm for solving clustering problems. There exist many kinds of clustering problems; in this paper we selected the well-known p -medoid problem as a preliminary study of the proposed hybrid model for solving clustering problems. Good results are shown for different problem instances taken from the literature.

The outline of the paper is organized as follows: Section 2 presents the clustering problems and their typically formulations. Section 3 describes the proposed hybrid model for solving clustering problems. Experiment results are analyzed in section 4. Finally, the paper concludes in section 5 with a summary of the performance and characteristics.

2 Clustering Problems

A clustering method attempts to group the objects based on the definition of similarity supplied to it. Sometimes the data is represented directly in terms of the proximity between pairs of objects. This type of data can be represented by an $n \times n$ matrix D , where n is the number of objects. In the most cases, dissimilarity between the objects i and j is defined by expression

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^m w_k d(x_{ik}, x_{jk}) \quad (1)$$

where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{im})$ is the feature vector of the object i , w_k is a weight assigned to the k th attribute regulating the relative influence of that variable in determining the overall dissimilarity between objects. A possible choice is the squared distance (2) or the absolute distance (3).

$$d(x_{ik}, x_{jk}) = (x_{ik} - x_{jk})^2 \quad (2)$$

$$d(x_{ik}, x_{jk}) = |x_{ik} - x_{jk}| \quad (3)$$

In many clustering problems one is particularly interested in the characterization of the clusters by means of typical or representative objects that are called prototypes or exemplars. The object of the cluster for which the average dissimilarity to all the objects of the cluster is minimal is called *medoid*. That is, the object $\hat{\mathbf{y}}$ is a *medoid* of the cluster C if

$$\sum_{\mathbf{x} \in C} D(\mathbf{x}, \hat{\mathbf{y}}) = \min_{\mathbf{y} \in C} \sum_{\mathbf{x} \in C} D(\mathbf{x}, \mathbf{y}) \tag{4}$$

Thus, the clustering problem is formulated as follows

Minimize

$$\sum_{j=1}^n \sum_{k=1}^n \sum_{i=1}^c D(\mathbf{x}_j, \mathbf{x}_k) v_{ij} u_{ik} \tag{5}$$

Subject to

$$\sum_{i=1}^c v_{ij} = 1 \quad j = 1, 2, \dots, n \tag{6}$$

$$\sum_{k=1}^n u_{ik} = 1 \quad i = 1, 2, \dots, c \tag{7}$$

where

n is the number of objects

c is the number of clusters

$$v_{ij} = \begin{cases} 1 & \text{if the object } j \text{ is assigned to cluster } i \\ 0 & \text{otherwise} \end{cases}$$

$$u_{ik} = \begin{cases} 1 & \text{if the object } k \text{ is the prototype of cluster } i \\ 0 & \text{otherwise} \end{cases}$$

Note that the location variables u_{ik} set the prototypes of clusters, i.e. $\mathbf{y}_i = \mathbf{x}_k$ if and only if $u_{ik} = 1$.

3 Hybrid Model

The proposed hybrid model is based on both paradigms genetic algorithms and neural networks. A good domain exploration is an excellent feature of GAs, due to their multi-points search. In other sense, NNs presents a good direct search and a rapid simulation. Although both techniques GAs and NNs are suitable to parallelism architecture, the aim of this work is to analyze simple and fast bio-inspired techniques for solving clustering problems.

GAs are general-purpose search algorithms inspired by natural population genetics to evolve solutions to problems. The basic idea of GAs is to maintain a population of chromosomes that represent candidate solutions. A chromosome is composed of a series of genes that represent decision variables or parameters. Each individual of the population is evaluated and assigned a measure of fitness as a solution. There are three traditional genetic operators: selection, crossover and mutation.

The selection operator assigns the reproduction probabilities to the chromosomes based on their fitness. The crossover operator combines the features of two parent

individuals to form one or two offspring. The last operator, mutation, alters one or more genes of the offspring with a very low probability to avoid local optima.

In this work, we provide a new operator, so-called neural operator, based on applying a neural network to a selection of the individuals. The proposed neural network [5,6] has been applied successfully to several optimization problems obtaining good results. This operator is based on the recurrent neural network before mentioned. Basically, it is consist of applying the proposed recurrent neural network to the individuals. The solution represented by the individuals is the input of the neural network. Then, the neural network evolves according its dynamical rule and provides a new solution. In other words, the new neural operator evolves a deep search to obtain offspring with better fitness.

3.1 Coding

Individuals are formed by n allocation variables and c location variables. Indexes of the allocation variables represent the points, and their allocated cluster is represented by the value of the allocation variables. In the same way, location variables represent the prototype position of the clusters. Clusters are represented by the indexes of the location variables, and their values represent the prototype positions of the clusters. Note that the proposed coding scheme is used when the prototypes are located at the points.

3.2 Population Size

Population size is other parameter of GAs which has to be specified. This parameter is usually proportional to the domain search of the problem. Alp et al. [7] proposed a population size proportional to the number of solutions. Thus, large-scale problems need more computational time and larger memory space. Resend et al. [8] proposed a dynamic population size introducing an insert operator. The size of population is controlled by this new operator.

In our studied case, we propose a fixed population size using elitism. Several dynamic population sizes were tested for the studied instances, but no accuracy was obtained. Thus, a fixed population size of 100 individuals was used for all instances, since it is more efficient than a dynamic population size.

3.3 Neural Operator

In this subsection a new operator is described. Basically, the neural operator consist of applying the proposed neural network to the selected individuals, i.e. the solution represented by an individual is the input (initial state) of the neural network, what evolves according its neural dynamics and provides an offspring (final state) with better fitness. The proposed neural network consists of two layers of nc binary neurons or processing elements, where n is the number of patterns and c is the number of cluster. The first layer is named allocation layer and is composed of nc allocation neurons (v_{ij}) representing either the pattern j is assigned to cluster i or not. The

second layer is named location layer and is composed of nc location neurons (u_{ik}) indicating either the pattern k is the prototype of cluster i or not. Thus, the proposed neural network is formed by $2nc$ binary neurons, whose neuron states are defined according to the binary variables of the model (5)-(7). This neural model has been successfully applied for solving the p -median problem [9]. Moreover, this neural model can be generalized for solving clustering problems [10]. In this sense, the generalized neural network is used in our proposed hybrid model for solving clustering problems.

3.4 Mutation

Mutation operator is one of the most important genetic operators, since it allows avoiding the local optima altering a bit or integer (depending on whether a binary or integer coding is used) of the individuals. In this paper, a new mutation operator is proposed for clustering problems. The proposed mutation operator is based on the dispersion of the clusters or prototypes.

Generally, a good solution of a clustering problem uses to have a significant dispersion between clusters. In this sense, the proposed mutation operator replaces probabilistically a prototype of a solution (individual) by other one not included in the solution. Therefore, the proposed mutation operator consists of selecting the prototype of the individual to be replaced, and replacing the selected prototype by other one, what is not included in the solution. The probability of selecting the prototype \mathbf{y}_r to be replaced is defined by expression (8). And the probability of replacing the selected prototype \mathbf{y}_r by the object \mathbf{x}_k is given by expression (9).

$$P[\textit{selecting } \mathbf{y}_r] = \frac{1}{c} \tag{8}$$

$$P[\mathbf{y}_r = \mathbf{x}_k / \textit{selecting } \mathbf{y}_r] = \frac{\sum_{i=1}^c D(\mathbf{x}_k, \mathbf{y}_i)}{\sum_{j=1}^n \sum_{i=1}^c D(\mathbf{x}_j, \mathbf{y}_i)} \tag{9}$$

3.5 Algorithm

Both the population size and the selection probability are the unique parameters of the hybrid algorithm. Several configurations have been tested and are shown in the next section. The termination condition uses to be related to the number of iterations or any measure of the population like the improved fitness between iterations or certain mature rate. The proposed hybrid algorithm (HA) uses a fixed number of iterations, since it is the easiest and fastest termination criterion and to avoid storage problems in large-scale instances.

The initial population is composed of randomly generated feasible solutions. Mutation and neural operators can produce more adapted individuals. The population for the next iteration is composed of the best 100 individuals from the previous population together with the new offspring generated.

Although a low mutation probability (less than 0.2) is usual in the genetic algorithms implementation, higher values (greater than 0.5) are proposed in this work due to the characteristics of the new proposed mutation operator.

The neural operator is applied to all muted individuals, i.e. the muted individuals are the initial states of the abovementioned neural network and they evolve to better solutions according to the neural dynamics.

4 Experiment Results

The goal of the computational analysis is to evaluate the performance of the proposed hybrid algorithms for solving a classical clustering problem: the p -median problem. Initially, it is common sense to expect some advantage of traditional customized solvers. In this case we are interested in analyzing the efficiency of the hybrid techniques in relation to the effort of algorithm construction and the computational time. In this sense, all analyzed algorithms are very simple and easily implementable. Moreover, we are interested in the final accuracy of the solutions of the provided methods.

For the experiments, we have performed 50 independent runs of each algorithm in order to provide meaningful average values. We explicitly show the error of the best solution as a percentage of its distance to the known optimum value for each instance. Benchmarks are composed by 40 instances from 100 to 900 points and from 5 to 200 medians. The p -median instances are provided by OR-Library [11].

In table 1, three different algorithms are analyzed for the p -median problem: a simple genetic algorithm (GA), the aforementioned hybrid algorithm with a crossover operator (HA-x) and the proposed hybrid algorithm (HA). The instances optimally solved are not included in the table for better reading. The first two columns describe the problem name and its input parameters: the number of points and the number of medians to locate. Optimum values are included in the third column. Columns 4-6 show the cost of the best solution obtained by the analyzed algorithms, and the relative errors are presented in the columns 7-9.

A simple genetic algorithm (GA) with the traditional selection, crossover and mutation operators was analyzed to compare the performance against the new proposed operators (mutation and neural operators). In this sense, the proposed hybrid algorithm was modified by adding a crossover operator (HA-x). Both GA and HA-x were compared to the proposed hybrid algorithm (HA).

The experimental results show that the proposed neural operator is able of reporting better results. Although small-scale instances (not included in the table 1) are optimally solved by the three algorithms, the unique algorithm without neural operator (GA) reports a significant error for solving large-scale instances. Both hybrid algorithms HA-x and HA provide an average error of 0.66% and 0.33%, respectively, for the large-scale instances.

The fact that the crossover operator included in the modified hybrid algorithm (HA-x) produces worse results demonstrates that the general-purpose operators are not suitable for reporting accurate solutions.

Table 1. Comparing results for the p -median problem

Problem	(n, p)	Optimum	Cost			Error (%)		
			GA	HA-x	HA	GA	HA-x	HA
pmed4	(100,20)	3034	3181	3046	3034	4,85	0,40	0,00
pmed5	(100,33)	1355	1663	1359	1358	22,73	0,30	0,22
pmed8	(200,20)	4445	4681	4449	4445	5,31	0,09	0,00
pmed9	(200,40)	2734	3260	2758	2747	19,24	0,88	0,48
pmed10	(200,67)	1255	1809	1295	1270	44,14	3,19	1,20
pmed13	(300,30)	4374	4802	4391	4374	9,79	0,39	0,00
pmed14	(300,60)	2968	3722	2984	2974	25,40	0,54	0,20
pmed15	(300,100)	1729	2517	1783	1755	45,58	3,12	1,50
pmed18	(400,40)	4809	5456	4822	4814	13,45	0,27	0,10
pmed19	(400,80)	2845	3662	2877	2855	28,72	1,12	0,35
pmed20	(400,133)	1789	2813	1834	1809	57,24	2,52	1,12
pmed23	(500,50)	4619	5335	4640	4628	15,50	0,45	0,19
pmed24	(500,100)	2961	3903	2999	2978	31,81	1,28	0,57
pmed25	(500,167)	1828	2912	1890	1865	59,30	3,39	2,02
pmed28	(600,60)	4498	5322	4524	4507	18,32	0,58	0,20
pmed29	(600,120)	3033	4052	3083	3056	33,60	1,65	0,76
pmed30	(600,200)	1989	3193	2065	2028	60,53	3,82	1,96
pmed33	(700,70)	4700	5585	4715	4702	18,83	0,32	0,04
pmed34	(700,140)	3013	4171	3048	3035	38,43	1,16	0,73
pmed37	(800,80)	5057	6178	5089	5074	22,17	0,63	0,34
pmed40	(900,90)	5128	6197	5149	5151	20,85	0,41	0,45

5 Conclusion

In this work we are analyzed several configuration of the proposed hybrid model for solving a classical clustering problem: the p -median problem. The experiments results have shown that a simple genetic algorithm is accurate for small-scale instances (e.g. all instances with a number of medians lower than 20 have been solved exactly). The numerous tuning parameters to obtain good results are the main disadvantage of GAs.

Other heuristics applied to the clustering problems involve somewhat arcane analogies from the physical or biotic world, where many unintuitive parameters such as temperature, long and short term memory, or pheromone characteristics, must be devised, estimated and applied. Both the population size and the mutation probability are the unique parameter of the proposed hybrid model. Our experiments demonstrated that a population size of 100 individuals and a mutation probability greater then 0.5 are sufficient for obtaining good results.

Two new operators are provided for solving clustering problems. The proposed mutation and neural operators were analyzed under different configurations. Also, they were compared to traditional operators. The experimental results show that our hybrid model is able of reporting good results in a very scalable manner. Moreover,

the new proposed operators provided better results than the traditional operators. The fact that a simple hybrid model is able of computing accurate solutions along with its simplicity is an important point, since most authors never actually use algorithms reporting high accuracy due to they are difficult to implement or understand. Motivations for using GAs and NNs include improvement in speed of operation through massively parallel computation, and easy hardware implementation.

References

1. J. MacQueen: Some methods for classification and analysis of multivariate observations. In 5th Berkeley Symp. Mathematical Statistics and Probability, pp. 281--297 (1967)
2. J. H. Holland: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975).
3. D. E. Goldberg: Genetic algorithms in search, optimization and machine learning. Addison Wesley (1989).
4. J. Hopfield & D. Tank: Neural computation of decisions in optimization problems. Biological Cybernetics 52, 141--152 (1985).
5. E. Dominguez & J. Muñz: A recurrent neural network for location problems. In 50th Regional Science Association International, Philadelphia (2003).
6. E. Dominguez & J. Muñz: Hierarchical neural techniques for location problems. In IASTED Artificial Intelligence and Soft-Computing, Marbella, Spain (2004).
7. O. Alp, E. Erkut & Z. Drezner: An efficient genetic algorithm for the p-median problem. Annals of Operations Research 122, 21--42 (2003).
8. Resende & Werneck: A hybrid heuristic for the p-median problem. Journal of Heuristics 10 (1), 59--88 (2004).
9. E. Domínguez & J. Muñz: An efficient neural network for the p-median problem. LNAI vol. 2527, pp. 460--469. Springer (2002).
10. E. Dominguez & J. Muñz: Bidirectional neural network for clustering problems, LNAI vol. 3315, pp. 788—798. Springer (2004).
11. J. E. Beasley: OR-Library. Distributing test problems by electronic mail. Journal of the Operational Research Society 41 (11), 1069--1072 (1990).

Clustering Search Heuristic for the Capacitated p -Median Problem

Antonio Augusto Chaves, Francisco de Assis Correa, and Luiz Antonio N. Lorena

LAC – Laboratory of Computing and Applied Mathematics, INPE – National Institute for Space Research, 12227-010 São José dos Campos – SP, Brazil
{chaves, lorena}@lac.inpe.br, fcorrea@directnet.com.br

Abstract. In this paper we present a hybrid heuristic for the capacitated p -median problem (CPMP). This problem considers a set of n points, each of them with a known demand, the objective consists of finding p medians and assign each point to exactly one median such that the total distance of assigned points to their corresponding medians is minimized, and the a capacity limit on the medians may not be exceeded. The purpose of this paper is to present a new hybrid heuristic to solve the CPMP, called Clustering Search (CS), which consists in detecting promising search areas based on clustering. Computational results show that the CS found the best known solutions in all most instances.

Keywords: p -Median Problem, Hybrid heuristics, Clustering Search.

1 Introduction

This paper presents a new hybrid heuristic to solve the Capacitated p -Median Problem (CPMP). The CPMP is a classical location problem with various applications in many practical situations. It can be described as follows: given a set of n points (customers), each of them with a known demand, the problem consists of finding p medians (centers) and assign each point to exactly one median such that the total distance of assigned points to their corresponding medians is minimized, and the capacity limit on the medians may not be exceeded.

The CPMP also appears under the names Capacitated Clustering Problem, Capacitated Warehouse Location Problem, Sum-of-Stars Clustering Problem and others. Various heuristics and metaheuristics have been proposed for these problems, which are known to be NP-hard [4]. Osman and Christofides [11] propose a simulated annealing and tabu search method. Maniezzo *et al.* [7] present a bionomic algorithm to solve this problem. Lorena and Senne [6] explore local search heuristics based on location-allocation procedures and Lorena and Senne [5] use column generation to CPMP. Diaz and Fernández [2] examine a hybrid scatter search and path-relinking method and Scheuerer and Wendolsky [12] a scatter search method. Recently, Fleszar and Hindi [3] propose a variable neighborhood search heuristic and Osman and Ahmadi [10] investigate a guide construction search metaheuristic based on a periodic local search procedure or a greedy random adaptive construction search procedure (GRASP) to solve the CPMP.

The CPMP considered in this paper is modeled as the following binary integer programming problem:

$$z = \min \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij} \quad (1)$$

$$\text{subject to } \sum_{j \in N} x_{ij} = 1, \quad \forall i \in N \quad (2)$$

$$\sum_{j \in N} x_{jj} = p \quad (3)$$

$$\sum_{i \in N} q_i x_{ij} \leq Q x_{jj}, \quad \forall j \in N \quad (4)$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in N, \forall j \in N \quad (5)$$

where:

- $N = \{1, \dots, n\}$ is the index set of points to allocate and also of possible medians, where p medians will be located;
- q_i is the demand of each point and Q the capacity of each possible median;
- d_{ij} is a distance matrix;
- x_{ij} is the allocation matrix, with $x_{ij} = 1$ if point i is allocated to median j , and $x_{ij} = 0$, otherwise; $x_{jj} = 1$ if median j is selected and $x_{jj} = 0$, otherwise.

The objective of the CPMP is expressed in (1). Constraints (2) impose that each point is allocated to exactly one median. Constraint (3) set the number of medians to be located. Constraint (4) imposes that a total median capacity must be respected, and (5) provides the integer conditions.

In this paper, we propose a hybrid heuristic for the CPMP, known by Clustering Search (CS). The CS, proposed by Oliveira and Lorena [9], consists in detecting promising areas of the search space using a metaheuristic that generates solutions to be clustered. These promising areas should be explored through local search methods as soon as they are discovered.

The remainder of the paper is organized as follows. Section 2 describes the basic ideas and components of CS. Section 3 present the CS applied to CPMP. Section 4 presents the computational results and conclusions are presented in Section 5.

2 Clustering Search

The Clustering Search (CS) generalizes the Evolutionary Clustering Search (ECS) proposed by Oliveira and Lorena [9] that employs clustering for detecting promising areas of the search space. It is particularly interesting to find out such areas as soon as possible to change the search strategy over them.

In the ECS, a clustering process is executed simultaneously to an evolutionary algorithm, identifying groups of individuals that deserve special interest. In the CS,

the evolutionary algorithm was substituted by distinct metaheuristics, such as Tabu Search, Variable Neighborhood Search or Simulated Annealing.

The CS attempts to locate promising search areas by framing them by clusters. A cluster can be defined as a tuple $G = \{c; r; s\}$ where c , r and s are, respectively, the center and the radius of the area, and a search strategy associated to the cluster.

The center c is a solution that represents the cluster, identifying the location of the cluster inside of the search space. The radius r establishes the maximum distance, starting from the center, that a solution can be associated to the cluster. The search strategy s is a systematic search intensification, in which solutions of a cluster interact among themselves along the clustering process, generating new solutions.

The CS is hybrid metaheuristic that consists of four conceptually independent components with different attributions: a search metaheuristic (SM); an iterative clustering (IC); an analyzer module (AM); and a local searcher (LS).

The SM component works as a full-time solution generator. The algorithm is executed independently of the remaining components and must to be able provide the continuous generation of solutions directly to the clustering process. Simultaneously, clusters are kept to represent these solutions. This entire process works like an infinite loop, in which solutions are generated along the iterations.

The IC component aims to gather similar solutions into groups, keeping a representative *cluster center* for them. To avoid extra computational effort, IC is designed as an online process, in which the clustering is progressively fed by solutions generated in each iteration of SM. A maximum number of clusters NC is an upper bound value that prevents an unlimited cluster creation. A distance metric must be defined, a priori, allowing a similarity measure for the clustering process.

The AM component provides an analysis of each cluster, in regular intervals, indicating a probable promising cluster. A cluster density, δ_i , is a measure that indicates the activity level inside the cluster. For simplicity, δ_i counts the number of solutions generated by SM and allocated to the cluster i . Whenever δ_i reaches a certain threshold, such cluster must be better investigated to accelerate the convergence process on it.

At last, the LS component is a local search module that provides the exploitation of a supposed promising search area, framed by cluster. This process happens after AM finds a promising cluster and the local search is applied on the cluster center. LS can be considered as the particular search strategy s associated with the cluster, i.e., a problem-specific local search to be employed into the cluster.

3 CS for CPMP

A version of CS for CPMP is presented in this section. The application details are now described, clarifying this approach. The component SM, responsible for generating solutions to clustering process, was a Simulated Annealing (SA) metaheuristic [13], which is capable to generate a large number of different solutions for this process. The others components of CS are also explained in the following.

3.1 Simulated Annealing

In the component SM, the metaheuristic used to generate solutions to clustering process is based on the Simulated Annealing (SA) [13]. The algorithm starts from a random initial solution which is obtained choosing randomly the medians and assigned the points to the closer median that not exceed the capacity of it. The next step follows the traditional simulated annealing algorithm schema. Given a temperature T , the algorithm randomly selects one of the moves to a neighborhood and computes the variation of the objective function. If it improves the current solution the move is accepted, otherwise there is a probability of acceptance that is lower in low temperatures.

Four different moves have been defined to compose distinct kinds of neighborhood, named N^1 , N^2 , N^3 and N^4 , from a solution s . N^1 is obtained by swapping the allocation of two points of different medians. N^2 is obtained by swapping a median with an assigned point to it. N^3 is obtained by dropping a point of a median and add in other median. And N^4 is obtained by swapping a median with any other point that is not a median.

The parameters of control of the procedure are the rate of cooling or decrement factor α , the number of iterations for each temperature (SA_{max}) and the initial temperature To . In this paper, we use $\alpha = 0.95$, $SA_{max} = 1000$ and $To = 1000000$.

3.2 The CS Application

The IC is the CS's core, working as a classifier, keeping in the system only relevant information, and driving the search intensification in the promising search areas. Initially, all clusters are created randomly ($NC = 20$), the i^{th} cluster has its own center c_i , and a radius r that is identical to the other clusters.

Solutions generated by SA are passed to IC that attempts to group as known solution, according to a distance metric. If the solution is considered sufficiently new, it is kept as a center in a new cluster. Otherwise, redundant solution activates the closest center c_i (cluster center that minimizes the distance metric), causing some kind of perturbation on it. In this paper, the metric distance was the number of points assigned to different medians in the solutions of the SA and the cluster center, and a larger distance imply in more dissimilarity.

Perturbation means an assimilation process, in which the cluster center is updated by the new generated solution. Here, we used the path-relinking method [8], that generates several points (solutions) taken in the path connecting the solution generated by SA and the cluster center. The assimilation process itself is an intensification mechanism inside the clusters. The new center c_i' is the best evaluated solution sampled in the path.

Path-relinking starts from two solutions. The first is the solution that comes from the SA (*initial*). The second is the closest cluster center c_i (*guide*). Starting from the first to the second solution, paths are generated and explored in the search for better solutions. To generate paths, moves are selected by changing one median of the *initial* by one from the *guide*, changing the allocation solution. The best solution in one move is defined as the *new initial*.

The AM is executed whenever a solution is assigned to a cluster, verifying if the cluster can be considered promising. A cluster becomes promising when reaches a certain density δ_i ,

$$\delta_i \geq PD \cdot \frac{NS}{|Clus|} \quad (6)$$

where, NS is the number of solutions generated in the interval of analysis of the clusters, $|Clus|$ is the number of cluster, and PD is the desirable cluster density beyond the normal density, obtained if NS was equally divided to all clusters. In this paper, we use $NS = 100$ and $PD = 2$.

The component LS is activated when the AM discover a promising cluster. The LS uses the Location-Allocation heuristic [6], which seeks to improve the center of the promising cluster. This heuristic is based on the observation that the cluster center have p medians and their allocated points, and, this solution can be improved by searching for a new median, swapping the current median by a non-median point assigned to it, and reallocating. We consider two steps for reallocating the points. The first one is to examine the points that were allocated to the current median and reallocate to the closest one. The second step is to examine the points assigned to the others medians and calculate the *saving* of moving them to the new one, if it improves the solution the point is reallocated to the new median. If the solution is improved, the process can be repeat until no more improvements occur.

The whole CS pseudo-code is presented in following.

Procedure CS

```

Create Initial Solution (s)
Create Initial Clusters
IterT = 0
T = To
while (T > 0.0001)
  while (IterT < SAmix)
    IterT = IterT + 1
    Generate at random  $s' \in N^k(s)$ 
     $\Delta = f(s') - f(s)$ 
    if ( $\Delta < 0$ )
       $s = s'$ 
    else
      Let  $x \in [0, 1]$ 
      if ( $x < e^{-\Delta/T}$ )
         $s = s'$ 
      end-if
    component IC ( $s'$ )
    component AM (active cluster)
    if (active cluster is promising) then
      component LS (cluster center)
    end-while
  T = T x  $\alpha$ 
  IterT = 0
end-while
end-CS.
```

4 Computational Results

The CS was coded in C++ and the computational tests executed on a Pentium IV 3.02 GHz. Two problem sets are used in this tests: a classical set introduced by Osman and Cristofides [11], that contains 10 problems of size $n = 50$ and $p = 5$, and 10 problems of size $n = 100$ and $p = 10$ (these problems are named of $p1$ to $p20$), and a set of real data collected at the São José dos Campos city introduced by Lorena and Senne [5], that contains 6 problem instances named “ sjc ”. Those instances can be downloaded from or through the OR-Library.

Table 1 gives the obtained results for each instance, comparing the performance of the CS and the SA without the clustering process. Column *best* gives the values of the best known solutions found in literature. The best solutions found by the approaches (*sol**), the time to obtain the best solution (*Time**) in seconds, the average of solutions (*sol*), and the total time (*Time*) in seconds are used to compare the CS and SA. Each instance has been run 10 times. The improvement of the CS in relation to SA is reported in terms of the relative percentage deviation (RPD).

Table 1. Results of the CS

ID	CS					SA			RPD
	best	sol*	Time*	sol	Time	sol*	sol	Time	
p1	713	713	0.02	713.00	2.24	713	721.60	1.56	1.21
p2	740	740	0.01	740.00	2.34	740	740.00	1.59	0.00
p3	751	751	0.08	751.00	2.29	751	751.60	1.46	0.08
p4	651	651	0.03	651.00	2.24	651	651.20	1.52	0.03
p5	664	664	0.03	664.00	2.35	664	664.40	1.51	0.06
p6	778	778	0.01	778.00	2.35	778	778.00	1.57	0.00
p7	787	787	0.01	787.00	2.34	787	788.60	1.55	0.20
p8	820	820	0.08	820.00	2.38	821	825.80	1.60	0.71
p9	715	715	0.02	715.00	2.36	715	717.80	1.46	0.39
p10	829	829	2.09	829.00	2.36	829	833.80	1.58	0.58
p11	1006	1006	7.97	1006.00	35.24	1007	1015.40	10.11	0.93
p12	966	966	0.05	966.00	49.39	968	970.20	12.74	0.43
p13	1026	1026	0.06	1026.00	41.05	1026	1028.80	10.00	0.27
p14	982	982	19.80	982.80	39.14	985	997.20	10.35	1.47
p15	1091	1091	1.30	1091.00	44.98	1092	1102.60	12.15	1.06
p16	954	954	15.94	954.00	36.03	957	960.40	10.37	0.67
p17	1034	1034	19.50	1034.20	48.18	1037	1043.00	12.24	0.85
p18	1043	1043	48.62	1044.60	46.39	1045	1048.20	11.41	0.34
p19	1031	1031	10.17	1031.80	38.56	1034	1042.80	10.09	1.07
p20	1005	1005	5.41	1005.40	48.77	1009	1015.20	12.45	0.97
sjc1	17288.99	17288.99	0.23	17288.99	22.80	17288.99	17343.96	10.43	0.32
sjc2	33270.94	33270.94	13.25	33275.43	120.40	33372.98	33491.75	25.70	0.65
sjc3a	45335.16	45335.16	405.50	45337.34	859.09	46746.68	47110.93	52.51	3.91
sjc3b	40635.90	40635.90	1626.52	40643.67	1649.41	41551.34	41888.14	54.01	3.06
sjc4a	61925.51	61928.72	938.45	62017.51	2601.81	63710.71	64574.92	77.19	4.10
sjc4b	52469.96	52531.27	1402.25	52540.67	7233.59	53789.61	54716.58	92.05	4.14

CS found the best known solutions for most of the instances. Except for *sjc4a* and *sjc4b*. The running times of the CS were very competitive, found better solutions in small times. The CS is very robust, producing average solutions close to the best known ones. The SA, without the clustering process, has worse results than the CS in quality of solutions, but the times are smaller. All SA solutions were improved by CS.

Table 2 present a comparison of the CS solutions with the results of two heuristics that have the best performing in the literature, the first is a scatter search with path-relinking (SS-PR) [12] and the second is a variable neighborhood search (VNS) [3]. Note that CS was very competitive to the SS-PR and VNS, failing in find the best known solution only for two instances. For the *sjc* instances, the total time of CS was smaller than the others heuristics in three instances.

Table 2. Comparison of the results

ID	SS-PR			VNS		CS	
	best	sol*	Time	sol*	Time	sol*	Time
p1	713	713	6	713	0.17	713	2.24
p2	740	740	6	740	0.05	740	2.34
p3	751	751	6	751	0.19	751	2.29
p4	651	651	6	651	0.11	651	2.24
p5	664	664	6	664	0.27	664	2.35
p6	778	778	6	778	0.11	778	2.35
p7	787	787	6	787	0.31	787	2.34
p8	820	820	6	820	0.92	820	2.38
p9	715	715	6	715	0.13	715	2.36
p10	829	829	6	829	0.75	829	2.36
p11	1006	1006	60	1006	7.91	1006	35.24
p12	966	966	60	966	4.81	966	49.39
p13	1026	1026	60	1026	2.17	1026	41.05
p14	982	982	60	982	10.33	982	39.14
p15	1091	1091	60	1091	10.23	1091	44.98
p16	954	954	60	954	4.20	954	36.03
p17	1034	1034	60	1034	5.50	1034	48.18
p18	1043	1043	60	1043	9.06	1043	46.39
p19	1031	1031	60	1031	8.64	1031	38.56
p20	1005	1005	60	1005	27.34	1005	48.77
sjc1	17288.99	17288.99	60	17288.99	50.50	17288.99	22.72
sjc2	33270.94	33293.40	600	33270.94	44.08	33270.94	112.81
sjc3a	45335.16	45338.02	2307	45335.16	8580.30	45335.16	940.75
sjc3b	40635.90	40635.90	2308	40635.90	2292.86	40635.90	1887.97
sjc4a	61925.51	61925.52	6109	61925.51	4221.47	61928.72	2885.11
sjc4b	52469.96	52531.46	6106	52469.96	3471.44	52531.27	7626.33

5 Conclusions

This paper has presented a solution for the Capacitated p-Median Problem (CPMP) using Clustering Search (CS). The CS is a new method that has been applied with

success in some combinatorial optimization problems, such as pattern sequencing problem [9] and prize collecting traveling salesman problem [1]. The results show that the CS approach is competitive for the resolution of this problem in reasonable computational times. For the first set of instances considered in computational tests, the optimal values have been found, and for instances of the second set the best values known have been found in most of cases. Therefore, these results validate the CS application to the CPMP, and this approach has an additional advantage: does not use any commercial solver.

Further works can be done by analyzing other metaheuristics to generate solutions for the clustering process of CS, such as Ant Colony System, Tabu Search and Genetic Algorithm, and by implementing new local search heuristics for the CPMP. Besides that, bigger instances of this problem can be generated and solved.

References

1. Chaves, A.A. and Lorena, L.A.N.: Hybrid algorithms with detection of promising areas for the prize collecting traveling salesman problem. Fifth international conference on hybrid intelligent systems (2005) 49–54
2. Diaz, J.A. and Fernandez E.: Hybrid scatter search and path relinking for the capacitated p -median problem. *European Journal of Operational Research*, Vol. 169 (2006) 570 – 585
3. Fleszar, K. and Hindi, D.S.: An effective VNS for the capacitated p -median problem. *European Journal of Operational Research* (2007), doi:10.1016/j.ejor.2006.12.055
4. Garey, M.R., Johnson, D.S.: *Computers and intractability: A guide to the theory of np-completeness*, W.H. Freeman and Co. (1979)
5. Lorena L.A.N. and Senne E.L.F.: A column generation approach to capacitated p -median problems. *Computers and Operational Research*, Vol. 31 (1994) 863–876
6. Lorena L.A.N. and Senne E.L.F.: Local search heuristics for capacitated p -median problems, *Networks and Spatial Economics*, Vol. 3 (4) (2003) 407–419
7. Maniezzo V, Mingozzi A and Baldacci R.: A bionomic approach to the capacitated p -median problem. *J Heuristics*, Vol. 4 (1998) 263–280
8. Laguna, M. and Martí, R.: *Scatter Search—Methodology and Implementations in C*, Kluwer Academic Publishers, Boston (2003)
9. Oliveira, A.C.M. and Lorena, L.A.N.: Detecting promising areas by evolutionary clustering search.. *Advances in Artificial Intelligence*. Springer Lecture Notes in Artificial Intelligence Series (2004) 385–394
10. Osman, I.H., Ahmadi, S.: Guided construction search metaheuristics for the capacitated p -median problem with single source constraint, *Journal of the Operational Research Society*, (2006) 1–15
11. Osman, I.H., Christofides, N.: Capacitated clustering problems by hybrid simulated annealing and tabu search. *Intern. Trans. in Operational Research*, Vol.1 (3) (1994) 317–336
12. Scheuerer S. and Wendolsky R.: A scatter search heuristic for the capacitated clustering problem. *European Journal of Operational Research*, Vol. 169 (2006) 533–547
13. S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi, *Optimization by Simulated Annealing*, *Science*, Vol. 220 (4598) (1983) 671–680

Experiments with Trained and Untrained Fusers

Michal Wozniak

Chair of Systems and Computer Networks, Wroclaw University of Technology
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
michal.wozniak@pwr.wroc.pl

Abstract. The *Multiple Classifier Systems* are nowadays one of the most promising directions in pattern recognition. There are many methods of decision making by the ensemble of classifiers. The most popular are methods that have their origin in vote method, where the decision of the common classifier is a combination of individual classifiers decisions. This work presents method of weighted classifiers combination and experimental results of proposed algorithm.

1 Introduction

The concept of the *Multiple Classifier Systems* has been known for over 15 years [1]. Some works in this field were published as early as the '60 of the XX century [2], when it was shown that the common decision of independent classifiers is optimal, when chosen weights are inversely proportional to errors made by classifiers. In many review articles this trend has been mentioned as one of the most promising in field of the pattern recognition [3]. In the beginning in literature one could find only the majority vote, but in later works more advanced methods of receiving common answer of the classifier group were proposed. Attempt to estimate classification quality by the classifier committee is one of essential trend. Known conclusions, derived on the analytic way, concern particular case of the majority vote [4] when classifier committee is formed from independent classifiers. Unfortunately this case has only theoretical characteristic and is not useful in practice. On the other hand the weighted vote is taken into consideration [5]. In this work it was stated that the optimal weight value should be dependent on the error of the single classifier and on the *prior* probability of the class, on which classifier points. One also has to mention many other works, that describe analytical properties and experimental results, like [6-8]. Paper presents method of weighted voting procedure based on errors of individual classifiers and its quality is compared to another voting schemes.

The problem of establishing weights for mentioned voting procedure is not simple. Many of authors have proposed treating the block of voting as kind of classifier [19,20] but the general question is “does fuser need to be trained?” [23]. The one of the most popular methods of learning the voting block is so-called *stacked generalization* [21,22] using the cross validation method. The idea of voting method is depicted in Fig.1.

The content of the work is as follows: next section shortly introduces into necessary background. Section 3 proposes how calculate weights of classifiers by

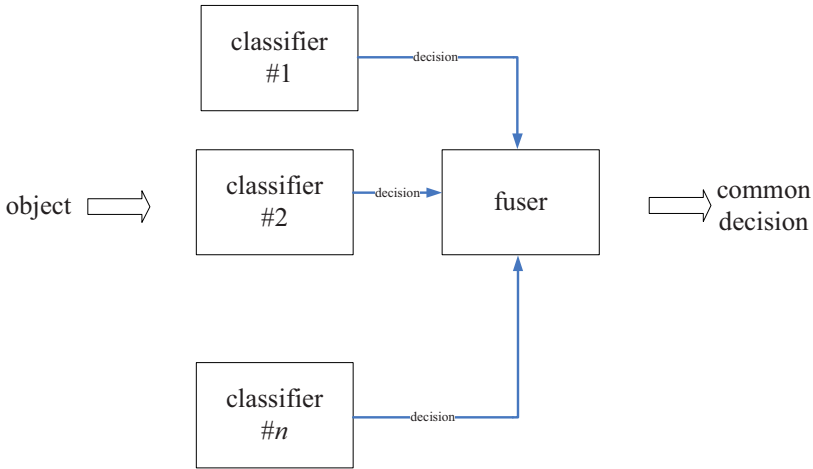


Fig. 1. Idea of voting method

original adaptive procedure. In section 4 the results of experimental investigation of methods under consideration are presented. The last section concludes the paper.

2 Problem Statement

Let us assume that we have n classifiers $\Psi^{(1)}, \Psi^{(2)}, \dots, \Psi^{(n)}$. Each of them decides if object belongs to class $i \in M = \{1, \dots, M\}$. For making common decision by the group of classifiers we use following common classifier $\bar{\Psi}$:

$$\bar{\Psi} = \arg \max_{j \in M} \sum_{i=1}^k \delta(j, \Psi_i) w_i \Psi_i, \quad (1)$$

where w_i is the weight of i th classifier and

$$\delta(i, j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}. \quad (2)$$

Let us note that the weights could play the role of the quality of classifier $\Psi^{(i)}$. The accuracy of classifier $\bar{\Psi}$ is maximized by assigning weights

$$w_i \propto \frac{P_{a,i}}{1 - P_{a,i}}, \quad (3)$$

where $P_{a,i}$ denotes probability of accuracy of i th classifier. Unfortunately it is not sufficient for guarantying the smallest classification error. The *prior* probabilities for each class have to be taken into account also [5]. In real decision problems the values of the *prior* probabilities are usually unknown.

3 Adaptive Weights Calculation Algorithm

There is the problem how to calculate the weights of simple classifiers. In [5,4,21] authors proposed to learn the fuser. According these suggestions we propose AWC algorithm (*Adaptive Weights Calculation*).

The idea of procedure has been derived from well known single perceptron learning concept presented in [24], where weight values are corrected in each iteration. The correction depends on estimator of probability errors of single classifiers and estimator of classifier committee based on weights counted in this step. AWC increases the weights of classifiers which accuracies are higher than accuracy of common classifier. The pseudocode of procedure is shown below.

Procedure AWC

Input: $\Psi_1, \Psi_2, \dots, \Psi_k$ - set of k classifiers for the same decision problem, Learning set LS ,
 T number of iterations

1.. For each classifier Ψ_i

a) compute weights of classifier $w_i(0) = \frac{1}{k}$

b) estimate accuracy probability of $\Psi^{(i)} - \hat{P}_{a,i}$

2. For $t:=1$ do T

a) estimate accuracy probability of $\bar{\Psi} - \hat{P}_a(t)$

b) sum_of_weights:=0

c) for each classifiers Ψ_i

$$w_i(t) = w_i(t-1) + \left(\frac{(\hat{P}_a(t) - \hat{P}_{a,i})}{(t + \gamma)} \right)$$

if $w_i(k) > 0$

then sum_of_weights := sum_of_weights + $w_i(k)$

else $w_i(t) := 0$;

d) if sum_of_weights > 0
 then

i) for each classifier Ψ_i

$$w_i(t) := \frac{w_i(t)}{\text{sum_of_weights}}$$

else

i) for each classifier Ψ_i

$$w_i(t) = w_i(t-1)$$

ii) $t := T$

The γ is integer, constant value fixed arbitrary (usually given by expert or obtained via experiments) which is connected with the speed of the weights learning process. The weight corrections in each step is reverse proportional to the γ value.

4 Experimental Investigation

The aim of the experiment is to compare the errors of the weighted combined classifiers with the quality of single classifiers and another concept of classifier fusion [9, 10]. The following classifiers and fusion methods were chosen:

1. Majority voting (mv),
2. Weighted voting which based on the weights reverse proportional to estimation of probability errors of single classifiers presented in equation (3) (pv),
3. Adaptive weighted voting classifier which uses weights obtained via AWC procedure described in section 3 (pv),
4. Single classifier based on the learning sequence generated according to the chosen unperturbed probability distribution (p1),
5. Single classifier learned on the set generated according to the chosen probability distribution, perturbed 10% of elements of the uniform distribution (p2),
6. Single classifier learned on the set generated according to the chosen probability distribution, perturbed 20% of elements of the uniform distribution (p3),
7. Single classifier learned on the set generated according to the chosen probability distribution, perturbed 30% of elements of the uniform distribution (p4),
8. Single classifier learned on the set generated according to the chosen probability distribution, perturbed 40% of elements of the uniform distribution (p5).

The conditions of experiments were as follow:

1. All experiments were carried out in Matlab environment using the PRtools toolbox [11] and own software.
2. Errors of the classifiers were estimated using the ten fold cross validation method.
3. In all experiments two-class recognition task was considered with the *prior* probability values equal 0.5 for each classes.
4. The conditional density functions were generated according the Normal or Banana distribution. The examples of learning sets are depicted in Fig.2.
5. In each experiment the size of learning set were fixed on 300.

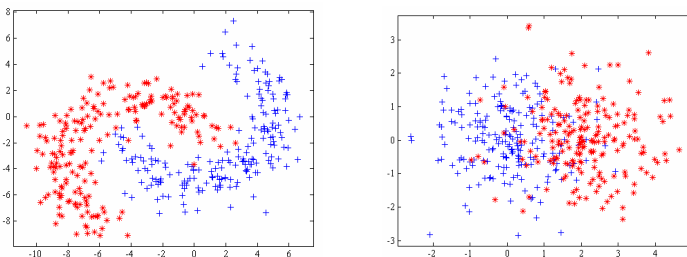


Fig. 2. The examples of learning sets generated according Banana and Normal distributions

We use decision tree obtained via C4.5 algorithm [14] and two classifiers based on the estimation of the probability density [12,13] via k_n -Nearest Neighbor and the Parzen procedure. Selected results of experiments are presented in Table 1.

Table 1. Classification error [%] of simple classifiers and committees of classifiers

Learning method	Distribution	γ	Classifier							
			p1	p2	p3	p4	p5	Av	mv	Pv
k_n -NN	Banana	2	2.4	2.6	3.8	9.2	26.6	1.9	2.0	2.1
Parzen	Banana	2	2.4	2.9	3.7	7.8	25.6	2.3	2.7	2.5
C4.5	Banana	10	6.1	19.4	28.3	33.6	39.5	7.2	12.7	13.3
k_n -NN	Highleyman's	2	7.6	9.8	13.1	22.6	39.0	7.7	10.0	10.2
Parzen	Highleyman's	3	6.8	11.2	15.2	23.7	42.1	8.9	13.5	12.6
C4.5	Highleyman's	10	8.0	20.1	29.4	39.4	51.9	9.8	17.5	16.0
k_n -NN	Normal	1	17.9	17.4	18.9	22.1	33.9	17.7	17.0	17.4
Parzen	Normal	1	16.9	17.0	18.5	24.6	35.8	16.8	17.3	17.1
C4.5	Normal	10	21.4	29.7	24.8	38.9	43.9	21.5	24.0	23.2

4.1 Experimental Results Evaluation

Firstly, one has to note that we are aware of the fact that the scope of computer experiments were limited. Therefore, making general conclusions based on them is very risky. In our opinion mentioned below statements should not be generalized at this point, but they should be confirmed by other experiments in much broader scope. In the case of the presented experiment:

1. combined algorithms gave results slightly worse than the best single classifier. It was also observed that classifiers, for which decision making used weighted answer of single classifiers (AV, PV), recognized objects definitely better than the rest of classifiers except for the best single one;
2. AWC algorithm (AV) improved the quality of weighted voting (PV) which based on the weights reverse proportional to estimation of probability errors of single classifiers presented in equation (3);
3. in adaptive weights calculation procedure the values of each classifiers weights were probably closer their optimal value than weights presented in equation (3);
4. the error of the best simple classifier is close to the qualities of combined classifiers but sometimes slightly worse sometimes slightly better than proposed concept. We must have respect to fact that for proposed experiment this classifier based on the learning set without noises.

We were presenting only the piece of experimental results. During the experiments we observed that γ factor had been playing the important role in the learning process. For presentation we chose only one value of γ with the best quality of classification.

We stated that the weights values established in the first 5-10 iterations. For some experiments we observed that after about 10 iterations classification error of combined classifier was increasing. In some cases the weights of the best individual

classifier exceeded 0.5 what caused that quality of decision of committee of classifiers was the same as for the best classifier.

During learning we noticed that when the differences between the qualities of individual classifiers had been quite big (how it took place e.g. in the case of the C4.5 algorithm), the AWD procedure assigned the weight bigger than 0.5 to the best classifier. It means that decision of the group classifiers is *de facto* decision of the best classifier. It is not undesirable behavior because lets consider the following team of five independent classifiers

$$\Psi^{(1)}, \Psi^{(2)}, \Psi^{(3)}, \Psi^{(4)}, \Psi^{(5)}$$

with accuracies

$$P_{a,1} = 0.55, P_{a,2} = 0.60, P_{a,3} = 0.65, P_{a,4} = 0.70, P_{a,5} = 0.80.$$

The probability of error of the committee of classifiers (voting according majority rule) is $P_e^{(5)} = 0.20086$. As we see it would be better if we remove the worst classifiers and reduce the ensemble to the individual and the most accurate classifier $\Psi^{(5)}$ [5].

5 Final Remarks

The original method of weights calculation for ensemble of classifiers was presented in this paper. This study presented the experimental quality evaluation of the combined classifiers recognition also.

Obtained results seem to confirm the sense of using combing methods and they are similar as published in [15-18]. Unfortunately, as it was stated, it is not possible to determine their value in the analytical way. One should although hope that in case of application of above mentioned methods for real decision tasks, we have judgment of the expert, who can formulate the heuristic function of weights selection or be calculated by heuristic procedure like methods proposed in this paper. This function then can be verified and improved by the computer experiment. We have to stress that proposed method suffers from overfitting. Therefore we have to be very careful when we fixed the number of procedure iterations.

Acknowledgments. This work is supported by The Polish State Committee for Scientific Research under the grant which is realizing in years 2006-2009.

References

1. Xu L., Krzyżak A., Suen Ch.Y., Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, *IEEE Trans. on SMC*, no. 3, 1992, pp.418-435.
2. Chow C.K., Statistical independence and threshold functions, *IEEE Trans. on Electronic Computers*, EC-16, 1965, pp.66-68.
3. Jain A.K., Duin P.W., Mao J., Statistical Pattern Recognition: A Review, *IEEE Trans. on PAMI*, vol 22., No. 1, 2000, pp. 4-37.

4. Hansen L.K., Salamon P., Neural Networks Ensembles, *IEEE Trans. on PAMI*, vol. 12, no. 10, 1990, pp. 993-1001.
5. Kuncheva L.I., *Combining pattern classifiers: Methods and algorithms*, Wiley, 2004.
6. Hashem S., Optimal linear combinations of neural networks, *Neural Networks*, 10(4), 1997, pp.599-614.
7. Tumer K., Ghosh J., Linear and Order Statistics Combiners for Pattern Classification [in:] Sharkley A.J.C. [ed.] *Combining Artificial Neural Networks*, Springer, 1999, pp. 127-155.
8. Fumera G., Roli F., A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems, *IEEE Trans.on PAMI*, vol. 27, no 6, 2005, pp.942-956.
9. Kittler J., Alkoot F.M., Sum versus Vote Fusion in Multiple Classifier Systems, *IEEE Trans.on PAMI*, vol. 25, no. 1, 2003, pp. 110-115.
10. Kuncheva L.I., Whitaker C.J., Shipp C.A., and Duin R.P.W., Limits on the Majority Vote Accuracy in Classier Fusion, *Pattern Analysis and Applications*, 6, 2003, pp. 22-31.
11. Duin R.P.W., Juszczak P., Paclik P., Pekalska E., de Ridder D., Tax D.M.J. *PRTTools4, A Matlab Toolbox for Pattern Recognition*, Delft University of Technology, 2004
12. Devijver P. A., Kittler J., *Pattern Recognition: A Statistical Approach*, Prentice Hall, London, 1982
13. Duda R.O., Hart P.E., Stork D.G., *Pattern Classification*, John Wiley and Sons, 2001.
14. Quinlan J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
15. Duch W., Grudziński K., Ensembles of Similarity-Based Model, *Advances in Soft Computing*, Springer Verlag 2001, pp.7585.
16. Duin R.P.W., Tax, D.M.J., Experiments with Classifier Combining Rules, *Lecture Notes in Computer Science*, No. 1857, 2000, pp.16-29.
17. Kuncheva L.I., Bezdek J.C., Duin R.P.W., Decision templates for multiple classifier fusion: an experimental comparison, *Pattern Recognition*, 34, 2001, pp. 299-314.
18. Van Erp M., Vuurpijl L.G., Schomaker L.R.B., An overview and comparison of voting methods for pattern recognition, *Proc. of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR.8)*, Niagara-on-the-Lake, Canada, pp. 195-200.
19. Kuncheva L.I. Jain L.C., Designing classifier fusion systems by genetic algorithms, *IEEE Transactions on Evolutionary Computation*, Vol. 4, 2000, pp.327-336.
20. Inoue H., Narihisa H., Optimizing a Multiple Classifier Systems, *Lecture Notes in Computer Science*, Vol. 2417, 2002, pp.285-294.
21. Ting K.M., Witten I.H., Issues in Stacked Generalization, *Journal of Artificial Intelligence Research*, No. 10, 1999, pp. 271-289.
22. Wolpert D.H., Stacked generalization, *Neural Networks*, No. 5(2), 1992, pp. 241-260.
23. Duin R. P.W., The Combining Classifier: to Train or Not to Train?, *Proc. of the ICPR2002*, Quebec City, 2002.
24. Nilsson, N. J. *Learning machines*, New York: McGraw-Hill, 1965.

Fusion of Visualization Induced SOM

Bruno Baruque and Emilio Corchado

Department of Civil Engineering, University of Burgos, Spain
{bbaruque, escorchado}@ubu.es

Abstract. In this study ensemble techniques have been applied in the frame of topology preserving mappings with visualization purposes. A novel extension of the ViSOM (Visualization Induced SOM) is obtained by the use of the ensemble meta-algorithm and a later fusion process. This main fusion algorithm has two different variants, considering two different criteria for the similarity of nodes. These criteria are Euclidean distance and similarity on Voronoi polygons. The goal of this upgrade is to improve the quality and robustness of the single model. Some experiments performed over different datasets applying the two variants of the fusion and other simpler models are included for comparison purposes.

1 Introduction

Nowadays, with the incredibly high amounts of data that industrial and business operations processes as well as research studies generate; the main problem is not the extraction of the data itself, but the extraction of useful information from within those huge databases. Among the great variety of analysis tools for multivariate data analysis and pattern recognition that are being developed, we can find the artificial neural networks.

Topology Preserving Maps [1], which include the Self-Organizing Maps (SOM) [2] and the Visualization Induced SOM [3], were originally created as a visualization tool; enabling the representation of high-dimensional datasets onto two-dimensional maps and facilitating the human expert the interpretation of data.

A general way of boosting the classification capabilities of classic classifiers (such as decision trees) is the construction of ensembles of classifiers [4], [5]. Following the idea of a ‘committee of experts’, the ensemble technique consists of training several identical classifiers on slightly different datasets in order to constitute a ‘committee’ to classify new instances of data. The topology preserving mapping algorithms can be considered in a certain way generic classifiers due to their inherent pattern storing and recognition capabilities [6]. Each of the units or neurons of these maps can be considered a classification unit specialized in the recognition of several similar patterns, as it reacts with higher values than the other to the presence of those patterns and not to the presence of others.

The guiding idea of this work is using the ensemble meta-algorithm approach on several topology preserving models to improve their stability and visualization performance.

2 The SOM and the ViSOM

Both models belong to a family of techniques with a common target: to produce a low dimensional representation of the training samples while preserving the topological properties of the input space. The best known technique is the Self-Organizing Map (SOM) algorithm [2]. It is based on a type of unsupervised learning called competitive learning; an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, sets of samples in a specific domain of the input space [1].

One interesting extension of this algorithm is the Visualization Induced SOM (ViSOM) [3], [7] proposed to directly preserve the local distance information on the map, along with the topology. The ViSOM constrains the lateral contraction forces between neurons and hence regularises the interneuron distances so that distances between neurons in the data space are in proportion to those in the input space.

The difference between the SOM and the ViSOM hence lies in the update of the weights of the neighbours of the winner neuron as can be seen from Eq. (1) and Eq. (2).

Update of neighbourhood neurons in SOM:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v, k, t)(x(t) - w_v(t)) \quad (1)$$

Update of neighbourhood neurons in ViSOM:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v, k, t) \left([x(t) - w_v(t)] + [w_v(t) - w_k(t)] \left(\frac{d_{vk}}{\Delta_{vk}\lambda} - 1 \right) \right) \quad (2)$$

where w_v is the winning neuron, α the learning rate of the algorithm, $\eta(v, k, t)$ is the neighbourhood function where v represents the position of the winning neuron in the lattice and k the positions of the neurons in the neighbourhood of this one, x is the input to the network and λ is a “resolution” parameter, d_{vk} and Δ_{vk} are the distances between the neurons in the data space and in the map space respectively.

To evaluate the quality of the adaptation of the map to the dataset represented a very widely known measure is the mean square quantization error. This can be represented as:

$$MSQE = \frac{1}{|D|} \sum_{x_i \in D} \|x_i - m_{c(x_i)}\|^2 \quad (3)$$

where $|D|$ is the number of data in the dataset D , and $m_{c(x_i)}$ is the best matching unit of the map to the data sample x_i of the dataset [8].

Another way of determining the quality of a map, when datasets include a class attribute for each of its entries, is the classification accuracy of the network. A high accuracy in the classification rate implies that the neurons of the network are reacting in a more consistent way to the classes of the samples that are presented, As a consequence, the map should represent the data distribution more accurately [9].

3 Mapping Fusion

3.1 The Ensemble Meta-algorithm

The ultimate goal for designing pattern recognition systems is to achieve the best classification performance possible for the task at hand. It has been observed in several studies that although one of the classifiers in an ensemble would yield the best performance, the sets of patterns misclassified by the different classifiers would not necessarily overlap. This suggests that different classifier designs potentially offer complementary information about the patterns to be classified and could be harnessed to improve the performance of the selected classifier [4]. As explained before, topological preserving maps can be considered as classifiers. Their learning algorithm specifies that their composing units (or neurons) specialize during the algorithm iterations in recognizing a certain type of patterns. This process is clearly a kind of classification, as that process can be used to group patterns by their similarity. The main problem of competitive learning based networks is that are inherently unstable due to the nature of their learning algorithm. Considering the topology preserving maps as a kind of classifiers; the main idea of this work is that the effect of this instability may, however, be minimized by the use of ensembles [10].

The algorithms to combine classifiers can be divided into two broad classes. The simpler variety of algorithms merely combines, by averaging in some way, the results of each of the composing classifiers of the ensemble yields into a final result. More complex types of algorithms try to combine not only the results but the whole set of classifiers in order to construct a single better one that should outperform its individual components. In the case of this paper this second perspective, the concept of a single “summary” or “synthesis” of the patterns stored within the whole ensemble, is the one followed to improve the model performance. This is because obtaining a unique map is the easiest way to obtain a clear visualization of data for human inspection, rather than representing several similar maps. Our main objective is to obtain a unique map that may be seen to represent all of the features of the maps in the ensemble.

3.2 Proposed Combinations

A number of ensemble techniques are applied to the SOM [11], ViSOM and other topological mapping networks, mainly for classification purposes.

In the context of the visualization, some adaptations are necessary to build a meaningful combination of the maps they represent. In this work a main algorithm for mapping fusion with two different variants is used for the first time in combination with the ViSOM. The objective is the comparison of the two in order to obtain conclusions that can be used in further studies to generate a more accurate model.

The procedure is the same for the training of the networks that compose the ensembles. All are trained using typical cross-validation, with the dataset divided into several folders, leaving one of them out to test the classification accuracy. To train the network ensemble the meta-algorithms called bagging [12] is used. It consists on obtaining n subsets of the training dataset through re-sampling with replacement and

trains individual classifiers on such re-sampled subsets. This permits to generate n different trained networks which are combined into a final network that is expected to outperform each of them individually. The combination of maps is done once all the networks composing the ensemble have finished their training.

This combination is done in a neuron by neuron basis. That is, neurons that are considered ‘near enough’ one to the other are fused to obtain a neuron in the final fused network. This is done by calculating the centroid of the weights of the neurons to fuse:

$$w_{neuAvg} = 1/n \cdot \sum_{i=1}^n w(i) \tag{4}$$

That process is repeated until all neurons in all trained networks are fused into a unique final one. The criteria to determine which neurons are ‘near enough’ to be fused is what determines the two variants of the main algorithm.

Voronoi Polygons: Each neuron in a Self-Organizing Map can be associated with a portion of the input data space called the Voronoi polygon. That portion of the input multi-dimensional space is the portion that contains data for which that precise neuron is the Best Matching Unit (BMU) of the whole network [1]. It is therefore a logical conclusion to consider that neurons that are related to similar Voronoi polygons can be considered similar between them, as they should be situated relatively close in the input data space.

To calculate the dissimilarity between two neurons, a record of which data entries activated each neuron as the BMU can be kept. This can be done easily associating a binary vector to the neuron which length is the size of the dataset and contains ones in the positions where the neuron was the BMU for that sample and zeros in the rest of positions. The dissimilarity (i.e. the distance) between neurons can therefore be calculated as:

$$d(v_r, v_q) = \frac{\sum_{l=1}^n XOR(v_r(l), v_q(l))}{\sum_{j=1}^n OR(v_r(j), v_q(j))} \tag{5}$$

being r and q the neurons to determine their dissimilarity and v_r and v_q the binary vectors relating each of the neurons with the samples recognized by it. A much more detailed explanation can be found in [13].

The main problem with this proximity criterion is that it depends on the recognition of data by the network, rather than on the network definition itself. This means that a neuron that does not react as the BMU for any data could be considered similar to another neuron with the same characteristic, although they can be relatively far from each other in the input data space. To avoid this, all neurons with a reacting rate lower than a threshold are removed before calculating the similarities between them. This implies that the neighbouring properties of the network are no longer considered. To keep a notion of neighbouring between the neurons of the fused network the similarity criteria must be used again. Neurons with dissimilarity less than a threshold will be considered as neighbours in the fused network.

Algorithm 1. Fusion based on similarity in Voronoi Polygons

-
- 1: Being $nNet$ the number of networks to be fused
 fus the resultant fused network
 θ_u , θ_f and θ_c the usage, fusion and connection thresholds respectively
 - 2: **for** $i=1:nNet$
 - 3: remove from the network the neurons that have a recognition rate lower than a usage threshold ($\sum_i v_r(i) < \theta_u$)
 - 4: add all the rest of the neurons of network(i) to the set of all nodes of ensemble
 - 5: **end**
 - 6: calculate the dissimilarity between all neurons contained in the set obtained in 3-6 using Eq. 5
 - 7: group in different sub-sets the nodes that satisfy that the dissimilarity between all of them is lower than the dissimilarity threshold and the distance between each of them and the rest of nodes in other sub-set is higher than that threshold.

$$\begin{cases} ds(v(n_r), v(n_q)) < \theta_f & \text{for all } n_r, n_q \in S_k \\ ds(v(n_r), v(n_q)) \geq \theta_f & \text{for all } n_r \in S_k, n_q \in S_l, k \neq l \end{cases}$$

The result will be a set of sub-sets (S).
 - 8: ‘fuse’ all the nodes in each sub-set to form a node of the final fused network by calculating the centroid of the nodes in each sub-set (see Eq. 4). The fused network will have as many nodes as sub-sets in are contained in S .
 - 9: create the final network (fus) including in it all the fused nodes
 - 10: create connections between fused nodes in the fused network (fus) to represent neuron neighbourhood. Connections will be established if the distance between fused nodes is lower than the connection threshold, considering this distance as:

$$\min_{n_r \in N_r, n_q \in N_r} ds(v(n_r), v(n_q)) < \theta_c$$

Euclidean Distance: This method involves comparing the networks neuron by neuron in the input space. This implies that all the networks in the ensemble must have the same size. First, it searches for the neurons that are closer in the input space

Algorithm 2. Fusion based on Euclidean Distance

-
- 0: Train several networks by using the bagging (re-sampling with replacement) meta-algorithm
 - 1: Being $nNet$ the number of networks to be fused
 $nNeur$ the number of neurons composing each network
 fus the resultant fused network
 - 2: Initialize fus with the neuron weights of the first network
 - 3: **for** $i=2:nNet$
 - 4: **for** $j=1:nNeur$
 - 5: $neuFus$: neuron (j) of the fus network
 - 6: calculate Euclidean Distance (ED) between $neuFus$ and ALL neurons of network(i)
 - 7: $neuNet$: neuron with the minimum ED
 - 8: calculate $neuAvg$: neuron whose weights are the average of the weights of $neuFus$ and $neuNet$ i.e. the centroid of both neurons’ weights (see Eq. 4).
 - 9: remove $neuNet$ from the set of neurons of the network
 - 10: replace $neuFus$ by $neuAvg$ in the fus network (in position j of the network)
 - 11: **end for**
 - 12: **end for**
-

(selecting only one neuron in each network of the ensemble) then it “fuses” them to obtain the final neuron in the “fused” map. This process is repeated until all the neurons have been fused. To deal with the high computational complexity of the algorithm, it can be implemented using dynamic programming. A more detailed description of this procedure can be found in [14].

The difference with the previous criteria is that, in this case, a pair wise match of the neurons of each network is always possible, so the final fused network has the same size as the single composing ones. This implies that a certain neighbouring structure can be kept and reconstructed in the fused network.

4 Experiments and Results

To test the characteristics and capabilities of the fusion of ViSOM and compare both of its variants several real datasets have been employed. Data were extracted from the UCI repository [15] and include the iris, the Wisconsin breast cancer and the wine datasets.

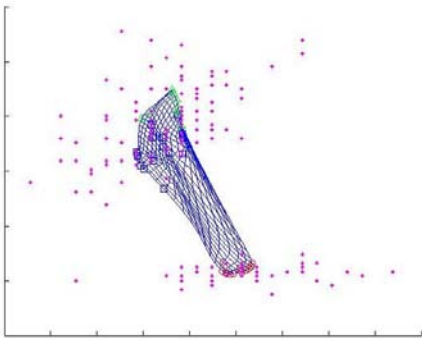


Fig. 1. A single ViSOM network represented over the iris dataset

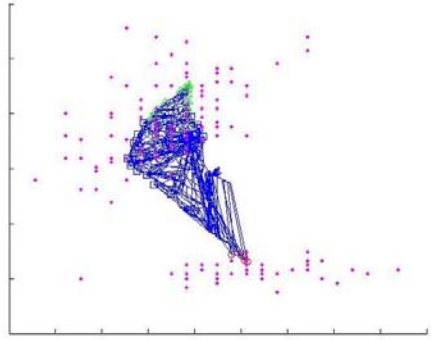


Fig. 2. The fusion of 5 ViSOM networks using the Euclidean Distance criterion

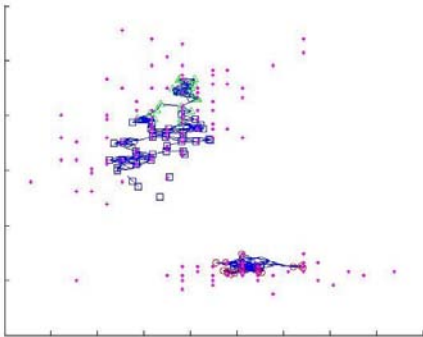


Fig. 3. The fusion of 5 ViSOM networks using the Voronoi polygon similarity criterion

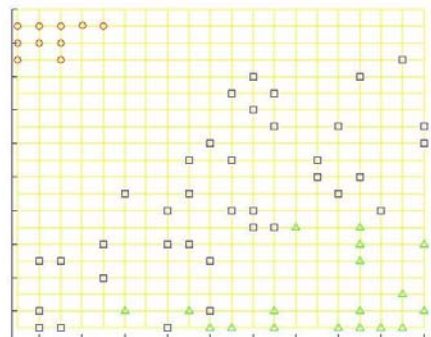


Fig. 4. The 2D map representation of the fused network represented on Fig. 2

The first thing that is important noting is a structural difference of the two variants of fusion. As explained before, the Euclidean distance variant enables the pair wise fusion of nodes of networks, so topology preservation is still valid in the fused network. This allows obtaining 2D maps such as the one showed in Fig. 4., which can be also easily obtained from single maps. This is impossible to do with the Voronoi similarity, as some neurons are not related to others and a position in the map in relation with the rest can not be determined.

Regarding the performance results, several fusions were calculated varying the amount of networks for the fusion from 1 to 20 (see Fig.5 to Fig. 8). Complementary results can be observed. For the Mean square error, this decreases when using more networks for the fusion if the Voronoi polygons similarity criterion is used, but will increase when using the Euclidean distance (Fig 5 and Fig 6). The same phenomenon can be appreciated when talking about the classification accuracy of the networks.

This suggests that while as the number of networks increases the Euclidean distance fusion variant tends to make network nodes to move away from the data in the input space. It also tends to avoid overfitting, while the Voronoi similarity variant has the opposite effect.

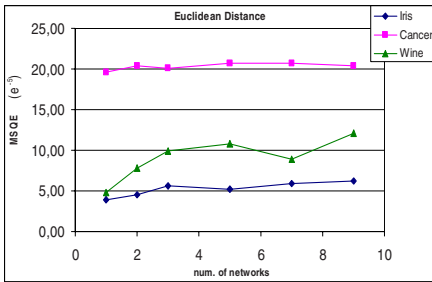


Fig. 5. Mean Square Quantization Error in the fusion using Euclidean distance for the three datasets when varying the number of networks to fuse

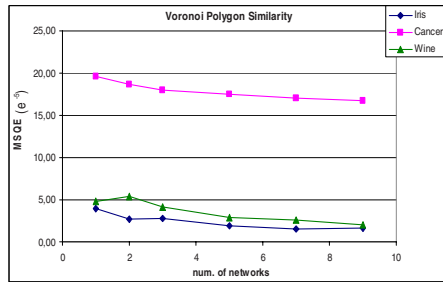


Fig. 6. Mean Square Quantization Error in the fusion using Voronoi polygons similarity for the three datasets when varying the number of networks to fuse

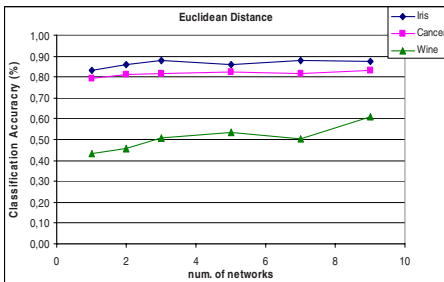


Fig. 7. Classification Accuracy (in %) in the fusion using Euclidean distance for the three datasets when varying the number of networks to fuse

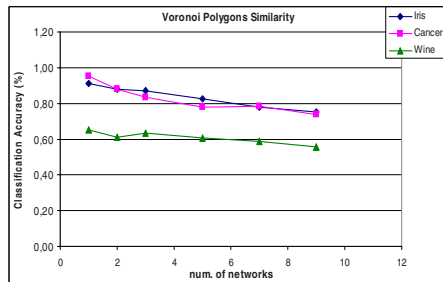


Fig. 8. Classification Accuracy (in %) in the fusion using Voronoi polygons similarity for the three datasets when varying the number of networks to fuse

5 Conclusions and Future Work

A study of a technique of improving the robustness and accuracy of a topology preserving model such as the ViSOM is presented in this work. This is achieved by the use of the ensemble theory and a fusion process. Two different variants of the algorithm are considered and studied, each one obtaining rather complementary results regarding their performance and the visual inspection suitability of their structure. Future work should include a further study of the capabilities of each one of the variants. This is considered as a previous step to the proposition of a fusion algorithm developed to bring together the best characteristics of both variants. Also application and comparison of this new fusion algorithm with some other topology preserving ensemble models will be performed.

Acknowledgments. This research has been supported by the MCyT project TIN2004-07033 and the project BU008B05 of the JCyL.

References

1. Kohonen, T., et al., *A principle of neural associative memory*. Neuroscience, 1977. 2(6): p. 1065-1076.
2. Kohonen, T., *The self-organizing map*. Neurocomputing, 1998. 21(1-3): p. 1-6.
3. Yin, H., *ViSOM - a novel method for multivariate data projection and structure visualization*. Neural Networks, IEEE Transactions on, 2002. 13(1): p. 237-243.
4. Kuncheva, L.I., *Combining Pattern Classifiers: Methods and Algorithms*. 2004: Wiley-Interscience.
5. Ron, M. and R. Gunnar, *An Introduction to Boosting and Leveraging*. Advanced Lectures on Machine Learning: Machine Learning Summer School 2002, Canberra, Australia, February 11-22, 2002. Revised Lectures. 2003. 118-183.
6. Kaski, S., *Data exploration using self-organizing maps*, in *Department of Computer Science and Engineering*. 1997, Helsinki University of Technology: Espoo, Finland.
7. Yin, H., *Data visualisation and manifold mapping using the ViSOM*. Neural Networks, 2002. 15(8-9): p. 1005-1016.
8. Kaski, S. and K. Lagus, *Comparing Self-Organizing Maps*, in *International Conference on Artificial Neural Networks (ICANN'96)*. 1996 Springer-Verlag.
9. Kraaijveld, M.A., J. Mao, and A.K. Jain, *A nonlinear projection method based on Kohonen's topology preserving maps*. Neural Networks, IEEE Transactions on, 1995. 6(3): p. 548-559.
10. Ruta, D. and B. Gabrys, *A theoretical analysis of the limits of Majority Voting errors for Multiple Classifier Systems*. Pattern Analysis and Applications, 2002. 5(4): p. 333-350.
11. Petrakieva, L. and C. Fyfe, *Bagging and Bumping Self Organising Maps*. Computing and Information Systems, 2003.
12. Breiman, L., *Bagging Predictors*. Machine Learning, 1996. 24(2): p. 123-140.
13. Saavedra, C., et al. *Fusion of Self Organizing Maps*. in *9th International Work-Conference on Artificial Neural Networks, IWANN 2007*. 2007. San Sebastián, Spain: Springer.
14. Georgakis, A., H. Li, and M. Gordan, *An ensemble of SOM networks for document organization and retrieval*. Int. Conf. on Adaptive Knowledge Representation and Reasoning (AKRR'05), 2005: p. 6.
15. Newman, D.J., et al. *UCI Repository of machine learning databases*. 1998; Available from: <http://www.ics.uci.edu/~mllearn/MLRepository.html>.

Open Intelligent Robot Controller Based on Field-Bus and RTOS

Zonghai Chen and Haibo Wang

Department of Automation, University of Science and Technology of China,
Hefei, P.R. China
chenzh@ustc.edu.cn

Abstract. Robot controller is one of the key facts affecting performance of robot. To meet the requirements of theoretical and application research, a mobile robot controller based on CAN Bus and Real-time Operating System (RTOS) is presented. In hardware aspect, a distributed architecture is achieved using CAN Bus and Bluetooth technology. Each module of the controller is integrated into a multi-agent based architecture, and is implemented in RTOS in software aspect. A comprehensive illustration of ATU-II mobile robot platform based on the proposed controller is presented and two experiments of ATU-II verify its performance.

1 Introduction

Development of open autonomous mobile robot system is an active issue in the area of artificial intelligence ^[1]. To meet the demands of real-life application and theoretical research, a mobile robot system should not only be an experimental platform, but also a prototype robot with the capability of being further developed into robot product for specific usage. Since the robot controller is the key element in determining the performance of a mobile robot system, it should have the following properties ^[2]:

1) Openness

Openness means the robot system support adding new software and hardware modules to the system. This property is very important because mobile robot system tends to evolve in terms of both software and hardware and with such property upgrade can be done easily. Although there is not a standard on the openness of a robot controller yet, but there is a widely accepted basic principle that an open robot controller should be in possession of extensibility, scalability and portability.

2) Real-time property

A mobile robot is a complex system, in which there are complicated scheduling relationships among tasks inside a single module as well as among different modules. In order to function properly it must be guaranteed that tasks inside a single module and different modules be executed within time limit.

3) Stability

The robot controller should be able to work continuously for a long period of time and its overall performance should be guaranteed even when there are faults in some of its modules.

Many researchers have tried to develop an open mobile robot controller since 1990s. Stanford Research Institute international AI center developed Saphira [3]; Carnegie Mellon University (CMU) Robot Institute developed Teambots [4]; the Royal Institute of Technology designed BERRA [5]. Although the mentioned mobile robots perform soundly in the aspects of stability and real-time computing, it's still not convenient for the user to extend and upgrade the functions of the robot due to the limits of hardware and software structures. That is to say, the openness of the controller is not well accomplished.

Aiming at achieving above properties, a mobile robot controller based on CAN Bus and Real-Time Operating System (RTOS) is presented in this paper. In hardware aspect, a distributed architecture is achieved using CAN Bus and Bluetooth technology. Each module of the controller is then integrated into a multi-agent based architecture and implemented in RTOS in software aspect. A mobile robot platform named "ATU-II" using this kind of controller and two experiments of ATU-II are introduced in order to verify the performance of this robot controller.

In section 2, the architecture of mobile robot controller including hardware and software structure is described in detail. In section 3, the mobile robot platform ATU-II is introduced as well as two experiments. Conclusion is given in section 4.

2 Architecture of Controller

Openness is the most important property for an open mobile robot controller. Willian E F had proposed the main ideas of an open robot controller as following [1]:

- 1) Use a development of system based on a non-proprietary computer platform;
- 2) Use a standard operating system and a standard control language;
- 3) Based the hardware on a standard bus architecture that can interface with a variety of peripherals and sensor devices
- 4) Utilize networking strategies that allow workcell controllers to share database, and to be operated from remote locations

According to the above principles, taking into account the stability and real-time property, the architecture of the mobile robot controller is designed, as described in detail in the following section.

2.1 Hardware Structure of Controller

The hardware structure of controller is shown in Fig. 1 and is divided into two parts: a remote host and an onboard controller.

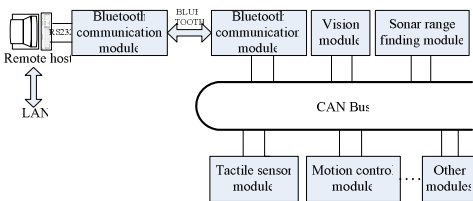


Fig. 1. Hardware architecture of robot controller

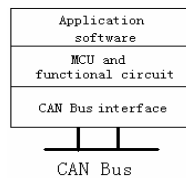


Fig. 2. Architecture of single module

The remote host is composed of a high performance PC where time-consuming and memory-demanding computational tasks are executed. It also provides human-machine interface (HMI). The host is connected to internet to provide internet access to the controller, is also connected to a Bluetooth communication module of onboard controller via a RS-232 interface.

Onboard controller is mainly classified into perception subsystem and execution subsystem, which can be further classified into more inter-independent hardware modules. Each module is connected as a network node into CAN Bus on which data communications are done. The architecture of a functional module is shown in Fig. 2. It consists of three parts: a processor with its supporting circuit, a module function circuit (such as sensor or actuator circuit) and a CAN interface circuit. The module function circuit under the control of the processor implements most of the functions of a module. In our implementation, notebook PC processor is selected in vision module and Intel 16-bit single chip 80c196KC is selected in other modules. The CAN interface circuit controlled by processor is in charge of communications between each module and the bus. And wireless communications between onboard controller and remote host are implemented by a special node of CAN Bus—the Bluetooth communication module.

The above hardware structure has the following advantages:

1) With above distributed structure, distributed computation of each module can be achieved. Thus computational burden of a single processor is greatly lowered and efficiency is greatly increased comparing to traditional centralized structure.

2) CAN Bus is implemented and its advantages can be naturally possessed by the controller. CAN Bus, which is one of the most popular field buses, is a highly reliable serial communication network protocol with the ability to resist strong electro-magnetic inferences, and with effective support to distributed and real-time control. Node of CAN Bus can be connected into and removed from the bus conveniently. Subsequently, each function module can be easily connected into and removed from onboard controller, which provides highly extensibility and scalability in hardware respect. Moreover, CAN Bus has other advantages compared to other kinds of buses: multiple communication methods, higher data transmission rate, easier scheduling and larger number of nodes can be contained. Especially, a node with severe errors can be removed from the bus automatically without affecting other nodes. Thus stability and other communication properties of the controller are guaranteed.

3) By dividing the controller into remote host and onboard controller, Internet access is reserved and a weighted and big-sized onboard PC or IPC is no longer needed. As a result, the weight and size of the robot are greatly decreased and much energy consumption is saved.

2.2 Software Structure of Controller

In the proposed mobile robot controller, a hybrid software architecture based on multi-agent^[6] is used integrating all modules shown in Fig.3.

Agents in this software architecture can be classified into three categories:

(1) Hardware interface agents, including sensor agents and actuator agents. Sensor agents perceive the status of the environment and the robot, and then process and

transmit the perceived information to blackboard system and reactive agent. Actuator agents receive commands from other agents and execute them on physical actuators.

(2) Deliberative agents, by which deliberative planning of mobile robot, such as navigation, map-building and global path planning, is executed. They bear a hierarchical structure based on task decomposition and can be implemented variably according to different task and different degree of abstraction.

(3) Reactive agents, including obstacle avoidance agents and sharp stopping agents, etc. Reactive agents map information from sensor agents to commands, and then transmit the commands to actuator agents. Besides input/output elements, a reactive agent has a reactive behavior rule base and a behavior combiner.

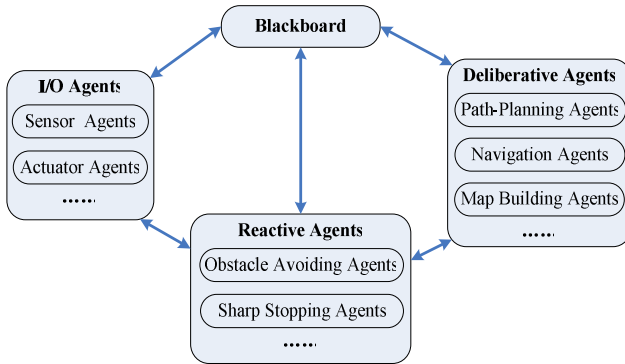


Fig. 3. Framework of multi-agent based hybrid architecture

Furthermore, to share relative information and knowledge, a blackboard system consisting of a memory element and a reasoning element is also used to organize information and make decisions.

In the above multi-agent software architecture, hardware interface agents correspond to perception subsystems and execution subsystems of hardware architecture, while the reactive agents, the deliberative agents and the blackboard system are purely implemented by software.

Among different agents as well as among different tasks inside a single module, there exist complicated scheduling relationships. In order for the mobile robot to function properly, these scheduling relationships must be handled to guarantee all the tasks of different agents be executed within time limit in certain orders. Operating system such as MS Windows and Linux cannot meet the requirements. Thus RTOS is a must for software system of a mobile robot system.

Essentially, a RTOS is a high efficient, real-time, multi-task kernel. It is capable of reasonably allocating CPU time among different tasks according to their priorities to achieve real-time execution. A good RTOS is optimized for dozens of series of embedded processors and provides similar APIs to achieve hardware abstraction in software development. It is estimated that for C programs based on RTOS only 1-5% codes need modification in transplanting between different processors.

Hardware interface agents, which correspond to perception subsystem and execution subsystem of hardware architecture and needs less computation, are implemented on

80c196kc processors in onboard controller. Considering the limit of memory and processing speed, a mini-kernel operating system $\mu\text{C}/\text{OS-II}$, which has advantages of small size, simple structure and high portability^[7], is adopted in such modules.

Meanwhile, reactive agents, deliberative agents and the blackboard system, which involves making high-level decision and require higher processing speed and larger memory, are implemented on stronger processor in remote host. Remote host software is developed using WATCOM C in QNX RTOS, which provides multi-process programming and communication using shared memory and thus has strong abilities in signal processing and interaction. In QNX, a process is awakened by interrupts or events. When a certain event or interrupt occurs, the waiting process is awakened and handles the event according to its category. After that, the process restores waiting status again. This property is valuable in controlling orders of different tasks and data transmissions so as to meet the requirements of scheduling relationships. Shared memory, which is a public data buffer, is used as a medium of different processes. Each process can communicate with shared memory individually and is under the control of process scheduler. In this way, functions of each process can be integrated. In implementation, a process is created for each reactive agent, deliberative agent and the blackboard system. And priority and scheduling relationship is assigned to each process according to its relationship with other agents.

The above multi-agent hybrid software architecture has the following advantages:

1) Extension to controller software can be done easily only by modifications to certain agents without modifying the overall structure. For example, when a new function is extended to the robot, the only thing to do is to add a new function agent and make some slight modification to the information management program.

2) All software of the controller is running in RTOS system ensures real-time processing and guarantees that tasks inside a single module and different modules can be executed within time limit.

3) Modularized programming of software using C language makes it easy to be modified, extended and transplanted and meets the requirements of openness.

4) $\mu\text{C}/\text{OS-II}$ has the advantage of hardware abstraction. When the processor is changed, controller software can be easily transplanted to the new processor without any modification as soon as the $\mu\text{C}/\text{OS-II}$ system can be transplanted.

3 Experiments

A mobile robot platform named "ATU-II" is implemented using the controller presented above, as shown in Fig.4. ATU-II is a double-wheel driven mobile robot with four wheels. Sensors of ATU-II include one CCD camera, two sonar rings, one ring of tactile sensors and one ring of photoconductive sensor. Main hardware parameters of ATU-II are shown in table.1. Table.2 shows the performance of movement of ATU-II. As an experimental tool, ATU-II has been widely used in our research. The following section will show two experiments, which test the performance of ATU-II.

Table 1. Hardware parameters of ATU-II

Weight	20kg(with camera), 15kg(without camera)
Height	80cm(with camera), 60cm(without camera)
Radius	20cm
Line speed	2m/s
Rotate speed	360° /s

Table 2. The performance of movement

Moving manner	Warp to the standard
Straight ahead for 10m	<10cm
Turn left for 720 degree	<5degree
Turn right for 720 degree	<5degree
Straight back for 10m	<10cm

3.1 Navigation Experiment

In the experiment of navigation, the task of ATU-II is to avoid obstacles while finding the target. The target is a burning candle and the sensors used to find it is the ring of photoconductive sensors. Only one sonar ring is used for obstacle avoidance.

For this task, the multi-agent based software architecture is designed as Fig.5. The deliberative agents include dead-lock release agent and the task planning agent, which respectively correspond to the function of releasing dead-lock and planning the sub-goal of the robot. The reactive agents include obstacle avoidance agent which is used to avoid obstacle while moving and the target finding agent which is used to judge whether the robot has found the target. The deliberative agents and reactive agents both have the ability of self-learning, which is achieved by using reinforcement learning. For more detail of this architecture, please refer to [8].

This experiment is executed in the indoor environment as shown in Fig.6.a. Fig.6.b, 6.c, 6.d show the process of the experiment and the path that the robot moved is also shown in Fig6.a. It can be seen that ATU-II has successfully completed this task. During the experiment, ATU-II performed stably, which verified the stability of ATU-II.

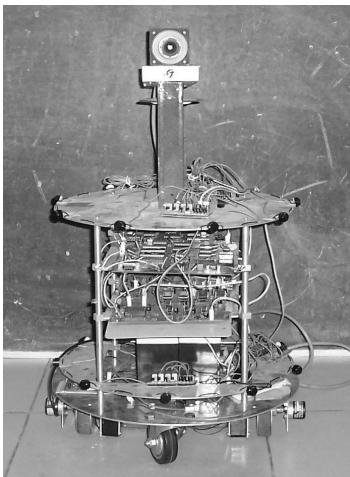


Fig. 4. ATU-II mobile robot

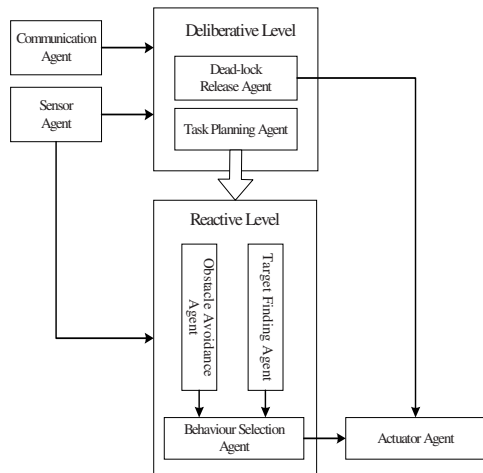


Fig. 5. Architecture for navigation experiment

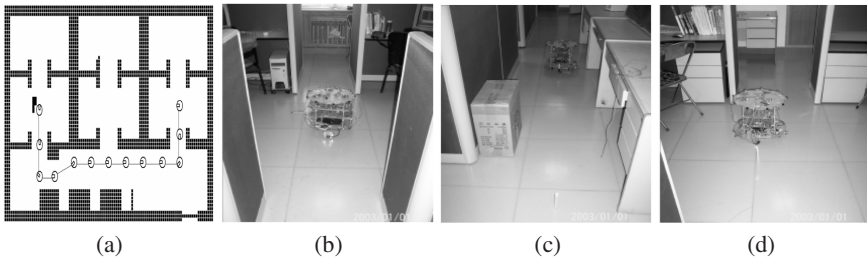


Fig. 6. Navigation experiment

3.2 Map Building Experiment

In the experiment of map building, the task of ATU-II is to build the map of the environment while exploring in it. The experiment is executed in the environment as Fig.7.a, and the built map is shown in Fig.7.b.

Comparing with the navigation experiment, one more sonar ring is used and the ring of photoconductive sensors is removed. In order to accomplish this, a module which controls the extended sonar ring is connected to the CAN Bus as a CAN node and the module which controls the ring of photoconductive sensors is removed from the CAN Bus. Due to the openness of hardware, this extension is very convenient to be implemented. The information flow between different agents of software structure is shown in Fig.7.c, which is fairly different with the navigation experiment, but it is still convenient to modify the system software to implement this task. What should be done is just to create the necessary agents and modify the information management program to fit the corresponding information flow. In a word, although the requirements for the mobile robot are fairly different between the navigation experiment and the map building experiment, it's easy to modify the software and hardware of the robot controller to switch from one task to the other, which proves the openness of the robot controller.

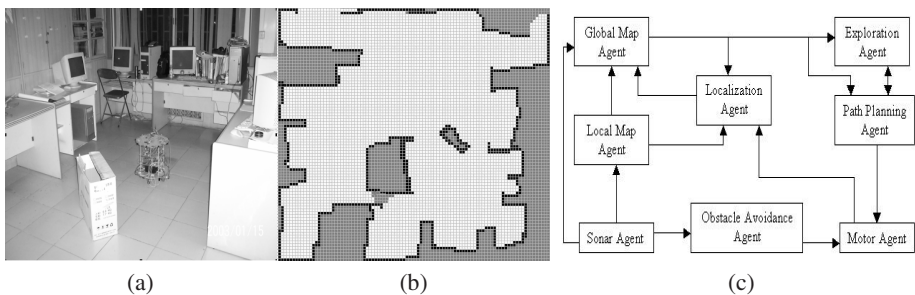


Fig. 7. Map-building experiment

This complex task is executed for many times in a fairly long time, but ATU-II works very well without any fault. The experimental results are shown in table.3. Its sound performance verifies the stability and real-time property of the robot controller. See [9] for more details of this experiment.

Table 3. Experimental results

Time for moving	Time for observing	Total time	Accuracy of the map
243s	99s	342s	88.6%

4 Conclusions

A mobile robot system should not only be an experimental platform, but also a prototype robot with the capability of being further developed into robot product for specific usage. To meet such requirement, a mobile robot controller based on CAN Bus and RTOS is presented. In hardware aspect, a distributed architecture is achieved using CAN Bus and Bluetooth technology. Each module of the controller is integrated into a multi-agent based architecture, and is implemented in RTOS in software aspect. A mobile robot platform named “ATU-II” is developed using such controller and its performance is verified by two experiments including navigation and map building.

Acknowledgements. This work has been supported by the National Natural Science Foundation of China under Grant No 60575033.

References

1. E.F. Willian: What is an Open Architecture Robot Controller. In: Proceedings of IEEE International Symposium on Intelligent Control, Columbus, Ohio, USA, pp.16--18(1994)
2. J.Q. Jia, W.D. Chen, Y.G. Xi: Design and Implementation of an Open Autonomous Mobile Robot System. In: Proceedings of IEEE International Conference on Robotics & Automation, New Orleans, LA, pp.1726--1731(2004)
3. K. Konolige, K. Myers: The Saphira Architecture for Autonomous Mobile Robots. SRI International(2000)
4. T. Balch. 2000. TeamBots. Available at World Wide Web, www.teambots.org
5. M. Lindstrom, A. Oreback, H. I. Christensen. Berra: A Research Architecture for Service Robots. In: Proceedings of IEEE International Conference on Robotics & Automation, San Francisco, CA, USA, pp.3278--3283(2000)
6. C.L. Chen, Z.H. Chen, G.M. Zhou: Multi-Agent Based Hybrid Architecture for Autonomous Mobile Robot. System Engineering and Electronics, 26(11):1746--1748(2004)
7. T. Wang: Design And Application Development of Embedded System. Tsinghua University Press, Beijing(2003)
8. R. Zhuo: A Multi-agent Hybrid Architecture of Mobile Robot Based on Reinforcement Learning. Master thesis, Department of Automation, University of Science & Technology of China(2005)
9. G.M. Zhou, Z.H. Chen Zonghai, N.Q. Liu, M.L. Jia: Mapping Algorithm of Mobile Robot in Unknown Indoor Environment Based on Active Exploration. Pattern Recognition and Artificial Intelligence, 19(5):591--597(2006)
10. C.L. Chen , Z.H. Chen: Reinforcement Learning for Mobile Robot: Form Reaction to Deliberation. Journal of System Engineering and Electronics, 16(3):611--617(2005)

Evolutionary Controllers for Snake Robots Basic Movements

Juan C. Pereda, Javier de Lope, and Maria Victoria Rodellar

Perception for Computers and Robots Group

Universidad Politécnica de Madrid

Campus de Montegancedo, 28660 Madrid

{jpgranados, jdlope}@dia.fi.upm.es, victoria@pino.datsi.fi.upm.es

Abstract. A method to generate movements in a snake robot using proportional-integral-derivate controllers (PID) and adjust the constants values to be natural is proposed. Specifically, the method is applied to adjust the movement of a snake robot to natural postures defining a simplify PID controller and adjusting the constants values of the controller. Our approach is based on proportional-integral-derivate controllers, using genetics algorithms to solve the problem. In this paper we explain how adjust the restrictions that must be accomplished for generate a natural movement and make an exhaustive study about snake robots, proportional-integral-derivate controllers and genetics algorithms.

Keywords: Snake robots, Proportional-integral-derivate controllers, Genetics algorithms.

1 Introduction

A snake can move in several different ways, but we can consider this in just one and unique movement. Thanks to this, the snake can reach places than other animals can not. These reptiles have a great stability and accessibility that it is why make them peculiar for different studies. These studies are necessary for understanding the dynamic and finally design, build and control snake robots.

The main goal in the design of snake-like robots is to imitate the body configuration and, therefore, the movements of biological snakes [1]. The snakes are able to crawl on almost any surface, including slopes, slippery ground or both. With more or less difficulty, the snake will arrive at the goal zone.

This kind of robots are mainly indicated when the objective is to reach zones that are difficult to access, for example, due to surface irregularities in which wheeled robots can not be used or because the workspace and the environment do not allow movements of conventional, wheeled or legged, robots. A classical application example of this kind of robots is the displacement in channels or pipes.

Another important topic in snake-like robots is the redundancy of joints. This redundancy extends the robot workspace and permits the robot head to reach more inaccessible places. In addition, the redundancy contributes to the robot being able to operate when one or more joints fail. This fact makes snake-like robots very suitable for highly hostile environments. Furthermore, snake-like robots are stable in almost all situations, only when they must climb or go over obstacles or unstructured terrain

can there be stability problems. Therefore, a difference with the gait development for other articulated robot is that no special care is necessary for preserving equilibrium.

Snake-like robots also present some disadvantages. For example, they cannot carry so much load, the advantage of a very efficient configuration for accessibility is not appropriate for the transportation of material between places. Another feature that can be interpreted as a disadvantage is the redundancy. The redundancy increases the work-space and maneuverability, but it also represents a higher complexity when we are controlling a redundant kinematics chain and In regards to the reliability of the electromechanical elements used in the construction. In addition, although the fastest biological snake can reach 3 m/s, currently its artificial counterparts are far from this speed, being one of the slowest robots.

According with the previous studies that it has been made about snake's movement, based in Hirose studies [1], it is suggested an equation system in which we can represent the biological movement of the snakes. In the majority of these studies based in the Hirose equation system, it is used a combination of links, joint and wheels for the design and construction of the robots. It is important to say that this is not the only design about snake's robots [2,3].

There are different kinds of studies in relation with this issue, based in direct the snake head wherever the robot want to go. It has to be considered that the rest of the robot has to pass through the same way that the head did. The disadvantage of these control systems it's that in the majority of the cases, there is not enough free redundant grades. That it is the consequence of a system control less effective and more complicate [4].

For resolving this problem it is suggested different designs of redundant controls like Matsuno and Mogi presented in their studies [5,6]. They suggested a redundant system control based on PID (proportional integral derivative controller). In this case, the velocity of the snake robot head does not determine the velocity of the different body joints. Using this redundancy it is possible to unify in the principal control goal the position of the snake robot in the space, and the position of the snake robot head and the way of the articulations of the body in different times.

The other approach is to define patterns generators whose outputs are applied consecutively and cyclically to the robot for producing the desired movement. Some of the most representative papers in this area are those by Shan and Koren's [7], in which a set of patterns are established for controlling the robot and make it move forward and turn, and Nilsson [8] who defines a pattern for climbing. We have extended these ideas in a previous work [9] by means of artificial neural networks.

Another research scope related to snake robots are polybots, originally proposed by Yim [10,11]. In this case the idea is to develop robots without any kind of limbs, composed by modules which have the same features. Depending on how this composition is made, they obtain different configurations and the most common configurations are similar to snakes or caterpillars. The underlying idea is that the robot can reconfigure itself for adapting to the environment.

In this paper we propose a solution based on PID controls and the redundant control system suggested by Matsuno and Mogi. In our case it is not necessary to add a term that it is going to correct the null values of the B matrix. One of the most important problems with PID controls is to adjust the proportional, derivative and integral constants. We are going to suggest a redundant equation system that describes

the movement of snake robots (Hard Computing) and using an evolutionary algorithm we are going to obtain the K_p , K_i , K_d constants of the equation (Soft Computing). We can adjust these constants to produce a more natural snake movement.

2 Proportional Integral Derivate (PID)

The transfer function of a PID controller is as a follow:

$$u = K_p e + K_i \int e dt + K_d de \div dt \quad (1)$$

where K_p is the proportional gain, K_i is the integral gain and, K_d is the derivative gain. The (e) variable represents the tracking error and it is the difference between the actual state and the future desirable state. In our case the error will be the difference between the positions of the articulation at that moment and the future position of the articulation. The error signal is sent to the PID controller and the controller executes the error derivate and integer. Doing this correction of the error, the articulation will move. This will make the robot go from initial point to the desirable point.

It will be introduce to the controller, the articulation position. The controllers return the position where the articulation should be moved and this generates a movement of the snake robot. Once the robot has been moved to this position, the process will be repeated, introducing the new position where the articulations are and the coordinates of the new position.

The proportional controller (K_p) has the effect of reducing the rise time, and reduces but never eliminates, the steady state error. The integral controller (K_i) has the effect of eliminate the steady state error, although it generates a worse system transitory response. The derivative controller (K_d) has the effect of increase the system stability, reducing considerably the “overshoot” and improving the transitory response. That kind of characteristic produces that K_p , K_i , K_d controllers depends one to each others. If we change one of the variables, it can produce a change in the other two.

One of the problems that we are going to find is the permanent error that generates the proportional controller. This error appears because the time is not considers in the proportional controller. That is why it is necessary to add some others components that consider the variation of the time. As we said before, it is so necessary to shape the integral and derivate actions.

However, in our case, the PID system control only has the proportional and the derivate parts. It is not necessary to add the integral part. Thanks to the proportional part we can reduce the rise time and adjusting the proportional constant we reduce until a satisfactory level the steady state error. We introduce the derivate constant to arise the system stability, reducing in that way the “overshoot” improving the transitory response. Nonetheless, we have eliminates in our system the integral controller. That is why adjusting the proportional constant we can reduce the steady state error to a reasonable level and in this way we can eliminate the effect that produce the integral constant. That effect makes a worse transitory response of the system. For ours operations we can consider that the K_i constant is equal 0. It is important to keep the derivate control because, as we mention before, the time is not

relevant in the proportional controller. According with these considerations the PID function can be simplify to PD. Following these instructions:

$$u = K_p e + K_d de \div dt \tag{2}$$

Furthermore, in this study we have to consider that the error (e) has been defined as the difference between the actual articulation position in the snake robot and the desirable future position of the snake robot. In this case, we can define the equation as a follow:

$$u = K_p (w-w_d) + K_d (w'_d) \tag{3}$$

where, w it is the actual position of the snake robot head, w_d is the desirable future position of the snake head that we want to achieve, and w'_d is the difference derivate.

According to the studies of F. Matsuno and K. Mogi [5,6] we must consider some design conditions of the snake robot. Although, K_p and K_d constants have to satisfy some other conditions to obtain the desirable solution.

The considerations of the snake robot design are:

1. The head link can not contain wheel.
2. The tail link must contain wheel.
3. The i -th joint is a control configuration point. The i -th link must be free of wheels.
4. The passive joint angle is equivalent to a variable state for being controlled as the shape controllable point.

In the other hand, considering the redundancy of our system and the K_p and K_d constants conditions, we can define our system as a follow:

$$Bu = Aw'_d - K(w-w_d) \tag{4}$$

Clearing up B from the equation we obtain:

$$u = B + Aw'_d - K(w-w_d) \tag{5}$$

Therefore, from this equation we can express as:

$$u = K_d w'_d + K_p (w-w_d) \tag{6}$$

where:

$$K_d = B + A \tag{7}$$

$$K_p = B + AK \tag{8}$$

So, based on the obtained results [5,6], the A matrix must satisfy that it is a full column rank matrix. This makes a unique system solution. B must satisfy that it is a full row rank.

However, unlike the system showed in [5,6], it is not necessary to add the $(I-B^+B)K$ term. This term correspond to the null values of the B matrix. That is why the value of

the A , B and K matrix and therefore the value of K_p and K_d constants are obtained with genetic algorithm. We are capable to adjust until the best combination of K_p and K_d .

3 Application of Genetics Algorithms for the Adjust of the K_p and K_d Constants

Adjust the constants values K_p , K_i , K_d is one of the most tedious and complicated part of a PID control system. We find another difficulty in the relation among these values. If you change one of them the others two may be modify. In our particular case, we only have to adjust and obtain the constants values the K_p and K_d . We can consider the K_i as a 0. This dose not makes the problem easier. This is because for obtaining K_p and K_d values we have to consider in every moment the other constant. As we have seen in the previous paragraph, our K_p and K_d constants are compounds for A , B and K matrix. The snake robot movement depends on these constants in a different way.

We have used a genetic algorithm for obtain these values because we think it is the best way for obtaining it. The algorithm will find the three A , B and K matrix for resolving the previous problem.

The genetic algorithm is a guided search method. That is the reason why we use a genetic algorithm to resolve our problem. Generally, a weak search conditions are introduced to the genetic algorithm to keep the elitism. The aim is to keep always the best of the population and make the algorithm probability converges to the optimum. In our case, the conditions have been suggested by the own system and the natural snake movement.

The system conditions have been suggested in the previous paragraph. In one hand we have the physic design, and in the other hand the A matrix has to be a full column rank and the B matrix has to be a full row rank. For select the most natural solutions and eliminate some dynamic correct movement we apply biological conditions to the robot.

4 Simulation

For carrying out our tests, we are going to create a 6-links snake robot. 3 of those 6 links will have free wheels. For free wheels we mean that those wheels will not be controlled by any engine therefore these wheels will move in the way of movement. Also the snake robot in which we are going to base will have 5 actives joints. Unlike passive joints we can control in every moment the position of these joints. Therefore we can adjust the posture of the snake robot in every moment. These physical characteristics of the snake robot make that A , B and K matrix from those lately we will obtain the proportional constant (K_p) and the derivation constant (K_d) are: $A^{(3 \times 3)}$ and $K^{(3 \times 3)}$ square matrix and $B^{(3 \times 5)}$ matrix. The form of the B matrix is based on 3 links with free wheels and the number of links that form the body less one that belongs to the head. The form of the A and K matrix are based on 3 links with free wheels and 3 plus the shape controllability index, in our case the value of the shape controllability is 0. All these values are base on Matsuno and Mogi studies, but in our case the K matrix is not necessary to be diagonal.

To obtain the matrix A , B and K constants values we use a genetic algorithm, as we said before. We will construct fitness function that is split in two parts. In the first part we apply PID theory but in this case we simplify it as we have mentioned before. We obtain the different joints values across the time. Next we obtain the space positions of the links and therefore we can obtain the posture of the snake robot. In the second part of the fitness function we penalize those postures that are not natural. The genetic algorithm uses the fitness function to select the best A , B and K constants matrix that generate the most natural movement.

For the simulation we have taken as initial joints values $[\pi/100; \pi/90; \pi/80; \pi/70; \pi/60]$ and as initial position we have considered that the head of the snake robot is situated in the position $[0,0]$ and with an angle of π radians (Fig. 1(a)). In this simulation we move the snake robot parallel the x axis. We can produce this kind of movement keeping the y coordinate constant.

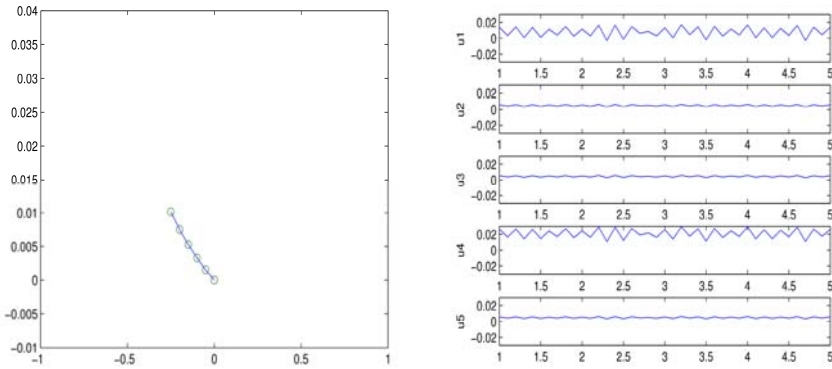


Fig. 1. (a) Initial position of the snake robot, (b) Transient responses for out u_1, u_2, u_3, u_4 and u_5

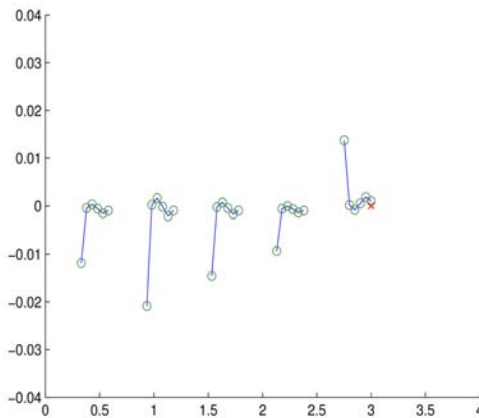


Fig. 2. Bad selection of restrictions in order to produce natural movements

Once we have got the best A , B and K matrix we can apply these values to the PD controller and obtain the movement of the joints that produce a change from one position to another Fig. 1(b). This change of position produces the final movement of the snake robot.

It is important to introduce a correct natural restriction for produce a natural movement in the snake robot. If the restriction is soft, the movement it is not natural as we want Fig. 2.

Finally, as we can see in Fig. 3 we can generate a natural movement of a snake if we introduce hard natural restrictions.

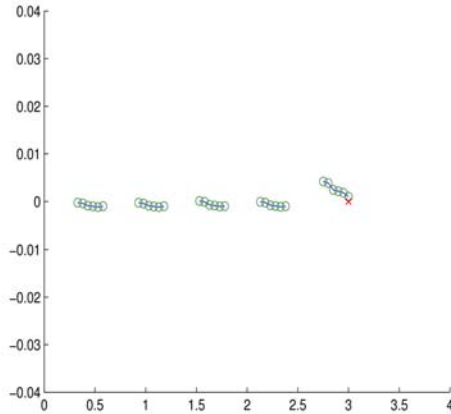


Fig. 3. Final natural movement of the snake robot

5 Conclusions and Further Work

As we have demonstrated in this article, applying a genetic algorithm, we can simplify a PID control system. We adjust the K_p and K_d constants to minimize the error value inside an acceptable range. Using genetic algorithm we can say that the values of K_p and K_d are optimal. Therefore we can guarantee that the movement of the snake robot is the most natural and fluent. In the simulation we use a 6 links snake robot. This kind of robots generate a $A^{(3 \times 3)}$ and $K^{(3 \times 3)}$ square matrix and a $B^{(3 \times 5)}$ matrix. Obtaining these results are trivial but if we change the snake robot to 10 or more links the matrix change to $A^{(5 \times 5)}$, $K^{(5 \times 5)}$ and a $B^{(5 \times 9)}$ matrices. In this case is not trivial anymore. In our case using genetic algorithm we can obtain these matrices independently of the number of links that have the snake robot.

Currently we are working on introduce a new restriction in the genetic algorithm to find the best constants values of the PID that control the robot with the smaller consume.

Acknowledgments. This work has been partially funded by the Spanish Ministry of Science and Technology, project DPI2006-15346-C03-02.

References

1. S. Hirose. *Biologically Inspired Robots (Snake Like Locomotor and Manipulator)*. Oxford University Press, 1993.
2. S. Shiotani, S. Hamada, T. Oomichi, Y. Ozawa, S. Tsuzuki. Development of Hyper-Degree of Freedom Control Method for Holonic Mechanism. In *Proc. COE Super Mechano-Systems Workshop*, pp. 89–92, 1999.
3. N. Takanashi, S. Yashima, K. Aoki, T. Nishizawa. Complete Modular Links for Hyper Redundant Robots. In *Proc. COE Super Mechano-Systems Workshop*, pp. 93–98, 1999.
4. P. Prautesch, T. Mita, H. Yamauchi, T. Iwasaki, G. Nishida. Control and Analysis of the Gait of Snake robots. In *Proc. COE Super Mechano-Systems Workshop*, pp 257–265, 1999.
5. K. Mogi, F. Matsuno. Control of a Snake Robot with Redundancy Based on Kinematic Model. In *Proc. 5th. Int. Symposium of Artificial Life and Robotics*, pp. 507–510, 2000.
6. K. Mogi, F. Matsuno. Redundancy Controllable System and Control of Snake robots Based Kinematic Model. In *Proc. 39th IEEE Conference on Decision and Control*, pp. 4791–4796, 2000.
7. Y. Shan, Y. Koren. Design and Motion Planning of a mechanical Snake. *IEEE Trans. on Systems, Man, and Cybernetics*. 23(4):1091–1100, 1993.
8. M. Nilsson. Snake Robot Free Climbing. *IEEE Control Systems*, pp. 21–26, Feb. 1998.
9. J.A.Becerra, F. Bellas, J. de Lope, R. Duro. Snake like Behaviours using Macroevolutionary Algorithms and Modulation Based Architectures. In *Proc. FLINS 2006*.
10. M. Yim. New Locomotion Gaits. *Proc. IEEE-ICRA'94*, pp. 2508–2514, 1994.
11. M. Yim et al. Modular Self-Reconfigurable Robot Systems [Grand Challenges of Robotics]. *IEEE Robotics & Automation Mag.* pp. 43–52, March 2007.

Evolution of Neuro-controllers for Multi-link Robots

JoséAntonio Martí H.¹, Javier de Lope², and Matilde Santos¹

¹ Facultad de Informática, Universidad Complutense de Madrid
jamartinh@fdi.ucm.es, msantos@dacya.ucm.es

² Dept. Sistemas Inteligentes Aplicados, Universidad Politécnica de Madrid
javier.delope@upm.es

Abstract. A general method to learn the inverse kinematics of multi-link robots by means of neuro-controllers is presented. We can find analytical solutions for the most used and known robots in the bibliography. However, these solutions are specific to a particular robot configuration and are not generally applicable to other robot morphologies. The proposed method is general in the sense that it is not dependant on the robot morphology. We base our method in the Evolutionary Computation paradigm for obtaining incrementally better neuro-controllers. Furthermore, the proposed method solves some very specific issues in robotic neuro-controller learning. (1) It allows to escape from any neural network learning algorithm which relies on the classical supervised input-target learning scheme and hence it lets to obtain neuro-controllers without providing targets or correct answers which -in this case- are unknown in prior. (2) It can converge beyond local optimal solutions which is one of the main drawbacks of some neural-network training algorithms based on gradient descent when applied to highly redundant robot morphologies. (3) Using learning algorithms such as the Neuro-Evolution of Augmenting Topologies (NEAT) it is also possible learning the neural network topology on-the-fly which is a common source of empirical testing in neuro-controllers design. Finally, experimental results are provided by applying the method in two multi-link robot learning tasks with a comparison between fixed and learnable topologies.

Keywords: Neuro-Evolution, Multi-Link Robots, Inverse Kinematics, Reinforcement Learning.

1 Introduction

One of the most traditional areas in Robotics is dedicated to the study of kinematics of articulated robots. As it is well known, robot kinematics establishes a mechanism to determine the relationship between the joint and Cartesian coordinates. The direct kinematics problem has been successfully solved in the middle of the past century [1]. Currently there exist automatic procedures to assign the different parameters to every robot's joint and obtaining the end-effector position and orientation referred to a base frame for each point expressed in the joint coordinate frame. Unfortunately the inverse kinematics problem has not been solved analytically in a general way. We can find analytical solutions for the most used and known robots in the bibliography. However, the drawback of these solutions is that they are specific to a particular robot configuration and are not applicable to other robots. The problem gets worst if we consider that currently several kinds of new articulated robots are being proposed for

different tasks. Humanoid and modular robots are usual topics in the most important international conferences on Robotics. These multi-articulated robots make inapplicable some of the classical procedures to calculate the inverse kinematics mainly due to such methods assume some specific configurations which are not used in the new articulated robots. The new articulated robots are inherently multi-redundant. For example, a simple biped robot needs 12 degrees-of-freedom (DOF) to reach the most common configurations required for realistic postures. A whole humanoid robot has around 30 DOFs, depending on the number of modeled joints. There is not a predefined value for the case of modular and serpentine robots. This degree of redundancy makes the development of an analytic solution for the inverse kinematics practically unfeasible.

In this paper we propose a general method in the sense that it is not dependant on the robot morphology. We compare two different learning algorithms in order to obtain the best neuro-controllers to solve the inverse kinematics of two models of robot manipulators.

2 Description of the Proposed Methods

Traditionally it is considered there are two major methods to train artificial neural networks by means of evolutionary strategies. The first one evolves the network weights trying to minimize the error in the output layer related to a previously defined dataset or a simulated environment. Apart from this parametric evolution, there is also a kind of structural learning in which the network topology is also evolved [2]. Again, two categories can be defined: the constructive methods, in which starting from a simple network, the nodes and links are added and destructive methods in which the process starts with large networks and the superfluous components are pruned off. We introduce and compare two methods for neural network training, the first is an example of the parametric approach and the second also modifies the structural aspects of the neural networks.

2.1 The CMA Evolution Strategy

The covariance matrix adaptation evolution strategy (CMA-ES) [3] is one of the most powerful evolutionary algorithms for real-valued optimization. By this reason we have considered to compare it with the NEAT algorithm.

The covariance matrix adaptation is a de-randomized method to adapt the covariance matrix of the multivariate normal mutation distribution in the Evolution strategy. The covariance matrix describes the pair-wise dependencies between the variables. The adaptation of the covariance matrix leads to learning a second order model of the underlying objective function similar to the approximation of the inverse Hessian matrix in the Quasi-Newton method in classical optimization. In general the adaptation principle is based on the idea of increasing the probability of a successful mutation step. The covariance matrix is changed such that the likelihood of the successful steps of the last generation to appear again is increased. The adaptation of

the multivariate normal distribution in the CMA evolution strategy consists of two parts: (1) adaptation of the covariance matrix, which can be interpreted as executing an online-PCA, and (2) adaptation of the overall step length (step-size), implemented by measuring the length of the evolution path of the population.

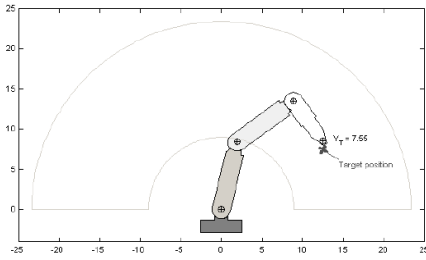
2.2 Neuro-Evolution of Augmenting Topologies (NEAT)

NEAT [4] is a method for evolving artificial neural networks proposed as an alternative to other methods which do not modify the network topology during the evolution process. Both kinds of methods present advantages for a particular problem. The advantages are related to the saved time of having to find the right number of hidden neurons by methods which evolve topologies. However, evolving topology along with weights might make the search more difficult. The NEAT's genomes are linear representations of network connectivity. Thus, the genes are easily lined up during the crossover. Each genome includes a list of connection genes, each of which refers to two node genes being connected. Each connection gene specifies the in-node, the out-node, the weight, a value which determines if the connection gene is enabled, and a value the innovation number that represents the instant in which the gene was added to the genome and it is used during the crossover to create more easily an offspring. By using the innovation number mechanism, it can be easily determined which genes match up with which. The basic idea is to use a counter which is incremented when a new gene appears, assigning the value at this moment to the gene. So, individuals which share genetic information can be detected easily due to the innovation number in their genes. Then, the crossover generates a new offspring from two parents in which are included the shared genetic material and also the new contributed genes which are only present in one of the parents. The mutation operator can change both the connection weights and the network structure. The weights are mutated as usual; each connection may or may not be perturbed at each generation. The network structure mutation expands the size of the genome by adding one or more genes and enabling or disabling node connections.

3 Experimental Application Framework Definition

Experiments have been realized on two multi-link robot models: a three-link planar robot and a SCARA robot. The experiments are made with the intention that the robots learn their inverse kinematics in order to reach a desired configuration in the joint and Cartesian spaces.

The three link planar robot includes some redundancy in the calculus of the inverse kinematics by means of analytic procedures. For this reason, it is very interesting for the verification of our methods in this kind of mechanical structures. The same approach could be applied for controlling the movement and body coordination of, for example, snake-like robots. Fig. 1 shows a diagram of this robot and the D-H parameters. Its direct kinematics is defined in (1).



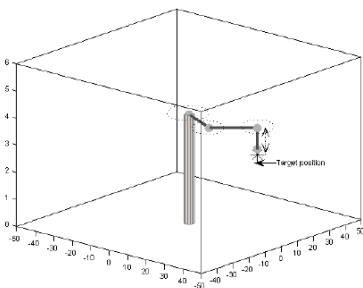
i	θ_i	d_i	a_i	α_i
1	θ_1	0	$a_1 = 8.625$	0
2	θ_2	0	$a_2 = 8.625$	0
3	θ_3	0	$a_3 = 6.125$	0

Fig. 1. Three link-planar robot model with its D-H parameters

$$T^3 = \begin{pmatrix} C_{123} & -S_{123} & 0 & a_1 C_1 + a_2 C_{12} + a_3 C_{123} \\ S_{123} & C_{123} & 0 & a_1 S_1 + a_2 S_{12} + a_3 S_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{1}$$

where S_{123} and C_{123} correspond to $\sin(\theta_1 + \theta_2 + \theta_3)$ and $\cos(\theta_1 + \theta_2 + \theta_3)$, respectively (and equally for S_1, S_{12}, C_1 and C_{12}), θ_1, θ_2 and θ_3 define the robot joint coordinates --- θ_1 for the first joint, located near to the robot base, θ_2 for the middle joint and θ_3 for the joint situated in the final extreme and a_1, a_2 and a_3 correspond to the physical length of every link, first, second and third, respectively, numbering from the robot base.

The SCARA robot was selected since it is widely used in the industry. It has well-known properties for the use as manufacturing cells assisting conveyor belts, electronic equipment composition, welding tasks, and so on. In this case we are using a real three dimension robot manipulator. A physical model of the SCARA robot is shown in Fig. 2 with its corresponding D-H parameters. The direct kinematics matrix is shown in (2).



i	θ_i	d_i	a_i	α_i
1	θ_1	0	$a_1 = 20$	0
2	θ_2	0	$a_2 = 20$	π
3	θ_3	0	0	0
4	0	d_4	0	0

Fig. 2. SCARA robot model with its D-H parameters

$$T^4 = \begin{pmatrix} C_{12-3} & -S_{12-3} & 0 & a_1 C_1 + a_2 C_{12} \\ S_{12-3} & C_{12-3} & 0 & a_1 S_1 + a_2 S_{12} \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{2}$$

where $S_{12,3}$ and $C_{12,3}$ correspond to $\sin(\theta_1 + \theta_2 + \theta_3)$ and $\cos(\theta_1 + \theta_2 + \theta_3)$, respectively (and equally for S_1, S_{12}, C_1 and C_{12}), θ_1, θ_2 and θ_3 and d_4 are the joint parameters for the shoulder, elbow, wrist and prismatic joints, respectively, and a_1 and a_2 the lengths of the arm links.

3.1 Experiments Design

The control task consists of reaching a given target position of the robot end effector by means of a neuro-controller. The goal positions are defined in such a way that they are always reachable for the robot. The robots are controlled by means of assigning a desired angle position to each articulated link, thus if the goal (target position) is to reach the point p the neuro-controller must provide the robot with a vector of angular values $\theta(p)$ in order that when applying a direct kinematics method using $\theta(p)$ the position (p') of the robot end-effector minimizes the MSE error $\|p-p'\|^2$. Through the different trials, a model of the robot inverse kinematics is learned by the system by means of an evolutionary program that evolves neuro-controllers. For learning purposes we use a fixed random generated set of 70 different points. Although probably using a non-fixed random generated set of targets may force the learning process to obtain initial networks that generalize well over the complete space of targets, the inclusion of non-fixed training targets adds a huge amount of complexity to the optimization cost function (fitness function) due to the effect of evaluating individuals for different target positions which is actually a lot of noise and hence the evolutionary process may become unstable and this can result in poor performance and undesirable side effects. By this reason a fixed set of training goals is highly preferable in this kind of learning problems combined with an adequate topology design in order to prevent over-fitting and thus promoting good generalization capabilities. We have used for all of our experiments a population size of 50 individuals over 1000 generations.

We have tested mainly two different approaches. The former uses a fixed topology specifically a multi-layer perceptron which is evolved with the CMA-ES algorithm. The second approach uses NEAT to incrementally learn the network weights and at the same time learn the more appropriate network topology. Since CMA-ES is applied to a fixed topology network the process of designing the individual's genotype (vector of real numbers) and phenotype (neural-networks) is trivial. Each individual's genotype is a vector whose length is determined by three fixed variables:

($ni = 2 + \text{bias}$) the number of input nodes, ($nh = 5$) the number of hidden nodes in the hidden layer and ($no = 3$) the number of output nodes of the multi-layer perceptron. For constructing an individual's phenotype, that is, its respective neural network, we need just to extract the corresponding segments of the genotype vector in order to create the weights matrix of each layer.

For the NEAT algorithm the process of transforming the individual's genotype involves the determination of the number of hidden nodes and its respective non-disabled connections. Basically the process consists of a loop through any starting node and to find its corresponding end nodes in order to contribute to the total activations of the end nodes. This process is repeated up to any connection has been processed. For NEAT networks we use the same activation function as in the CMA-ES networks (hyperbolic tangent) which allow handling in more easy way both

positive and negative values. Also we have determined empirically that this activation function (*tanh*) is best suited for this kind of task where the interesting output values are continuous instead of (0,1) on-off.

We have used the next functional parameters for the NEAT algorithm; the more relevant parameters which also are the most problem dependant are presented.

- connected input nodes = all which means that the algorithm will start with all connections for the inputs nodes enabled.
- mutation probability add node = 0.002
- mutation probability add connection = 0.002
- mutation probability mutate weight = 0.5
- mutation weight cap = 1000 weights will be restricted from (-mutation weight cap to mutation weight cap)
- mutation weight range = 1 random distribution with width mutation weight range, centered on 0.
- mutation probability gene re-enabled = 0.25, probability of a connection gene being re-enabled in offspring if it was inherited disabled..

As we can see the [probability add node] and [probability add connection] parameters are fixed to a very small value. This is done with the purpose of preventing premature growth of the network topology letting small topologies to have enough time to converge.

4 Experimental Results

Fig. 3 and 4 shows the main results of our experimental work. We must recall that for each robot we have followed two approaches, a fixed topology with the CMA-ES algorithm and topology learning with the NEAT algorithm. The graphs showed below shows the behavior of these two methods for each robot, so we can see easily the convergence curves of both methods. Generalization tests were made with new random target positions showing that the learned neuro-controllers generalize very well over the whole space showing a deviation of 0.001% of the error mark obtained during the learning process. This is consistent with our claim that it is better to train the neuro-controllers with a good representative set of fixed targets positions instead of random targets positions for the learning process that will introduce noise in the cost function and may result in poor convergence.

Fig. 3 shows the convergence curve of both methods for the problem of learning the inverse kinematics of the three-link planar robot. As we can appreciate, CMA-ES with fixed topology outperforms the NEAT algorithm at the end producing neuro-controllers with very low error while NEAT remains trapped. One aspect that is relevant is that NEAT was able to find good compromise solution in more short time than CMA-ES up to generation 151 when the lines cross and CMA-ES starts to adapt better to the problem. These results agree with the claims in [4] that starting to learn with a reduced topology and gradually adapt the network topology to the task may increase convergence speed. Although NEAT does a very good job at the first stages of the optimization problem CMA-ES is a more theoretically developed method

designed for numerical optimization which is mainly the reason why after the crossing line CMA-ES continues to adapt and to find more and more optimal solutions.

The general results shows that CMA-ES reaches a low quadratic error mark of $MSE = 0,1673$ which means a high accuracy while NEAT get stuck in the error mark $MSE = 5,849$ which is not bad but a little far from CMA-ES.

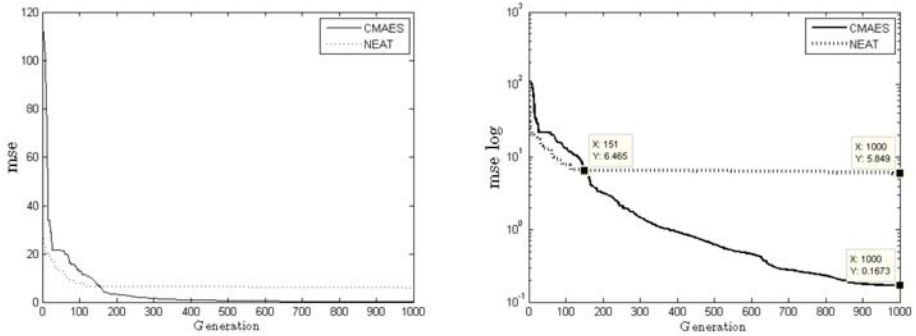


Fig. 3. Best convergence curve for the planar robot (logarithmic scale) over 15 runs

Fig. 4 shows the convergence curve of both methods for the problem of learning the inverse kinematics of the SCARA robot. As we can appreciate, CMA-ES with fixed topology again outperforms the NEAT algorithm at the end producing neuro-controllers with very low error while NEAT again remains trapped. As we can see this time NEAT was able to find good compromise solution in more short time that CMA-ES like in the previous experiment indeed up to generation 142 which is very near to the previous experiment. Again when the lines cross CMA-ES starts to adapt better to the problem.

The general results shows that CMA-ES reaches a low quadratic error mark of $MSE= 0,1989$ which means a high accuracy while NEAT get stuck in the error mark $MSE= 14,68$ which is not bad but a little far from CMA-ES.

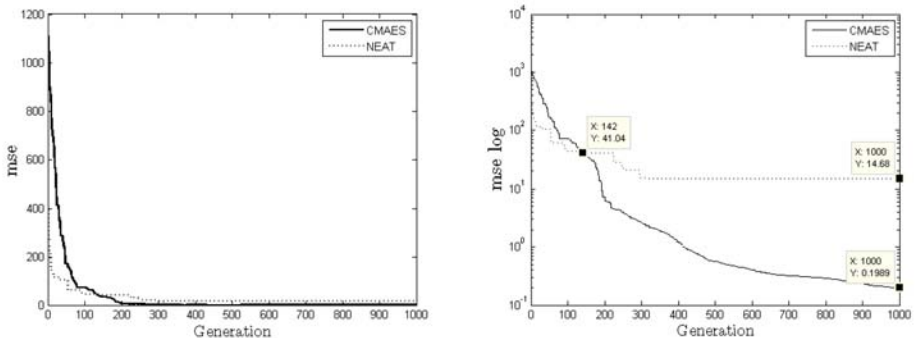


Fig. 4. Best convergence curve for the SCARA robot (logarithmic scale) over 15 runs

5 Conclusions

A general method to learn the inverse kinematics of multi-link robots by means of neuro-controllers was presented. The experimental results for both robot models are almost identical. These reiterative results shows that NEAT is a good alternative for very complicated problems where the network topology cannot be found by means of empirical test, but also, that CMA-ES is clearly superior when a good topology is known in advance. All of this suggests that more experimental work must be done in order to improve the convergence properties of the NEAT algorithm with respect to weights optimization.

Acknowledgements. This work has been partially funded by the Spanish Ministry of Science and Technology; project DPI2006-15346-C03-02.

References

1. J. Denavit and R. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME J. Applied Mechanics*, pp. 215-221, June 1955.
2. P.J. Angeline, G.M. Saunders, and J.B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Trans. Neural Networks*, 5(1):54-65, 1994.
3. N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. *In Int. Conf. on Evolutionary Computation*, pp 312-317, 1996.
4. K.O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evol. Comput.*, 10(2):99-127, 2002.

Solving Linear Difference Equations by Means of Cellular Automata

A. Fuster-Sabater^{1,*}, P. Caballero-Gil², and O. Delgado¹

¹ Instituto de Física Aplicada, C.S.I.C., Serrano 144, 28006 Madrid, Spain
{amparo, oscar.delgado}@iec.csic.es

² DEIOC, University of La Laguna, 38271 La Laguna, Tenerife, Spain
pcaballe@ull.es

Abstract. In this work, it is shown that linear Cellular Automata based on rules 90/150 generate all the solutions of linear difference equations with binary constant coefficients. Some of these solutions are binary sequences with application in cryptography. In this sense, we propose CA-based linear models that realize the solutions of difference equations as well as behave as cryptographic keystream generators. Due to the simple transition rules that govern these CA, the implementation of such models is quite easy. Some illustrative examples complete the work.

Keywords: Cellular automata operations, cryptographic sequences, CA-based models.

1 Introduction

Nowadays, stream ciphers are the fastest among the encryption procedures so they are implemented in many practical applications e.g. the algorithms A5 in GSM communications [10], the encryption system E0 in Bluetooth specifications [2] or the algorithm RC4 [14] used in Microsoft Word and Excel. From a short secret key (known only by the two interested parties) and a public algorithm (the sequence generator), stream cipher procedure consists in generating a long sequence of seemingly random bits. Such a sequence is called the keystream sequence. For encryption, the sender realizes the bit-wise (Exclusive-OR) XOR operation among the bits of the original message or plaintext and the keystream sequence. The result is the ciphertext to be sent. For decryption, the receiver generates the same keystream, realizes the same bit-wise XOR operation between the received ciphertext and the keystream sequence and recuperates the original message.

Most keystream generators are based on Linear Feedback Shift Registers (LFSRs) [8]. Their output sequences (the *PN*-sequences) are combined by means of nonlinear functions in order to produce keystream sequences of cryptographic application. Combinational generators, nonlinear filters, clock-controlled generators, irregularly decimated generators ... are just some of the most popular nonlinear sequence generators. All of them produce keystreams with high linear complexity, long period

* Corresponding author.

and good statistical properties (see [6] and [3]). On the other hand, it can be proved [4] that certain one-dimensional linear Cellular Automata (CA) generate exactly the same PN -sequences as those of the LFSRs. Thus, LFSRs and CA can be easily interchanged.

In this work, it is showed that linear CA based on rules 90/150 generate all the solutions of linear difference equations with binary constant coefficients. Some of these solutions correspond to the sequences produced by the previous keystream generators. In this way, we have simple CA that not only generate all the solutions of a kind of equation but also they are linear models of nonlinear cryptographic sequence generators. Due to the simplicity of the CA transition rules, the practical implementation of these CA-based design is adequate with VLSI technologies.

The work is organized as follows. In section 2, we present the kind of difference equations we are dealing with. The particular form of CA that generate the solutions of the previous equations are introduced in section 3. Cryptographic sequences as solutions of the above mentioned equations as well as a simple linearization algorithm are described in section 4. Several illustrative examples complete the work.

2 Linear Difference Equations with Constant Coefficients

In this work, we will consider the following kind of linear difference equations:

$$(E^r \oplus \sum_{j=1}^r c_j E^{r-j}) a_n = 0, \quad n \geq 0 \quad (1)$$

where E is the shifting operator that operates on a_n (i.e. $Ea_n = a_{n+1}$), $\{a_n\}$ with $a_n \in GF(2)$ is a binary satisfying the previous equation, $c_j \in GF(2)$ are binary constant coefficients and the symbol \oplus represents the XOR logic operation. The r -degree characteristic polynomial of the equation (1) is:

$$P(x) = x^r + \sum_{j=1}^r c_j x^{r-j} \quad (2)$$

and specifies the linear recurrence relationship of the sequence $\{a_n\}$. This means that its n -th element a_n can be written as a linear combination of the previous elements:

$$a_n \oplus \sum_{j=1}^r c_j a_{n-j} = 0, \quad n \geq r. \quad (3)$$

If the the characteristic polynomial $P(x)$ is primitive (a particular kind of irreducible polynomial) [8] and α one of its roots, then

$$\alpha, \alpha^2, \alpha^{2^2}, \dots, \alpha^{2^{(r-1)}} \in GF(2^r) \quad (4)$$

are the r different roots of such a polynomial (see [13]). In this case, the solutions of (1) are of the form:

$$a_n = \sum_{j=0}^{r-1} A^{2^j} \alpha^{2^j n}, \quad n \geq 0 \tag{5}$$

where A is an arbitrary element in $GF(2^r)$. According to equation (5), $\{a_n\}$ is a PN -sequence [8] whose starting point is given by the value of A .

Next we can generalize the difference equation given in (1) to a more complex kind such as follows:

$$(E^r \oplus \sum_{j=1}^r c_j E^{r-j})^p a_n = 0, \quad n \geq 0 \tag{6}$$

where p is a positive integer. In this case, the characteristic polynomial of (6) is of the form $P_c(x) = P(x)^p$ and its roots will be the same as those of $P(x)$ but with multiplicity p . The solutions of (6) are of the form:

$$a_n = \sum_{m=0}^{p-1} n \binom{r-1}{m} A_m^{2^j} \alpha^{2^j n}, \quad n \geq 0 \tag{7}$$

where the A_m are arbitrary elements in $GF(2^r)$. According to equation (7), $\{a_n\}$ is now the bit-wise XOR of p different sequences where each of them is the same PN -sequence as before $\{\sum_{j=0}^{r-1} A_m^{2^j} \alpha^{2^j n}\}$ generated by $P(x)$ starting at a particular point

determined by the value of A_m and weighted by a binomial number $\binom{n}{m}$. Different choices of A_m will give rise to different sequences $\{a_n\}$. Our analysis focuses on all the possible solutions of the equation (6).

3 Synthesis of Cellular Automata Generating the Solutions of Difference Equations

CA are particular forms of finite state machines defined as uniform arrays of identical cells in an n -dimensional space (see [12]). The cells change their states (contents) synchronously at discrete time instants. The next state of each cell depends on the current states of the neighbor cells according to its transition rule. If the transition rules are all linear, so will be the automaton under consideration. In the previous section, the solutions of linear difference equations with constant coefficients have been considered. Now the particular form of the CA able to generate such solutions is analyzed. This work is concentrated on one-dimensional binary linear CA with L cells and particular transition rules defined as follows (Wolfram notation [15]):

Rule 90: $x_{n+1}^i = x_n^{i-1} \oplus x_n^{i+1}$ Rule 150: $x_{n+1}^i = x_n^{i-1} \oplus x_n^i \oplus x_n^{i+1}$ where x_n^i ($i = 1, 2, \dots, L$) represents the binary content of the i -th cell at time n and the symbol \oplus represents the XOR logic operation. In addition, cells with permanent null contents are supposed to be adjacent to the array extreme cells. Remark that the transition rules are linear and involve just the addition of either two bits (rule 90) or three bits (rule 150). Consequently their implementation is quite simple.

For a cellular automaton of length $L = 10$ cells, configuration rules (90,150, 150,150,90,90,150,150,150,90) and initial state (0,0,0,1,1,1,0,1,1,0) Table (1) illustrates the behavior of this structure: the formation of its output sequences $\{x_n^i\}$ ($i = 1, \dots, 10$) (binary sequences read vertically) and the succession of states (binary configurations of 10 bits read horizontally).

Table 1. An one-dimensional linear cellular automata of 10 cells with rules 90/150

90	150	150	150	90	90	150	150	150	90
0	0	0	1	1	1	0	1	1	0
0	0	1	0	0	1	0	0	0	1
0	1	1	1	1	0	1	0	1	0
1	0	1	1	1	0	1	0	1	1
0	0	0	1	1	0	1	0	0	1
0	0	1	0	1	0	1	1	1	0
0	1	1	0	0	0	0	1	0	1
1	0	0	1	0	0	1	1	0	1
0	1	1	1	1	1	0	0	0	0
1	0	1	1	1	1	1	0	0	0

A natural way of representation for 90/150 linear CA is a binary L -tuple (rule vector) $\Delta = (d_1, d_2, \dots, d_L)$ where $d_i = 0$ if the i -th cell satisfies rule 90 while $d_i = 1$ if the i -th cell satisfies rule 150. Given a characteristic polynomial, the Cattell and Muzio synthesis algorithm [4] presents an easy method of computing two 90/150 CA corresponding to such a polynomial. The input algorithm is a primitive (in general irreducible) polynomial $Q(x)$ and the output are two reversal L -tuples corresponding to two different linear CA. The sequences generated by such L -tuples have $Q(x)$ as characteristic polynomial. The total number of operations required for this algorithm is linear in the degree of the polynomial see [4] (pp. 334).

Since the characteristic polynomials we are dealing with are of the form $P_G(x) = P(x)^p$, it seems quite natural to construct a cellular automaton by concatenating a number of times the automaton whose characteristic polynomial is $P(x)$. The procedure of concatenation is based on the following result.

Lemma 1. Let $\Delta = (d_1, d_2, \dots, d_L)$ be the representation of an one-dimensional binary linear cellular automaton with L cells and characteristic polynomial $P_L(x)$ of degree

L . The cellular automaton whose characteristic polynomial is $P_{2L}(x) = P_L(x)^2$ is represented by:

$$\Delta = (d_1, d_2, \dots, \overline{d_L}, \overline{d_L}, \dots, d_2, d_1) \tag{8}$$

where the overline symbol represents bit complementation.

Proof: The result follows from the fact that:

$$P_L(x) = P_L(x) + P_{L-1}(x)$$

where $P_L(x)$ is the polynomial corresponding to $\Delta = (d_1, d_2, \dots, \overline{d_L})$ and $P_{L-1}(x)$ the polynomial corresponding to $(d_1, d_2, \dots, d_{L-1})$. In the same way

$$P_{L+1}(x) = (x + d_L)P_L(x) + P_L(x)$$

$$P_{L+2}(x) = (x + d_{L-1})P_{L+1}(x) + P_L(x)$$

$$\vdots \quad \quad \quad \vdots$$

$$P_{2L}(x) = (x + d_1)P_{2L-1}(x) + P_{2L-2}(x).$$

Thus, by successive substitutions of the previous polynomial into the next one we get:

$$P_{2L}(x) = (x + d_1)P_{2L-1}(x) + P_{2L-2}(x) = P_L(x)^2. \tag{9}$$

The result can be iterated a number of times for successive polynomials and rule vectors:

with $P_L(x)$ we get $\Delta_L = (d_1, d_2, \dots, d_L)$,

with $P_L(x)^2$ we get $\Delta_{2L} = (d_1, d_2, \dots, \overline{d_L}, \overline{d_L}, \dots, d_2, d_1)$,

with $P_L(x)^{2^2}$ we get $\Delta_{2^2L} = (d_1, d_2, \dots, \overline{d_L}, \overline{d_L}, \dots, d_2, \overline{d_1}, \overline{d_1}, d_2, \dots, \overline{d_L}, \overline{d_L}, \dots, d_2, d_1)$

and so on. Thus, the concatenation of an automaton (with the least significant bit complemented) and its mirror image allows us to synthesize a new linear cellular automaton that includes the previous one. In fact, the automaton Δ_{2^kL} includes all the previous sub-automata Δ_{2^sL} with $0 \leq s < k$, that is the automaton Δ_{2^kL} generates all the sequences $\{x_i^n\}$ of the previous sub-automata whose characteristic polynomials are $P_L(x)^p$ with $1 \leq p \leq 2^k$. The choice of a particular initial state determines the corresponding characteristic polynomial $P_L(x)^p$ of the generated sequences. In brief, we have concatenated structures able to produce all the binary sequences whose characteristic polynomials are of the form $P(x)^p$ with $1 \leq p \leq 2^k$. That is to say, we have obtained in an easy way CA-based linear models that generate all the solutions of the equation (6).

4 Cryptographic Sequences as Solutions of Linear Difference Equations

In this section, we show how some of the previous sequential solutions have application in stream ciphers. In the literature, there are different families of pseudo-random sequences, the so-called interleaved sequences [9], with the common characteristic that each sequence can be written in terms of a unique shifted PN -sequence. Interleaved sequences are currently used as keystream sequences and they are generated: 1) by means of a LFSR controlled by another LFSR e.g. multiplexed sequences [11], clock-controlled sequences [1], cascaded sequences [7], shrinking generator sequences [5] etc or 2) by means of one or more than one LFSR and a feed-forward nonlinear function e.g. Gold-sequence family, Kasami (small and large set) sequence families, GMW sequences, Klapper sequences, No sequences etc. See [9] and the references cited therein. In brief, a large number of popular cryptographic sequences are included in the class of interleaved sequences.

On the other hand, since the characteristic polynomial of the interleaved sequences is $P(x)^p$, it is clear that all this kind of sequences are solutions of the equation (6) and can be generated by means of the CA-based concatenated model. In this way, linear CA can be used as sequence generators of cryptographic application.

As illustrative example, we develop the CA-based modelling of a very representative keystream generator: the shrinking generator.

4.1 CA-Based Model for the Shrinking Generator

The shrinking generator is a binary sequence generator [5] composed by two LFSRs: a control register R_1 that decimates the sequence produced by the other register R_2 . We denote by L_j ($j=1,2$) their corresponding lengths and by $P_j(x)$ ($j=1,2$) their corresponding characteristic polynomials of degree L_j ($j=1,2$), respectively.

The sequence produced by R_1 , denoted by $\{a_i\}$, controls the bits of the sequence produced by R_2 , that is $\{b_i\}$, which are included in the output sequence $\{z_j\}$ (*the shrunken sequence*), according to the following rule:

- If $a_i = 1 \Rightarrow z_j = b_i$
- If $a_i = 0 \Rightarrow b_i$ is discarded.

In brief, the sequence produced by the shrinking generator is an irregular decimation of $\{b_i\}$ from the bits of $\{a_i\}$. According to [5], its characteristic polynomial is $P(x)^{2^{L_1}-1}$ where $P(x)$ is an L_2 -degree polynomial, so that the shrunken sequence is a particular solution of the linear difference equation (6). Now the construction of such linear models from the shrinking generator parameters is carried out by the following *Linearization Algorithm*:

Input: A shrinking generator characterized by two LFSRs, R_1 and R_2 , with their corresponding lengths, L_1 and L_2 , and the characteristic polynomial $P_2(x)$ of the register R_2 .

Step 1: From L_1 , L_2 and $P_2(x)$, compute the polynomial $P(x)$ as

$$P(x) = (x + \alpha^N)(x + \alpha^{2N}) \dots (x + \alpha^{2^{L_2}-1}N)$$

with $N = 2^0 + 2^1 + \dots + 2^{L_1-1}$.

Step 2: From $P(x)$, apply the Cattell and Muzio synthesis algorithm to determine two linear 90/150 CA, notated s_i , with such characteristic polynomial.

Step 3: For each s_i separately, proceed:

Step 3.1: Complement its least significant bit giving rise to S_i .

Step 3.2: Compute the mirror image of S_i , S_i^* , and concatenate both

$$S'_i = S_i * S_i^*.$$

Apply steps 3.1 and 3.2 to each S'_i recursively $L_1 - 1$ times.

Output: Two binary strings of length $L = L_2 \cdot 2^{L_1-1}$ codifying two CA corresponding to the given shrinking generator.

It can be noticed that the computation of both CA is proportional to L_1 concatenations. Consequently, the algorithm can be applied to shrinking generators in a range of practical application. In contrast to the nonlinearity of the shrinking generator, the CA-based models that generate the shrunken sequence are linear.

5 Conclusions

All the sequential solutions of linear constant coefficient difference equations can be realized by means of linear models based on cellular automata. Some of these solutions have a straight cryptographic application in stream ciphers. In this way, very popular cryptographic sequence generators conceived and designed as nonlinear generators can be linearized in terms of cellular automata. The linearization algorithm is simple as well as its implementation and can be applied to cryptographic generators in a range of practical application. The linearity of these cellular automata can be advantageously used in the analysis/cryptanalysis of such keystream generators.

Acknowledgements. This work has been supported by Ministerio de Educaci3n y Ciencia (Spain), Projects SEG2004-02418 and SEG2004-04352-C04-03.

References

1. Beth, T., Piper, F.: The Stop-and-Go Generator. In Proc. EUROCRYPT'84. LNCS, vol. 228, pp. 124-132. Springer, Heidelberg (1985)
2. Bluetooth, Specifications of the Bluetooth system, <http://www.bluetooth.com/>
3. Caballero-Gil, P., Fúter-Sabater, A.: A Wide Family of Nonlinear Filter Functions with a Large Linear Span. Information Sciences. 164, 197--207 (2004)
4. Cattell, K., Muzzio, J.C.: Synthesis of One-Dimensional Linear Cellular Automata. IEEE Trans. Computers-Aided Design, 15, 325--335 (1996)
5. Coppersmith, D., Krawczyk, H., Mansour, A.: The Shrinking Generator. In Proc. ECRYPTO'93. LNCS, vol. 773, pp. 22-39. Springer, Heidelberg (1994)
6. Fúter-Sabater, A.: Run Distribution in Nonlinear Binary Generators. Applied Mathematics Letters. 17, 1427--1432 (2004)
7. Gollmann, D., Chambers, W.G.: Clock-Controlled Shift Register. IEEE J. Selected Areas Commun. 7, 525--533 (1989)
8. Golomb, S.W.: Shift Register-Sequences. Aegean Park Press, Laguna Hill, (1982)
9. Gong, G.: Theory and Applications of q-ary Interleaved Sequences. IEEE Trans. Information Theory. 41, 400--411 (1995)
10. GSM, Global Systems for Mobile Communications, available at <http://cryptome.org/gsm-a512.htm>
11. Jennings, S.M.: Multiplexed Sequences: Some Properties. In Proc. EUROCRYPT'83. LNCS, vol. 149, Springer, Heidelberg (1983)
12. Kari, J.: A Survey of Cellular Automata. Theoretical Computer Science. 334, 3--39 (2005)
13. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and Their Applications. Cambridge University Press, Cambridge (1986)
14. Rivest, R.L.: RSA Data Security. Inc., March 12 (1998)
15. Wolfram, S.: Random Sequence generation by Cellular Automata. Advances in Applied Mathematics. 123 (7) (1986)

Automated Classification Tree Evolution Through Hybrid Metaheuristics

Miroslav Bursa and Lenka Lhotska

Czech Technical University in Prague,
Technicka 2, Prague 6, Czech Republic
{bursam, lhotska}@fel.cvut.cz

Abstract. In present, data processing is an important process in many organizations. Classification trees are used to assign a classification to unknown data and can be also used for data partitioning (data clustering). The classification tree must be able to cope with outliers and have acceptably simple structure. An important advantage is the white-box structure. This paper presents a novel method called ACO-DTree for classification tree generation and their evolution inspired by natural processes. It uses a hybrid metaheuristics combining evolutionary strategies and ant colony optimization. Proposed method benefits from the stochastic process and population approach, which allows the algorithm to evolve more efficiently than the methods alone. The paper also consults the parameter estimation for the method. Tests on real data (UCI and MIT-BIH database) have been performed and evaluated.

Keywords: Classification Tree, Ant Colony Optimization, Evolutionary Algorithm, Metaheuristics, Artificial Intelligence.

1 Introduction

Data mining, data analysis and data processing is in present the main task of many companies, organizations and research groups. Therefore computer-aided methods are applied, leading to the simplification of the processing of the data. The main goal of computer usage is data reduction preserving the statistical structure (clustering, feature selection), data analysis, classification, data evaluation and transformation.

This paper presents a novel method called ACO-DTree which uses a tree-like structure (similar to classification or decision tree structure) to partition provided data set and to assign a classification to each datum of the dataset.

In data mining process, exact or heuristic methods can be used. Exact methods provide better results, but require a lot of computational power. But sometimes more robust results may be needed – the exact methods can be over-learned. This means that the method provides very good results on the training data set, but performs poorly on the testing data set. That's when the approximation techniques can be used. They yield only approximate solution (with acceptable error) which may be more robust. The appropriateness of the use of approximate methods must be considered.

2 Method Description

This section is organized as follows: first, an introduction to data analysis for the purpose of this article is presented, following an introduction of what we mean by classification tree. Next we describe the proposed method together with evolutionary and ant colony algorithms.

2.1 Data Processing

Consider a set S of n -dimensional data where each datum can be described by exactly $n \in \mathbf{N}$ features $f_i \in \mathbf{R}$. Each datum can be described as $s_i = (f_0, f_1, \dots, f_n)$; $i \in \{0, 1, \dots, |S| - 1\}$, $s_i \in S$. Each datum can be classified into exactly one class $c_j \in C$, $j \in \{0, 1, \dots, |C| - 1\}$ of the set of available classes C .

This method requires *training* and *validation* set with classification known in advance (a priori). These sets are used during the learning process of the classifier and to validate the model (avoid over-learning). After successful evolutionary process of the classifier, the testing set can be used (where the classification is not needed a priori, but should be known in order to be able to evaluate the performance of the classifier on testing (unknown) data). If the classification is not known a priori, cluster validation techniques as described in [4] and [5] can be used to measure the classification (or clustering) efficiency of the clustering obtained.

2.2 Classification Tree

By classification tree we mean hereby a tree-like structure composed of similar nodes. Each node can have left and right subnode. Each node is represented by a decision rule with two parameters (feature index $j \in \langle 0; n \rangle$ and decision value $decisionValue \in Supp(f_j)$) which can be described in the following way for a datum $s_i \in S$:

```

t1   if (si.feature(j) < decisionValue)
t2     classifyToLeftBranch
t3   else
t4     classifyToRightBranch
    
```

The root node processes the data in exactly the same manner.

By node level in the tree we mean hereby the distance from the given node to the root node (see Fig. 1). Tree height is a maximum level in the tree. The tree structure is depicted in Fig. 1. Note that for the data sets clustered the constraints $\bigcup_i X_i = S$ and

$$X_i \cap X_j = \{ \} \text{ for each } i \neq j \text{ must be satisfied.}$$

Depending on the classification tree generated, the data are divided into subgroups which should have similar properties (we perform the minimization of intra-cluster

distance) and the classes should be different (we maximize the inter-cluster distance). This process is known as data clustering. The figure also shows the height of the tree with appropriate levels in the tree.

After partitioning the data set into clusters with described properties, the clusters can be assigned a class (if the class is known). Several approaches exist; the simplest is to assign the cluster a class which is represented by majority in the cluster. This approach is used in our method. If the classification is not known, classes can be assigned some unique number. Even in this case, the structure of the data can be revealed and important properties of the data can be discovered.

Each tree can be assigned a real number $e \in \mathbf{R}$ which can be called fitness function. Such number represents the classification efficiency of the tree. In ACO-DTree method this number is determined by the ratio of incorrectly classified data to the total data in the class (in this paper it is called *error ratio*). The goal of our method is to obtain tree with the lowest error ratio on the dataset. For method evaluation, the training data set is used. Validation dataset is used to verify the process of learning and to discover the over-learning of the tree. Finally, the testing dataset is used to evaluate the tree on the unknown data (data which have never been presented to the tree). If the classification of the testing data is not known, cluster validation techniques can be used (as described in section 2.1).

2.3 Ant Colony Optimization

Plenty of nature inspired methods are studied and developed in present. One category is represented by methods, which are inspired by the behavior of ant colonies. These methods have been applied to many problems (often NP-hard). Review can be seen in [1] and [2].

The Ant Colony Optimization [1] is a metaheuristic approach, inspired by the ability of ants to discover shortest path between nest and food source. The process is guided by laying down a chemical substance (pheromone). As the ants move, they deposit the pheromone on the ground (amount of the pheromone deposited is proportional to the quality of the food source discovered). Such pheromone is sensed by other ants and the amount of pheromone changes the decision behavior of the ant individual. The ant will *more likely* follow a path with more pheromone.

This way the ants communicate indirectly through their environment. Such autocatalytic communication is called stigmergy (communication through the environment) (introduced by Grassé).

The basic idea of ACO can be described as follows:

```
a1   Repeat until stopping criterion is reached
a2   Create ants
a3   Construct solutions
a4   Evaporate Pheromone
a5   Daemon Actions (pheromone deposit)
a6   End Repeat
```

New solutions are constructed continuously. It is a stochastic decision process based on the pheromone amount sensed by the ants. As in nature, the pheromone slowly evaporates over time (over iterations) in order to avoid getting stuck in local

minimum and to adapt to dynamically changing environment. Daemon actions represent ‘background’ actions which consist mainly of pheromone deposition. The amount is proportional to the quality of solution (and appropriate adaptive steps).

Main parameters of the algorithm are (with major importance to the method proposed): pheromone lay rate, pheromone evaporate rate, number of solutions created (number of ants), number of iterations, etc.

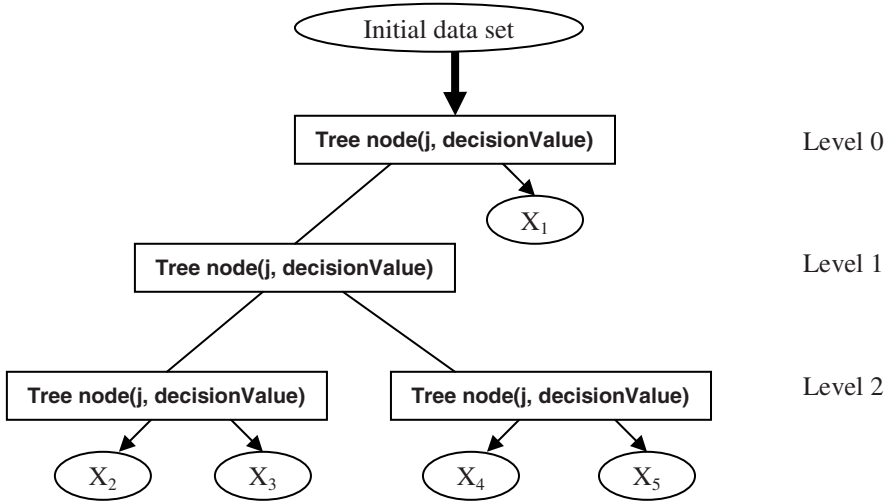


Fig. 1. Classification tree with height of 2 which classifies data into 5 classes (X_1 to X_5). The topmost node is called root node. Rectangles represent tree nodes with parameters (feature index and decision value), ellipses represent data sets. The largest ellipse represents input data set (to be classified) and smaller ellipses represent classes determined by the decision tree.

2.4 Evolutionary Computing

Evolutionary algorithms [9] are inspired by Darwinian evolution in the nature. A set of individuals are recombined and/or mutated during time and only the best subsample of the population is preserved. Many modifications exist, but the basic idea can be described as follows:

- e1 Generate initial population
- e2 Repeat until stopping criterion is reached
- e3 Mutate population
- e4 Recombine population
- e5 Select best individuals
- e6 End Repeat

At the start, a population of solutions is created. In each iteration the individual can be (with defined probability) mutated (changed) or recombined with another individual. After the process of recombination and mutation, the selection of best individuals is performed. The diversion of the population must be also preserved.

Main parameters of evolutionary algorithm are (with major importance to the method proposed): population size, probability of mutation and recombination, selection strategy, number of iterations, ...

2.5 ACO-DTree Method

With respect to the methods described in sections 2.3 and 2.4, the proposed method works as follows (for clarity, the number in parenthesis conforms to the respect line of code in either ACO method (ax) or EA method (ex)): first a population of random solutions is created (e1/a2). In each step the population is increased (e2/a3), improved (e3). Then the population is evaluated and only the specified number of solutions is conserved (e5). Then a pheromone is deposited in proportion to the quality of some best individuals (a5) and pheromone evaporation is performed (a4). The method can be described by the following code:

```

1   Generate initial population of solutions
2   Repeat until stopping criterion is reached
3     Create new solutions (pheromone driven)
4     Evolve population (mutation)
5     Evaluate population (determine classif. error)
6     Select best individuals
7     Pheromone evaporation
8     Daemon actions (pheromone laying, param. adapt.)
9   End Repeat

```

The pheromone matrix used in the method conforms to full graph (graph with all possible edges present) where nodes represent feature indexes. The edges contain pheromone and represent transition from one feature to another.

The tree is constructed as follows: First a root node with random index and value is created. Then, recursively for each possible sub-node (with certain probability), next feature is probabilistically chosen from the graph with preference increasing proportionally to pheromone contained on the edges. This process is iteratively repeated until the maximum level of the tree is reached. Only the best solutions (ants) are allowed to deposit pheromone into the matrix. The amount deposited is determined by the quality of the solution.

3 Experimental Part

Numerous tests have been performed to estimate the optimal parameters and measure the dependency of error ratio on relevant parameters. As a fitness function, the ratio of incorrectly classified data to the whole data set has been used ($e \in \langle 0;1 \rangle$). The minimization of this function is desired. First, the tests on artificial data set with Gaussian distribution have been performed. Then the iris data set [8] with 150 data items and 3 classes has been used to measure the performance and for preliminary parameter estimation. The main experimental part has been performed on the MIT-BIH database [7] (with 81 367 records and 2 classes with 10 features [6]).

3.1 Parameter Estimation

The most important parameters of the proposed method are: population size, number of new solutions added in each step, maximal number of iterations, max. tree height, pheromone lay/evaporate rate and the percent of ants which can deposit pheromone (elitist ratio). The overall results are better when the first four parameters increase, but the computational time rises. For other parameters, an optimum must be determined.

Based on the results of the preliminary experiments, population size and number of new solutions added has been fixed to 8 and 4 solutions respectively (in this experiment). These parameters actually increase/decrease the number of solutions generated over time. Similar effect can be obtained by setting maximum number of iterations. Elitist ratio (number of best solutions which can deposit pheromone has been also fixed the value of $\frac{1}{2}$ of the population (with minimum of 5).

The optimized parameters tuned are: maximum tree height and number of iterations. The error behavior when changing the parameters is shown in Fig. 2 and Fig. 3. The items on the graph represent an average of the 100 runs for each measurement. The average has been selected, because the probabilistic distribution of the result conforms to the Poisson distribution.

3.2 Parameter Adaptation

In order to avoid premature convergence and maintain diversity in the population of solutions, adaptive techniques have been used. First, the pheromone amount on the edge is limited and can vary in the range $\langle 0.05; 1.05 \rangle$, the evaporation rate and lay rate is adaptively changed to maintain an average pheromone value over the whole pheromone matrix stable (if the average pheromone drops by 10 %, the pheromone lay rate is increased, similar policy is applied to the pheromone evaporate rate; both values are bounded by the minimum and maximum value). This could lead to saturation of pheromone values in some cells, thus a countermeasure to maintain saturated edges on the minimum is also used. The adaptive balancing process diversifies the population and avoids getting stuck in local minima.

3.3 Results

As shown in Fig. 2 and Fig. 3, the ACO-DTree method outperforms the evolutionary algorithm where random selection of attributes has been performed. When the maximum number of iteration is limited, both the methods perform poorly, but when the limit value is increasing, the ACO-DTree method performs better on both validation and testing data set. The same situation arises when determining the maximum tree height. Note that this parameter depends on the nature of data, mainly on the number of classes and number of clusters contained in the data.

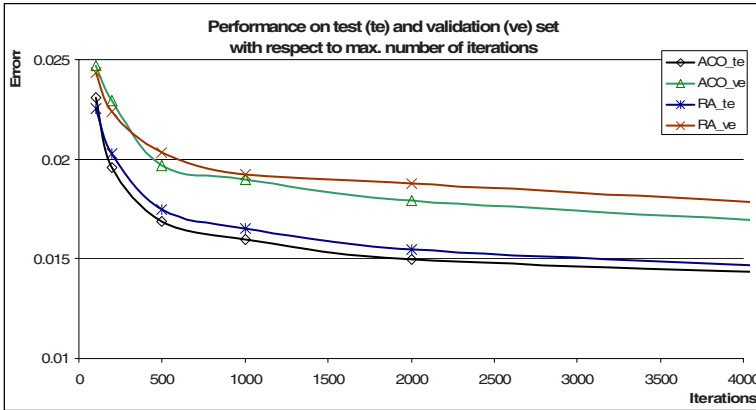


Fig. 2. Dependency of the error on the number of iterations over test set and validation set is presented on the graph. ACO stands for ACO-DTree method, RA stands for simple evolutionary algorithm. “_te” suffix stands for testing dataset error, suffix “_ve” stands for validation data set error. Each point measured represents an average over 100 runs.

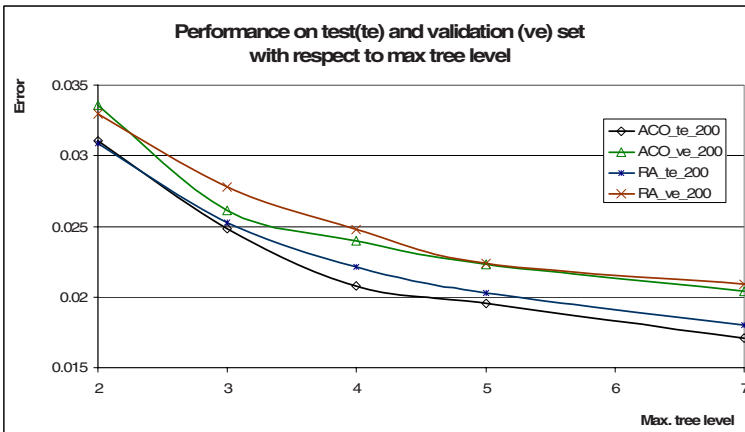


Fig. 3. Dependency of the error on the max. tree level over test set and validation set is presented on the graph. ACO stands for ACO-DTree method, RA stands for simple evolutionary algorithm. “_te” suffix stands for testing dataset error, suffix “_ve” stands for validation data set error. Each point measured represents an average over 100 runs.

4 Conclusion

In this paper we have presented a novel method called ACO-DTree which for data partitioning. The method assigns a classification to each datum presented and is based on the hybrid combination of evolutionary algorithm with ant colony optimization. This combination allows better convergence and increases robustness. The method has been compared with simple evolutionary algorithm, without the use of

pheromone. The ACO-DTree method outperformed the other method in all cases. The method has been (after preliminary tests) applied the MIT-BIH database with more than 80.000 records: The average accuracy of the method is sensitivity 93.22 % and specificity 87.13 %.

The number of maximum iterations strongly influences the run time, the value of 2000 seems as an acceptable compromise. The maximum tree depth doesn't influence so strongly, the value of 5 to 7 is reasonable. The advantage of classification tree obtained is: very fast application to newly incoming data and its white box structure (the classifier structure is known).

Acknowledgments. This research has been supported by the research program No. MSM 6840770012 "Transdisciplinary Research in the Area of Biomedical Engineering II" of the CTU in Prague, sponsored by the Ministry of Education, Youth and Sports of the Czech Republic. This work has been developed in the BioDat research group <http://bio.felk.cvut.cz>.

References

1. Dorigo, M., Stützle, T.: *Ant Colony Optimization*, MIT Press, Cambridge, MA, 2004
2. Blum, C.: Ant colony optimization: Introduction and recent trends, *Physics of Life Reviews*, Volume 2, Issue 4, 2005, 353-373
3. Stützle, T., Hoos, H.H.: MAX-MIN Ant system, *Future Gen. Comput. Syst.* 16, 2000, 8, 889-914
4. Bezdek, J., Li, W., Attikiouzel, Y., Windham, M.: A geometric approach to cluster validity for normal mixtures, *Soft Computing*, 1997, 1, 166-179
5. Davies, D.L., Bouldin, D.W.: A cluster separation measure, *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1979, 1 No. 2, 224-227
6. Chudacek, V., Lhotska, L.: Unsupervised creation of heart beats classes from long-term ECG monitoring, 18th Int. EURASIP Conf. Biosignals 2006, 18, 199-201
7. Goldberger, A.L., Amaral, L. A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals, *Circulation*, 2000 (June 13), 101, e215-e220
8. Newman, D., Hettich, S., Blake, C., Merz, C.: *UCI Repository of machine learning databases*, Univ. of California, Irvine, Dept. of Information and Computer Sciences, 1998
9. Beyer H.G., Brucherseifer E., Jakob W., Pohlheim H., Sendhoff B., Bing T.: Evolutionary algorithms – Terms and Definitions. <http://ls11-www.cs.uni-dortmund.de/people/beyer/EA-glossary/def-engl-html.html>

Machine Learning to Analyze Migration Parameters in Parallel Genetic Algorithms*

S. Muelas, J.M. Peñ, V. Robles, A. LaTorre, and P. de Miguel

CeSViMa, Computer Architecture Department, Universidad Politénica de Madrid
smuelas@cesvima.upm.es, {jmpena, vrobles, atorre, pmiguel}@fi.upm.es

Abstract. Parallel genetic algorithms (PGAs) are a powerful tool to deal with complex optimization problems. Nevertheless, the task of selecting its parameters accurately is an optimization problem by itself. Any additional help or hints to adjust the configuration parameters will lead both towards a more efficient PGA application and to a better comprehension on how these parameters affect optimization behavior and performance. This contribution offers an analysis on certain PGA parameters such as migration frequency, topology, connectivity and number of islands. The study has been carried out on an intensive set of experiments that collect PGA performance on several representative problems. The results have been analyzed using machine learning methods to identify behavioral patterns that are labeled as “good” PGA configurations. This study is a first step to generalize relevant patterns from the problems analyzed that identify better configurations in PGAs.

Keywords: parallel genetic algorithms, migration parameters, migration topologies, optimization problem hardness.

1 Introduction

Parallel genetic algorithms (PGAs) are powerful optimization tools for complex real-world problems. Besides the straightforward parallel implementation of genetic algorithms (master/slave models), PGAs have been a subject of research by themselves. Superlineal speed-ups reached by the most sophisticated PGA models (cellular and island models) have opened interesting questions about their particular behavior. Theoretical analyses [1] have been carried out to explain this behavior, but these studies are based on simplified models that are not useful when applied to complex optimization problems.

A genetic algorithm application requires many parameters to be configured; the number of these parameters is greater in the case of PGAs. This work presents a preliminary study on the relationship between these parameters and the algorithm performance. This study is not based on a theoretical point of view but on an analytical perspective of the results generated by many experimental runs.

The methodology proposed on this contribution is the following: (i) identify several PGA parameters to be analyzed, (ii) select a significant number of

* This work was supported by the Madrid Regional Education Ministry and the European Social Fund.

optimization problems, (iii) execute several times each experiment for each combination of the selected parameters, (iv) compare the results obtained on each optimization problem and label “good” and “bad” performing configurations, (v) analyze each problem data separately to identify efficient configuration patterns and (vi), try to combine the whole experimental data to extract general patterns common for all optimization problems.

The work performed on this study answers interesting questions but opens even more new questions to be analyzed in further detail.

The outline of this paper is as follows: section 2 presents an introduction of PGA taxonomy. In section 3, the experimental scenario is described in detail. Section 4, presents and comments the results obtained and lists the most relevant facts extracted from this analysis. Finally section 6 contains the concluding remarks obtained from this study.

2 Migration in Island-Model Parallel Genetic Algorithms

An important parameter on the performance of PGA (specifically in multipopulation models based on islands) is the migration strategy. This is configured through different parameters [1]:

1. Migration frequency: How often (in generations) are individuals sent?
2. Migration rate: How many individuals are migrated each time?
3. Immigrants’ selection: Which individuals are selected to migrate?
4. Replacement policy: How the combination among immigrants and the original population is made?
5. Migration topology: Where do individuals migrate to? Which island sends individuals to which other?

Upon studying these parameters [3] has proved that, even though subpopulations with a smaller size risk of falling into local optimum, an appropriate migration strategy can stop a suboptimum solution from dominating all populations. This appropriate strategy must be adjusted between the limits of a low interaction (which would practically imply the execution of N independent algorithms) and an excessive interaction (that would lead to the predominance of only one solution). A correct configuration can help to obtain better results with fewer evaluations [4].

2.1 Connection Topologies

For this study, several dynamic topologies have been selected. These topologies are a sample of all the possible connection schemas in PGA, but cover both static and dynamic approaches.

1. **Static Hypercube Topology:** Among the different alternatives for static configurations, the hypercube configuration has been selected. Hypercubes can be modified according to the desired dimensionality, allowing multiple connectivity (2, 3, ... n connections per node). Ring, double ring, mesh and other connection topologies are equivalent, at least in terms of structural connectivity, although hypercube keeps a minimal diameter.

2. **Dynamic Random Topology:** Dynamic topologies reconfigure their neighborhood relationships at each migration step. The simplest strategy to figure out the appropriate neighborhood is to select connection nodes based on randomness. This topology has been selected as a reference point for comparing other dynamic topologies.
3. **Dynamic Nearest Neighbor Topology:** Before the migration of individuals, each island determines the medoid of its population and broadcasts it to all other islands. In order to determine the neighbor relationships, each island uses the medoids and Gabriel relationship criterium. Gabriel neighborhood is proposed in [5] and in contrast with Delaunay neighbors with high-dimension problems, the neighbors can be computed efficiently. Each island then selects the nearest Gabriel neighbors until the topology degree is reached. If there are not enough Gabriel neighbors to reach the degree, the rest of the neighbors are completed with the nearest (not neighbors) islands.
4. **Dynamic Furthest Neighbor Topology:** This topology is similar to the previous one. Each island calculates its medoid and broadcasts it to the rest of the system so that each one can compute its Gabriel neighbors. The difference, in this case, is that the selected neighbors from the set of all Gabriel neighbors are the furthest ones. Furthermore, when not enough Gabriel neighbors are available, the completion of neighbors is based on the furthest not-neighbors ones.
5. **Hybrid Topology:** This topology is composed from a static hypercube topology and a dynamic nearest neighbor topology both with half of the hybrid degree. This implementation is used to analyze the beneficial effects of combining both static and dynamic topological approaches.

3 Experimental Scenario

Studies on the correct PGA configuration have been made by different authors. [6] analyzes the influence of the GA model (steady-state/generational), the PGA model (fine/coarse grain) and the number of processors on the performance of synchronous and asynchronous PGAs. However this study was carried out on only one problem.

[7] presents an interesting analysis of the migration frequency, considering also different migration sizes (number of individuals), but keeping the same topology and connectivity. The experiments have been repeated several times for statistical significance. The authors used specially created functions as well as standard test functions (Rosenbrock, Schwefel, Rastrigin, and Griewank).

3.1 Algorithm

The algorithm tested in this study follows the canonical PGA [1]. Based on [2], a set of relevant parameters have been selected in order to analyze their influence on the PGA performance. For these parameters, the following values have been used:

1. **Topology:** Static Hypercube, Dynamic Random, Dynamic Nearest Neighbor, Dynamic Furthest Neighbor, and Hybrid Topology.
2. **Number of islands:** 4, 8 and 16 (keeping the same overall population).
3. **Connectivity:** 2, 3 and 4.

4. **Migration frequency:** Migrations happen every 5 or 10 generations.

The rest of the parameters of the PGA have been fixed to the following values:

1. **Initialization:** Random uniform initialization of the population.
2. **Selection:** Tournament-based.
3. **Crossover probability:** 1.0 / **Mutation probability:** 0.01
4. **Full elitism:** The set of individuals of the next generation is composed upon the best ones among both parents and children of the current generation.
5. **Migration rate:** Top 10% / **Replacement policy:** Replace the worst individuals of the host population only if the immigrants have better scores.
6. **Convergence:** Fixed number of generations, estimated as 80% of the required generations in an average configuration combination.

3.2 Problems

Problems that represent different optimization difficulties in real-valued and binary representations have been selected, covering from simple to multimodal scenarios, and from epistatic and deceptive features.

In order to perform all the tests in the same conditions, the fixed values shown on table 1 have been used. The objective of this study is not to optimize table 1 parameters, which have been given a reasonable value, but to consider only the influence of the algorithm parameters listed in section 3.1.

Table 1. Dimension, overall population size and problem-specific parameters

Problem	Dimension	Generations	Pop. Size	Cross. Op.	Mut. Op.
Griewank	100	400	512	Blend	Gaussian
Rastrigin	100	400	512	Blend	Gaussian
Sphere	100	400	512	Blend	Gaussian
Deceptive	15	400	512	Blend	Gaussian
Two-Peaks	25	400	512	Blend	Gaussian
Royal-Road	209	400	512	Two Points	Bit Flip
Maxbit	700	400	512	Two Points	Bit Flip
TSP	48	400	512	Blend	Gaussian

3.3 Experiments Procedure

For each of the proposed problems, the following experimental procedure is performed: (i) all the combinations of the input parameters are considered, (ii) for each combination, 50 independent executions have been run, storing the fitness obtained when the maximum number of iterations is reached, (iii) the fitness results obtained for each of the combinations is pairwise compared, using a Wilcoxon non parametric t-test with $p < 0.001$. If one combination of parameters happens to be better than another (according to the t-test) the winning combination is granted with $+1$ wins and the losing combination penalized with -1 wins. As all the combinations are compared against each other, they are ranked (depending on how many other combinations are better/worse).

Once all the combination of parameters are ranked, they are labeled as “*high performance configurations*” if $wins \geq 10$, and as “*low performance configurations*”

if $wins \leq -10$. These values have been selected to roughly divide the configurations into three well balanced intervals.

4 Analyzing the Results

Once experimental results have been labeled, each individual problem has been analyzed using a C5.0 rule-induction algorithm (ten fold cross-validation) [12]. The results given on table 2 show the rules that classify high and low performance configurations, including rule support and accuracy. Most of the rules are based on migration frequency and the number of islands (it should be mentioned that as the global population is fixed, the higher the number of islands is, the smaller the population per island becomes). Topology and connectivity parameters are not considered in most of the rules and they only appear in low support rules. Another relevant aspect comes from the existence of rules that identify either HIGH or LOW performance configurations depending on the problem (e.g., Royal Road and TSP High performance rule #1 vs. Griewank low performance rule #2).

Table 2. Induced rules from individual problems (support, accuracy)

Problems	High Performance	Low Performance
Griewank	1. #islands=4 (27%,1.0) 2. frequency=5 and #islands<=8 (32%,0.842)	1. #islands=16 (37%,0.818) 2. frequency=10 and #islands>4 (37%,0.72)
Rastrigin	1. frequency=5 and connectivity>2 (32%,1.0) 2. frequency=5 and #islands<8 (32%, 0.947)	1.frequency=10 (50%,1.0)
Sphere	1. frequency=5 (50%,0.80)	1. #islands=16 and frequency=10 (18%,1.0) 2. connectivity=2 and frequency=10 (18%, 1.0) 3. #islands>4 and frequency=10 and connectivity<=3 (23%, 0.929)
Deceptive	1. #islands=16 (37%,1.0)	1.#islands=4 (27%,0.875) 2.#islands<=8 and topology=random (17%, 0.8)
Two-peaks	1. frequency=5 (50%, 0.73)	1.frequency=10 (50%, 0.767)
Royal Road	1. frequency=10 and #islands>4 (37%,0.72)	1. frequency=5 and #islands<=8 (32%,0.73)
Maxbit	1. frequency=10 and #islands<=8 (32%,1.0) 2. #islands=4 and connectivity=2 (17%, 0.9)	1. frequency=5 and connectivity>2 (32%,1.0) 2. #islands>8 (37%, 0.818)
TSP	1. frequency=10 and #islands>4 (37%,0.636)	1. #islands=4 (27%,1.0) 2. frequency=5 (50%,0.667)

4.1 Problem Characterization

PGAs show different behavior patterns on each different problem. Table 2 results are a clear example of how difficult it is to obtain general rules for the problem characterization.

Our new proposal is the following: with measures traditionally used to evaluate genetic algorithm hardness, it is possible to label problems (with their parameters) with valuable extra information. Thus, the analysis performed before can be extended.

Studies carried out by [8] on the difficulty of different optimization problems, measured the complexity of the problem as the correlation between fitness values and the distance in the solutions space (fitness-distance correlation, or FDC). For real-value problems, Manhattan distance has been used and for binary encodings hamming has been selected.

FDC requires information about the global optimum and its value. [9] has proved that the correlation metric between distance and fitness can also be applied on the individuals of a population without knowing the global optimum. This approach uses the best fitness value in the population as a reference. It is called local FDC (or LFDC). This approximation has been considered as an evaluation tool for the initialization processes quality. Davidor's Epistasis Variance and Normalized Epistasis have been also computed as defined by [10]. One interesting example is the study presented in [10], as it uses the FDC and epistasis variance for the purpose of generating problem equivalence classes.

Recently, [11] compares predictive and posterior measures of problem difficulty. Results show that there are cases in which predictive measures are limited in their accuracy, but, nevertheless, they are a helpful tool to guess problem hardness. Here, we propose a new difficulty measure named *Fitness Increment*. In Fitness Increment, the first step is to scale the fitness of a given population from 0 to 1. Then, the difference between the fitness of the best children and the fitness of the best parent is calculated. The fitness of the best children is also scaled according to the same mechanism used with its parents (this could yield scaled fitness values greater than 1 if the offspring is better than the overall best value of its parents' generation). This measure does not consider landscape difficulties, directly. It takes into account selection, crossover and mutation operators as well as their respective parameters.

4.2 General Discussion

In order to include difficult measures to characterize each of the problems, a random uniform population of 3000 individuals has been created. With all these individuals, FDC, LFDC and Fitness Increment have been computed. Since this work deals with both maximization and minimization problems, the sign of FDC and LFDC has been adjusted so that +1 and -1 correspond to maximum and minimum correlation respectively.

A correlation analysis between the five metrics has been performed, resulting in a higher correlation rate among FDC, LFDC and Fitness Increment that validates Kallel's hypothesis [9] (FDC and LFDC). There exists also a higher correlation result between FDC and Fitness Increment. This is considerably interesting because FDC measures fitness landscape properties, while Fitness Increment deals with the combination of the selection, crossover and mutation operators. Epistasis-related metrics are highly correlated between them but uncorrelated to the rest of the measures; this means that these measures represent different problem characteristics.

Table 3. Difficulty measures obtained for a random sample of 3000 individuals

	Epistasis Variance	Normalized Epistasis	FDC	LFDC	Fitness Increment
Royal road	0.0145	0.385	-0.133	0.006	-0.00376
Maxbit	0.0000903915	0.01	1	0.18	0.00469
TSP	0.0127	0.277	0.002	-0.009	-0.027
Griewank	0.00687	0.29	0.96	0.46	0.15
Sphere	0.006554	0.2795	1.0	0.44	0.148
Rastrigin	0.0080604	0.24	0.72	0.32	0.08
Deceptive	0.01	0.23	-0.42	-0.01	-0.0042
Two-Peaks	0.0105	0.205	0.19	0.21	0.007

All the 60 parameters combinations taken from each problem have been appended to a single table, including the columns that represent the 5 difficulty measures (FDC, LFDC, Fitness Increment, epistasis variance and normalized epistasis). This new data table has also been analyzed using C5.0 (10-fold cross-validation). The following induced rules were extracted:

Table 4. General rules with support and accuracy (Sup./Acc.)

Conditions	Perf.	Sup./Acc.
Fitness Increment > 0.00468 and frequency = 5 and #islands ≤ 8	High	16%/0.934
Fitness Increment in (0.00468, 0.148] and frequency = 5 and connectivity > 2	High	12%/0.877
Fitness Increment ≤ 0.00468 and frequency = 10	High	25%/0.55
Fitness Increment in (0.00468, 0.148] and frequency = 10	Low	19%/0.844
Fitness Increment > 0.00468 and frequency = 10 and #islands > 4	Low	18%/0.818
Fitness Increment ≤ 0.00468 and frequency = 5	Low	25%/0.542

Fitness Increment is selected as the best measure to characterize the problem. According to this measure and from the point of view of the used operators, Royal Road, TSP and Deceptive are considered difficult problems (Fitness Increment ≤ 0.00468), and the rest of problems are easy problems. For easy problems, good configurations should have frequent migration rates and either not too many islands or good connectivity. Difficult problems, instead, require less frequent migrations (although, this rule is not very accurate).

The presence of the number of islands in the extracted rules should be interpreted under two possible perspectives: the obvious number of different populations (dwelling the islands), and the population size in each of them. The experiments performed in this study have always used a fixed population size divided among the islands. In this sense, fewer islands also means more individuals for each of them.

It is remarkable the fact that topologies have minimal influence in the algorithm performance. This validates the results presented by Cantú-Paz [1], in which it is mentioned that topology could be generalized as the connectivity and diameter of the proposed island connections. Connectivity has only a marginal influence in the performance, compared with the number of islands and the migration frequency.

The general results presented by [7] also indicate that moderate migration intervals with a small migration size are better than large number of individuals migrated after

more generations. The present work shows that this rule is not valid in general, and would only be effective with simple problems. Note that Griewank and Rastrigin problems, present in both [7] and our study, have been considered “easy” by the Fitness Increment measure and the intervals induced by the machine learning algorithm.

5 Conclusions

The conclusions derived from this study have shown relevant patterns that identify better configurations in PGAs. Although this early results are quite promising, further analysis should be carried out including more parameters (migration rate, overall population size, more topologies and selection and replacement policies). Additionally, more values from the already considered parameters could be included. The selected problems sample could also be extended with more real world and synthetic problems. The results achieved by this study have required 24000 executions and were run on the Magerit system belonging to the CeSViMa Supercomputing Center.

References

1. CantúPaz, E. (1999). Designing Efficient and Accurate Parallel Genetic Algorithms. PhD thesis. University of Illinois at Urbana-Champaign.
2. M. Nowostawski and R. Poli. Parallel genetic algorithm taxonomy. In L. C. Jain, editor, Proc. KES'99. 88-92, 1999
3. Chrisila C. Petty and Michael R. Leuze. A theoretical investigation of a parallel genetic algorithm. In Proc. 3rd Int. Conf. on Genetic algorithms, pages 398-405, 1989
4. Whitley, D., Rana, S., Heckendorn, R.B.: The island model genetic algorithm: On separability, population size and convergence. Journal of Computing and Information Technology 7 33-47. 1999
5. Jaromczyk, J.W. and Toussaint, G.T: Relative Neighborhood Graphs And Their Relatives. Proc. IEEE Vol 80, pages 1502-1517, 1992
6. E. Alba, J.M. Troya, An Analysis of Synchronous and Asynchronous Parallel Distributed Genetic Algorithms with Structured and Panmictic Islands, LNCS 1586, pp. 248-256., 1999
7. Zbigniew Skolicki and Kenneth De Jong. The influence of migration sizes and intervals on island models. In GECCO '05. 1295–1302, 2005.
8. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. Proc. 6th Int. Conf. on Genetic Algorithms, pages 184–192 1995.
9. Leila Kallel, Marc Schoenauer. Alternative Random Initialization in Genetic Algorithms. ICGA 1997: 268-275
10. Bart Naudts, Leila Kallel. Comparison of Summary Statistics of Fitness Landscapes. IEEE Trans. Evol. Comp. V.4.1:1--15, 2000
11. J. He, C. Reeves, and X. Yao. A Discussion on posterior and prior measures of problem difficulties. In PPSN IX Workshop on Evolutionary Algorithms - 2006.
12. R. Quinlan. C5.0: An Informal Tutorial.1998. <http://www.rulequest.com/see5-unix.html>

Collaborative Evolutionary Swarm Optimization with a Gauss Chaotic Sequence Generator

Rodica Ioana Lung and D. Dumitrescu

Babeş-Bolyai University of Cluj Napoca
Department of Computer Science
{srodica, ddumitr}@cs.ubbcluj.ro

Abstract. A new hybrid approach to optimization in dynamical environments called Collaborative Evolutionary-Swarm Optimization (CESO) is presented. CESO tracks moving optima in a dynamical environment by combining the search abilities of an evolutionary algorithm for multimodal optimization and a particle swarm optimization algorithm. A collaborative mechanism between the two methods is proposed by which the diversity provided by the multimodal technique is transmitted to the particle swarm in order to prevent its premature convergence. The effect of changing the random number generator used for selection and for variation operators within CESO with a chaotic sequence generator is tested.

Keywords: crowding, differential evolution, particle swarm optimization, gauss map.

1 Introduction

One of the challenges presented by real-world applications is their dynamical character. The problem of detecting and tracking moving optima in a dynamical environment has been successfully addressed by evolutionary algorithms during the last years [4].

A new hybrid approach to solving optimization problems in dynamical environments called Collaborative Evolutionary - Swarm Optimization (CESO) is presented.

CESO is based on the collaboration between two optimization methods: an evolutionary algorithm for multimodal optimization and a particle swarm optimization algorithm. The evolutionary multimodal optimization algorithm provides a diversity preservation mechanism preventing the particle swarm's premature convergence to local optima.

The effect of changing the random number generator used by the selection and the variation operators within CESO with a chaotic sequence generator is studied.

2 Collaborative Evolutionary – Swarm Optimization

CESO algorithm is a simple method for detecting and tracking moving optima in a changing environment by using two populations of equal size. One of the population is responsible for preserving diversity of the search and the other one tracks the global optimum. CESO proposes a collaborative mechanism between the two populations in order to avoid premature convergence and to efficiently track moving optima.

2.1 CESO Populations

The rules by which the two populations used by CESO are evolved are described in what follows.

The CRDE Population

The first population called the CRDE population is evolved by an evolutionary multimodal optimization algorithm in order to maintain a good population diversity.

As a multimodal search operator CESO uses the Crowding Differential Evolution (Crowding DE) [10] algorithm, a very efficient method for detecting multiple optima in static environments.

Crowding Differential Evolution extends the Differential Evolution (DE) algorithm [8] with a crowding scheme. The only modification to the conventional DE is made regarding the individual (parent) being replaced. Usually, the parent producing the offspring is substituted, whereas in CrowdingDE the offspring replaces the most similar individual among the population (if it is fitter).

A *DE/rand/1/exp* [9] scheme is used. The best individual in the CRDE population is denoted *cbest*.

The SWARM

The second population used by CESO, called SWARM, is a particle swarm updated by using classical particle swarm optimization (PSO) rules [5].

Particle swarm optimization (PSO) is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling.

Within PSO each individual has a velocity vector associated. Each iteration, the following equations are used to compute the new position of individual $x=(x_1, \dots, x_n)$:

$$v_{i+1} = v_i + c_1 * \text{rand} * (pbest_i - x_i) + c_2 * \text{rand} * (gbest_i - x_i) \quad (1)$$

$$x_{i+1} = x_i + v_i \quad (2)$$

where $v=(v_1, \dots, v_n)$ is called the velocity of particle x ; *pbest* represents the best position of individual x so far, *gbest* represents the best individual in the whole population detected so far; *rand* is a random number between (0,1) and c_1, c_2 are learning factors. Usually $c_1 = c_2 = 2$. Two parameters, *vmin* and *vmax* are used to limit the velocity.

Endowing PSO with an efficient diversity preserving mechanism it becomes a very powerful optimization technique.

2.2 The Collaboration

One of the main problem in dealing with dynamical environments is the premature convergence to local optima. To cope with it CESO uses the CRDE population to maintain a set of local and global optima during the entire search process. The SWARM population is used to detect the global optimum and to indicate - if necessary - its position to the CRDE population.

Both CRDE and SWARM populations evolve in their 'natural' manner, i.e. no additional mechanism is added to them individually.

The collaborative mechanism proposed by CESO implies two-ways communication between the SWARM and CRDE:

Transmitting Information from the CRDE to the SWARM Population

The CRDE population maintains a good diversity over the search space by maintaining a set of local optimal solutions. CRDE information is transmitted to the SWARM by copying all individuals from the CRDE to the SWARM. Thus the SWARM is actually reinitialized. The reinitialization of the SWARM takes place if one of the followings occur:

- a change is detected in the environment (the test is made by re-evaluating cbest); in this case all individuals are evaluated;
- the distance between cbest and gbest is lower than a prescribed threshold θ (for example 0.1)

Transmitting Information from the SWARM to CRDE Population

At each iteration gbest replaces cbest if it has a better fitness value. Thus the CRDE population contains the best optima detected at each iteration.

3 CESO Algorithm

Within CESO both populations are randomly initialized and evaluated. At the beginning of each iteration a test is performed to check if a change in the environment has occurred during the last iteration. The best individual in the CRDE population is re-evaluated: if a difference appears between the new and the old fitness value it is considered that a change took place.

In case a change appears in the environment or if the distance between cbest and gbest is very small, the search of the SWARM is restarted by copying the CRDE individuals to it. By re-starting the search with particles scattered over the search space, the SWARM presents a good potential to locate the global optimum.

At the end of each iteration gbest replaces cbest if it is better than cbest. Therefore, at each iteration, the CRDE contains the best individual found so far.

CESO technique is outlined in what follows:

```

Parameters setting;
Randomly initialize CRDE and SWARM;
Evaluate populations;
While ( final condition not met)
    If (change in landscape)
        Copy CRDE to SWARM;
        Evaluate populations;
    EndIf
    If (distance between cbest and gbest less than 1)
        Copy CRDE to SWARM;
    EndIf
    Update SWARM;
    Evolve CRDE;
    Evaluate populations;
    If (gbest better than cbest)
        gbest replaces cbest in CRDE;
    EndIf
EndWhile

```

4 Chaotic Sequence Generators

In a study performed by Caponetto et al. [3] the effect of the use of a chaotic sequence generator instead of a random number generator within EAs is studied. By means of numerical experiments it is shown that replacing the random number generator with a chaotic sequence generator during the selection and variation processes of an EA leads to better performance and faster convergence. It is also shown that EAs can be very sensible to different

Particularly the Gauss Map has proven to enhance the performance of EAs.

The Gauss Map is described by the following equations:

$$x_{k+1}=G(x_k) \quad (3)$$

and

$$G(x) = \begin{cases} 0, & x = 0 \\ \frac{1}{x}, & x \in (0,1) \end{cases} \quad (4)$$

The effect of the use of the Gauss Map generator within CESO is studied by means of numerical experiments.

5 Numerical Experiments

Numerical experiments concerning the Moving peaks benchmark (MPB), scenario2, as proposed by Branke in [2] were performed. This scenario has also been used by several authors and allows the comparison of results obtained by different methods. The settings for this scenario are presented in Table 1.

Table 1. Standard settings for the Moving Peaks Problem

Parameter	Setting
Number of peaks p	10
Number of dimensions d	5
Peak heights	[30,70]
Peak widths	[1,12]
No. of evals. between changes	5000
Change severity s	1.0
Correlation coefficient λ	0

Results obtained by CESO have been compared [7] compared with those reported by the following methods:

- the Self Organizing Scouts (SOS) [2];
- the Multiswarms (MPSO) methods [1];
- The Particle Swarm with Speciation and Adaptation (SPSO) [6].

The best results obtained using the three methods considered, where applicable, were compared with those obtained by CESO.

MPSO and SPSO may have several configurations and variants. The best reported results have been chosen from the various configurations in order to compare them with those obtained by CESO.

Results are averaged over 50 runs with different random seed generator for CESO.

In all cases the results CESO outperformed the other methods in terms of offline error.

In the following, the performance of CESO using the standard random generator provided by the implementation (Delphi 7.0) is compared with the results obtained by using the chaotic sequence generator Gauss Map [3].

Parameter Settings for CESO

CESO uses only three parameters: the population size and vmin/vmax to control the particles velocities. For the other parameters specific to Crowding DE and PSO the usual values are used.

Parameters setting for CESO are presented in Table 2. Numerical results indicate that using small CRDE and SWARM population size, CESO results are significantly better than those obtained by other methods as far as the average offline error is concerned.

Table 2. Parameter Settings for CESO

Parameter	Setting
CRDE and SWARM sizes	10
vmin,vmax	-0.1,0.1

In order to emphasize the effect of the hybridization Table 3 presents results obtained when the CRDE population and the SWARM evolve independently. It can be noticed that results obtained by CESO are better than the other ones, indicating that the collaboration mechanism proposed is efficient.

Table 3. Offline error and standard error for CESO and for the Crowding DE and PSO without any collaboration

Method	Value
CESO	1.38±0.02
Crowding DE	3.98±0.14
PSO	24.23±1.30

Chaotic Sequence Generators

The effect of changing the random number generator of CESO with the Gauss Map time series number generator is presented. Other sequence generators whose effect has been studied are: the logistic map, the tent map and the sinusoidal iterator. Because the use of these generators lead to results far worst in terms of offline performance they are not reported here.

Results reported by CESO show that among the parameters of the MPB problem the one to which CESO (and other methods) is most sensible is the shift severity s.

That is why the performance of CESO using the Gauss Map (G-CESO) is presented against results previously reported. G-CESO was run 50 times for each experiment setting starting with a different seed. Table 5 minimum and maximum values obtained for the two methods. Table 4 presents average and standard error for the offline performance as well as results for the Wilcoxon non-parametric rank-sum test performed to determine weather the difference between results can be considered significant. P-values less than 0.05 indicate that results are significantly different.

Table 4. Minimum and Maximum values for the offline performance

Shift severity s	CESO		G-CESO	
	Min	Max	Min	Max
0	0.58	1.29	0.57	1.85
1	1.13	1.75	1.19	1.68
2	1.50	2.46	1.50	2.35
3	1.68	2.90	1.55	3.21
4	1.72	3.23	1.69	3.47
5	1.72	3.64	1.61	4.40

Table 5. Average offline error and standard error for different shift severity values

Shift severity s	CESO	G-CESO	p-value
0	0.85±0.02	0.98±0.03	0.006
1	1.38±0.02	1.36±0.01	0.87
2	1.78±0.02	1.80±0.15	0.61
3	2.03±0.03	2.19±0.24	0.01
4	2.23±0.05	2.52±0.27	0.0002
5	2.52±0.06	3.09±0.33	4.5e-6

The performance of G-CESO is similar to that of CESO and is better in terms of offline performance than that of other methods in literature. For the standard setting $s=1$ the best reported result before CESO was 1.75 (MPSO) while G-CESO reports an average of 1.36. Still it cannot be stated that using the Gauss Map the performance of CESO is significantly improved. The use of the Gauss Map leads to results presenting higher standard error between different runs. Thus minimum values for G-CESO are lower than those provided by CESO in five out of six cases. This indicates that when using the Gauss Map the seed for the sequence generator is very important and influences the results. To test this theory, 100 runs were performed for the standard scenario of the MPB using different uniformly generated seeds for the Gauss Map generator. The average offline error obtained was 1.36 and corresponding standard error was 0.008. The minimum value was 1.20 and maximum 1.73 – not very different from the values obtained after 50 runs. Standard error indicates that the results for this setting do not depend so much on the seed of the generator. The same test was performed $s=5$ the setting presenting the highest standard error and the

results were as follows: 2.90 ± 0.10 for the offline error, minimum value of 1.63 and maximum value of 4.81.

The effect of the use of the Gauss Map generator only on the CRDE population and on the SWARM respectively for the standard scenario of the MPB is presented in Table 6. These results indicate that the best scenario would be to use the Gauss Map generator only for updating the SWARM.

Table 6. Results obtained when using the Gauss Map only for one of the CESO populations

Method	Min	Max	Avg. offline error	Standard error
G-CRDE+SWARM	1.14	1.59	1.36	0.01
CRDE + G-SWARM	1.13	1.56	1.34	0.01
G-CESO	1.19	1.68	1.36	0.01

6 Conclusions

CESO is a new hybrid method for tracking optima in dynamical environment. CESO algorithm combines the abilities of the Crowding Differential Evolution algorithm for multimodal optimization and of the Particle Swarm Optimization by using a collaboration mechanism in order to detect and track optima in a changing environment.

Two populations of individuals are evolved by CESO. The CRDE population uses the Crowding Differential Evolution algorithm to detect and to maintain a set of approximation of local optima while the SWARM population follows the rules of Particle Swarm Optimization to track the global optima.

The collaboration mechanism is applied whenever a change is detected in the search space, or if the best individual in the SWARM is too close to the best in CRDE the search of the SWARM is restarted by re-initializing the SWARM with the positions of individuals in the CRDE population. Due to the diversity offered by these individuals, the SWARM population is capable to locate the global optimum. If the SWARM locates an optimum better than the best individual in the CRDE population then replaces *cbest* in order to enhance the search of the CRDE.

The influence of using a Gauss Map chaotic sequence generator for the selection and variation part of CESO is studied. It is shown that results obtained by the adapted G-CESO differ from those of CESO and that G-CESO may be considered sensible to the starting values for the Gauss Map series. The separate effect on the populations used by CESO suggests however that best results are obtained when only the SWARM uses the Gauss Map. A further study of the influence of starting values is necessary and represents the object of current research.

References

1. T. Blackwell and J. Branke, "Multiswarms, exclusion, and anticonvergence in dynamic environments," *Evolutionary Computation*, IEEE Transactions on, vol. 10, issue: 4, 2006.
2. J. Branke, *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, 2001.

3. Caponetto, R.; Fortuna, L.; Fazzino, S.; Xibilia, M.G. Chaotic sequences to improve the performance of evolutionary algorithms, *Evolutionary Computation, IEEE Transactions on*, Vol.7, Iss.3, June 2003 Pages: 289- 304
4. Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey." *IEEE Trans. Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
5. J. Kennedy and R. C. Eberhart, *Swarm intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.
6. X. Li, J. Branke, and T. Blackwell, "Particle swarm with speciation and adaptation in a dynamic environment," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM Press, 2006, pp. 51–58.
7. R. I. Lung, D. Dumitrescu. A collaborative model for tracking optima in dynamic environments. In *IEEE Congress on Evolutionary Computation*, (accepted paper), 2007.
8. R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," Berkeley, CA, Tech. Rep. TR-95-012, 1995.
9. R. Storn and K. Price, "Differential evolution a simple evolution strategy for fast optimization." *Dr. Dobbs's Journal of Software Tools*, vol. 22, no. 4, pp. 18–24, 1997.
10. R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*. Portland, Oregon: IEEE Press, 20-23 June 2004, pp. 1382–1389.
11. S. Tsutsui, Y. Fujimoto, and A. Gosh, "Forking genetic algorithms: GAs with search space division," *Evolutionary computation*, vol. 5, pp. 61–80, 1997.
12. R. K. Ursem, "Multinational GAs: Multimodal optimization techniques in dynamic environments," in *Proceedings of the Second Genetic and Evolutionary Computation Conference (GECCO-2000)*, vol. 1. Riviera Hotel, Las Vegas, USA: Morgan Kauffmann Publishers, 2000, pp. 19–26.

A New PSO Algorithm with Crossover Operator for Global Optimization Problems

Millie Pant¹, Radha Thangaraj¹, and Ajith Abraham²

¹ Department of Paper Technology, IIT Roorkee, India
millifpt@iitr.ernet.in, t.radha@ieee.org

² Center of Excellency for Quantifiable Quality of Service,
Norwegian University of Science and Technology, Norway
ajith.abraham@ieee.org

Abstract. This paper presents a new variant of Particle Swarm Optimization algorithm named QPSO for solving global optimization problems. QPSO is an integrated algorithm making use of a newly defined, multiparent, quadratic crossover operator in the Basic Particle Swarm Optimization (BPSO) algorithm. The comparisons of numerical results show that QPSO outperforms BPSO algorithm in all the twelve cases taken in this study.

Keywords: Particle Swarm Optimization, Crossover, Global Optimization, Quadratic Interpolation.

1 Introduction

Evolutionary Algorithms (EA) and Particle Swarm Optimization (PSO) are perhaps the two most common stochastic techniques for solving global optimization problems. Both the techniques have proved their mettle in solving the complex optimization problems (test as well as real life problems). Some of the similarities between the EA and PSO as pointed out by Angeline [1] may be given as:

- Both are population based search techniques.
- Neither requires the auxiliary knowledge of the problem.
- In both the algorithms solutions belonging to the same population interact with each other during the search process.
- The quality of the solutions are improved using techniques inspired from real world phenomenon On one hand, EA are inspired from the metaphor of natural biological evolution using the concepts of selection, mutation and reproduction on the other hand PSO uses the complex social cooperative and competitive behavior exhibited by different species like birds, bees humans etc.

Both the algorithms have their share of weaknesses and strengths and it is quite natural to think that the fusion of the two will result in an algorithm which has properties of both the algorithms. A number of techniques suggesting a combo of these two methods are available in literature. Out of the EA operators' viz. selection, crossover and mutation, the one that has been used most frequently is the mutation

operator. Some of the commonly used mutation operators used in PSO algorithms are Gaussian, Cauchy, linear, nonlinear operators etc. For a detailed description, the reader is suggested [2] - [7]. However, for selection and reproduction operator only a few examples are available (see for instance Angelina [8], Clerc [9]). The objective of this paper is to see the performance of a BPSO after including a crossover operator in it. For this purpose we developed a new crossover operator called quadratic crossover operator. It makes use of three swarm particles to produce a new particle that lies on the point of minima of the quadratic curve (hence the name quadratic crossover) passing through the three chosen particles.

The structure of the paper is as follows: in section 2, we have briefly explained the Basic Particle Swarm Optimization, in section 3; we have defined the proposed QPSO algorithm. Section 4 deals with experimental settings. Section 5 gives numerical results and finally the paper concludes with section 6.

2 Basic Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a relatively newer addition to a class of population based search technique for solving numerical optimization problems. The particles or members of the swarm fly through a multidimensional search space looking for a potential solution. Each particle adjusts its position in the search space from time to time according to the flying experience of its own and of its neighbors (or colleagues).

For a D-dimensional search space the position of the i th particle is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$. Each particle maintains a memory of its previous best position $P_{besti} = (p_{i1}, p_{i2}, \dots, p_{iD})$. The best one among all the particles in the population is represented as $P_{gbest} = (p_{g1}, p_{g2}, \dots, p_{gD})$. The velocity of each particle is represented as $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. In each iteration, the P vector of the particle with best fitness in the local neighborhood, designated g , and the P vector of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity. The two basic equations which govern the working of PSO are that of velocity vector and position vector given by:

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

The first part of equation (1) represents the inertia of the previous velocity, the second part is the cognition part and it tells us about the personal thinking of the particle, the third part represents the cooperation among particles and is therefore named as the social component [10]. Acceleration constants c_1, c_2 [11] and inertia weight w [12] are the predefined by the user and r_1, r_2 are the uniformly generated random numbers in the range of [0, 1].

3 Proposed QPSO Algorithm

The proposed QPSO algorithm is a simple and modified integrated version of BPSO and EA. The quadratic crossover operator suggested in this paper is a nonlinear multi

parent crossover operator which makes use of three particles (parents) of the swarm to produce a particle (offspring) which lies at the point of minima of the quadratic curve passing through the three selected particles. The new particle is accepted in the swarm irrespective of the fact whether it better or worse than the worst particle present in the swarm. In this way the search is not limited to the region around the current best location but is in fact more diversified in nature.

The quadratic interpolation [13] uses $a = X_{\min}$, the particle having minimum function value and two other randomly selected particles $\{b, c\}$ from the swarm to determine the coordinates of the new particle $\tilde{x}^i = (\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^n)$, where

$$\tilde{x}^i = \frac{1}{2} \frac{(b^{i^2} - c^{i^2}) * f(a) + (c^{i^2} - a^{i^2}) * f(b) + (a^{i^2} - b^{i^2}) * f(c)}{(b^i - c^i) * f(a) + (c^i - a^i) * f(b) + (a^i - b^i) * f(c)} \quad (3)$$

The nonlinear nature of the quadratic crossover operator used in this work helps in finding a better solution in the search space. The computational steps of the proposed QPSO are given in Fig. 1.

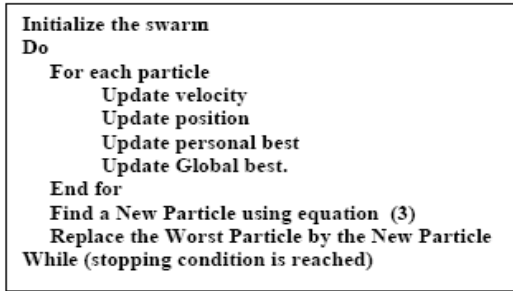


Fig. 1. Flow QPSO Algorithm

4 Experimental Settings

In order to make a fair comparison of BPSO and QIPSO, we fixed the same seed for random number generation so that the initial swarm population is same for both the algorithms. The number of particles in the swarm (swarm size) is taken as 30. A linearly decreasing inertia weight is used which starts at 0.9 and ends at 0.4, with the user defined parameters $c_1=2.0$ and $c_2=2.0$. We have also considered the measurement of diversity, which is an important aspect for checking the efficiency of the swarm. High diversity generally implies a better balance between exploration and exploitation. The diversity measure [14] of the swarm is taken as:

$$Diversity(S(t)) = \frac{1}{n_s} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n_x} (x_{ij}(t) - \overline{x_j(t)})^2} \quad (4)$$

where S is the swarm, $n_s = |S|$ is the swarm size, n_x is the dimensionality of the problem, x_{ij} is the j^{th} value of the i^{th} particle and $\overline{x_j(t)}$ is the average of the j^{th} dimension over all particles, i.e.

$$\overline{x_j}(t) = \frac{\sum_{i=1}^{n_s} x_{ij}(t)}{n_s} \tag{5}$$

For each algorithm, the maximum number of iterations allowed was set to 30,000 for 30 dimensions and 100,000 for 50 dimensions. For 30 dimensions, a total of 10 runs for each experimental setting were conducted and the average fitness of the best solutions throughout the run was recorded. For 50 dimensions, only a single run was performed and the best fitness value was recorded.

Table 1. Numerical benchmark problems

Function	Dim	Range	Min. Value
$f_1(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30/50	[-5.12,5.12]	0
$f_2(x) = \sum_{i=1}^n x_i^2$	30/50	[-5.12,5.12]	0
$f_3(x) = \frac{1}{4000} \sum_{i=0}^{n-1} x_i^2 - \sum_{i=0}^{n-1} \cos(\frac{x_i}{\sqrt{i+1}}) + 1$	30/50	[-600,600]	0
$f_4(x) = -\sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	30/50	[-500,500]	418.9829*n
$f_5(x) = \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} ((x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))) + (x_n - 1)(1 + \sin^2(2\pi x_n))$	30/50	[-10,10]	-21.5024
$f_6(x) = -\sum_{i=1}^n \sum_{j=1}^5 j \sin((j+1)x_i + j)$	30/50	[-10,10]	-
$f_7(x) = \sum_{i=0}^{n-1} x_i + \prod_{i=0}^{n-1} x_i $	30/50	[-10,10]	0
$f_8(x) = \max x_i , \quad 0 \leq i < n$	30/50	[-100,100]	0
$f_9(x) = (\sum_{i=1}^n x_i^2)^{1/4} [\sin^2(50(\sum_{i=1}^n x_i^2)^{1/10}) + 1.0]$	30/50	[-32.767,32.767]	0
$f_{10}(x) = (\sum_{i=0}^{n-1} (i+1)x_i^4) + rand[0,1]$	30/50	[-1.28,1.28]	0
$f_{11}(x) = \frac{1}{n} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	30/50	[-5,5]	-78.3323
$f_{12}(x) = \frac{\pi}{n} [10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(y_{i+1}\pi)] + (y_n - 1)^2] + \sum_{i=1}^n u(x_i, 10, 100, 4) *$	30/50	[-50,50]	0

Remark1: Functions sine and cosine take arguments in radians.

* $f_{12} : y_i = 1 + \frac{1}{4}(x_i + 1)$

$u(x, a, b, c) = b(x - a)^c, \quad \text{if } x > a$

$u(x, a, b, c) = b(-x - a)^c, \quad \text{if } x < -a$

$u(x, a, b, c) = 0, \quad \text{if } -a \leq x \leq a$

5 Experimental Results

In order to check the compatibility of the proposed QPSO algorithm we have tested it on a suite of 12 standard bench mark problems, generally used for testing the efficiency of global optimization algorithms, given in Table 1. The test bed comprises a variety of problems ranging from a simple spherical function to highly multimodal functions and also a noisy function (with changing optima). In Table 2, we have shown the results of problems with dimension 30 and in Table 3, are given the results of problems with increased dimension 50 in terms of best fitness

Table 2. Numerical results (dimension: 30)

Function	BPSO		QPSO	
	Mean Best Fitness	Diversity	Mean Best Fitness	Diversity
f_1	8.158668e+01	1.075614e-08	5.97167e-01	1.639511e-05
f_2	2.621440e+00	1.696744e-96	8.517991e-43	2.92223e-20
f_3	3.526500e-02	8.958573e-08	2.940000e-02	9.426306e-05
f_4	-8.406742e+03	4.550712e-06	-9.185054e+03	1.078529e+02
f_5	-1.301387e+01	5.412045e-08	-2.150231e+01	1.082100e-01
f_6	-1.556138e+02	1.508819e-08	-3.258289e+02	5.055760e-01
f_7	4.000000e+00	7.361003e-124	1.154568e-20	8.341826e-05
f_8	2.440000e-04	3.937000e-03	3.510000e-04	5.535000e-03
f_9	3.531709e+00	7.309540e-01	8.585330e-01	2.600000e-04
f_{10}	2.453298e+01	2.002060e-01	4.540630e-01	3.234900e-01
f_{11}	-7.701290e+01	1.799400e-07	-7.795535e+01	5.165870e-01
f_{12}	5.505851e-13	1.471375e-13	5.505851e-13	9.682600e-02

Table 3. Numerical results (dimension: 50)

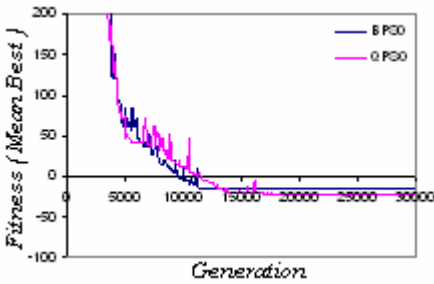
Function	BPSO		QPSO	
	Best Fitness	Diversity	Best Fitness	Diversity
f_1	1.923836e+02	2.412312e+00	0.000000e+00	8.158896e-11
f_2	5.768702e-08	1.492921e-07	9.714999e-74	9.735866e-37
f_3	1.966100e-02	8.376331e-08	2.466000e-03	3.654810e-01
f_4	-1.379016e+04	5.204209e-06	-1.407498e+04	1.464926e+02
f_5	-1.100000e+01	5.633709e-08	-2.150231e+01	1.596100e-02
f_6	-2.041946e+02	1.54925e-08	-5.243369e+02	5.555140e-01
f_7	9.782400e+00	3.776982e-15	7.651628e-37	1.238780e-01
f_8	1.514302e+00	2.753219e+01	4.875000e-03	1.240090e-01
f_9	1.001283e+01	2.475895e+01	2.509801e+00	2.387241e+00
f_{10}	1.105637e+02	2.459410e-01	6.736030e-01	3.990910e-01
f_{11}	-7.723433e+01	2.398338e-07	-7.776686e+01	1.11159e+00
f_{12}	2.535693e+00	4.611654e-06	3.303511e-13	3.386880e-01

value. In Table 4, we have shown the performance improvement of QPSO with BPSO. Fig. 2 show the mean best fitness curves for selected benchmark problems.

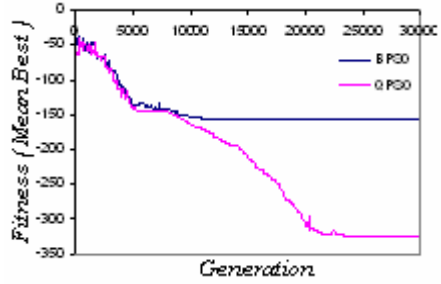
From the numerical results presented in Tables 2, 3, 4 and 5, it is evident that QPSO is a clear winner. It gave a better performance for all the problems with dimension 30 except for f_8 . However it is quite interesting to note that QPSO gave a much better performance for the same function (f_8) when the dimension is increased to 50. In terms of percentage of improvement (of average fitness function value), the

Table 4. Improvement (%) of QPSO in comparison with BPSO

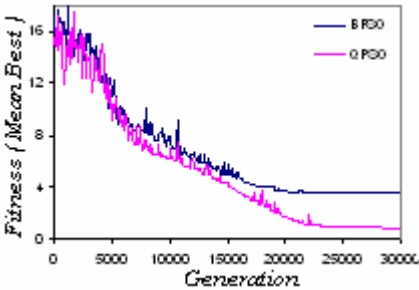
Function	Percentage Dim: 30	Percentage Dim: 50	Function	Percentage Dim: 30	Percentage Dim: 50
f_1	99.268058	100	f_7	100	100
f_2	100	100	f_8	-	99.678069
f_3	16.631221	87.457403	f_9	70.635432	74.934139
f_4	9.258188	2.065378	f_{10}	98.149173	99.390356
f_5	40.525241	95.475555	f_{11}	1.223753	0.689502
f_6	109.383101	156.782964	f_{12}	0.000000	100



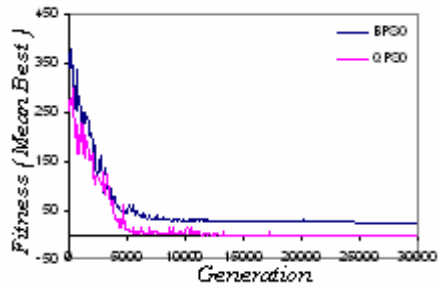
(a) f_5



(b) f_6



(c) f_9



(d) f_{10}

Fig. 2. Convergence graph for selected benchmark problems

results favor QPSO particularly for higher dimension (50). One interesting and noticeable fact about QPSO is that despite low diversity (although higher than BPSO in most of the cases) it gave good results.

6 Conclusion

In this research article, we proposed a simple and modified version of an integrated algorithm exploiting the features of PSO and EA. The novelty of the present work is the use of a nonlinear quadratic crossover operator and the manner in which it is applied. The new solution is accepted even if it is worse than the worst solution; this prevents the search space from contracting and thereby maintains diversity. The empirical results show that the proposed algorithm is quite competent for solving problems of dimensions up to 50. In future we shall also incorporate the phenomenon of mutation in the algorithm and will use it to solve the problems of higher dimensions.

Acknowledgments. The authors would like to thank the unknown referees for their comments, which helped in bringing the paper in more presentable format.

References

1. Angeline P. J.: Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Difference. The 7th Annual Conference on Evolutionary Programming, San Diego, USA, (1998).
2. Hu, X., Eberhart, R. C., and Shi, Y.: Swarm Intelligence for Permutation Optimization: A Case Study on n-Queens problem. In Proc. of IEEE Swarm Intelligence Symposium, pp. 243 – 246 (2003).
3. Miranda, V., and Fonseca, N.: EPSO – Best-of-two-worlds Meta-heuristic Applied to Power System problems. In Proc. of the IEEE Congress on Evolutionary Computation, Vol. 2, pp. 1080 – 1085 (2002).
4. Miranda, V., and Fonseca, N.: EPSO – Evolutionary Particle Swarm Optimization, a New Algorithm with Applications in Power Systems. In Proc. of the Asia Pacific IEEE/PES Transmission and Distribution Conference and Exhibition, Vol. 2, pp. 745 – 750 (2002).
5. Ting, T-O., Rao, M. V. C., Loo, C. K., and Ngu, S-S.: A New Class of Operators to Accelerate Particle Swarm Optimization. In Proc. of the IEEE Congress on Evolutionary Computation, Vol. 4, pp. 2406 – 2410 (2003).
6. Yao, X., and Liu, Y.: Fast Evolutionary Programming. In L. J. Fogel, P. J. Angeline, and T. B. Back, editors, Proceedings of the Fifth Annual Conference on Evolutionary Programming, MIT Press, pp. 451 – 460 (1996).
7. Yao, X., Liu, Y., and Lin, G.: Evolutionary Programming made Faster. IEEE Transactions on Evolutionary Computation, Vol. 3(2), pp. 82 – 102 (1999).
8. Angeline, P. J.: Using Selection to Improve Particle Swarm Optimization. In Proc. of the IEEE Congress on Evolutionary Computation, IEEE Press, pp. 84 – 89 (1998).
9. Clerc, M.: Think Locally, Act Locally: The Way of Life of Cheap-PSO, an Adaptive PSO. Technical Report, <http://clerc.maurice.free.fr/psol/>, (2001).

10. Kennedy, J.: The Particle Swarm: Social Adaptation of Knowledge. IEEE International Conference on Evolutionary Computation (Indianapolis, Indiana), IEEE Service Center, Piscataway, NJ, pg.303-308 (1997).
11. Eberhart, R.C., and Shi, Y.: Particle Swarm Optimization: developments, Applications and Resources. IEEE International Conference on Evolutionary Computation, pg. 81 -86 (2001).
12. Shi, Y. H., and Eberhart, R. C.: A Modified Particle Swarm Optimizer. IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, pg. 69 – 73 (1998).
13. Ali, M. M., and Torn, A.: Population Set Based Global Optimization Algorithms: Some Modifications and Numerical Studies. www.ima.umn.edu/preprints/, (2003).
14. Engelbrecht, A. P.:Fundamentals of Computational Swarm Intelligence. John Wiley & Sons Ltd., (2005).

Solving Bin Packing Problem with a Hybridization of Hard Computing and Soft Computing

Laura Cruz-Reyes, Diana Maritza Nieto-Yáñez, Pedro Tomás-Solis,
and Guadalupe Castilla Valdez

Instituto Tecnológico de Cd. Madero
Juventino Rosas y Jesús Urueta C.P. 89440 Cd. Madero Tamaulipas, México
lcruzreyes@prodigy.net.mx, diananieto@gmail.com,
pedro_tomas@msn.com, jgvaldez50@hotmail.com

Abstract. This paper presents a new hybrid intelligent system that solves the Bin Packing Problem. The methodology involves the fusion of Soft Computing by means a genetic algorithm and Hard Computing using limits criterion and deterministic strategies. The innovative proposal inverts minimum computational resources expressed in generations with a high level quality solution and shows the algorithm performance with statistical methods. The average theoretical ratio for 1370 standard instances was 1.002 and the best known solution was achieved in 83.72% of the cases. As future work, an exhaustive analysis of characteristics of the hardest instances is proposed; the purpose is to find new hybrid methods.

Keywords: bin packing problem, hard computing, soft computing, genetic algorithms, and hybrid intelligent system.

1 Introduction

Actually, several problems in real applications belong to NP-Hard class. That is characterized by the lack of efficient algorithms in order to solve them in an exact way. This class needs robust algorithms capable of adapting to different scenarios and obtaining good solutions without expending high volumes of resources. Two computational methodologies compete in the search of good solutions: Soft Computing (SC), that is known as a complementary set of techniques such as neural networks, fuzzy systems, or evolutionary computation which are able to deal with uncertainty, partial truth, and imprecision; and Hard Computing (HC), that is a huge set of traditional techniques. Both can be compared with the two hemispheres of the brain because humans apply logical and mathematical reasoning in order to solve problems using the left hemisphere of the brain (HC), and they also excel in pattern recognition, creativity, synthesis of ideas using the right hemisphere (SC).

Nevertheless the fusion of hard and soft computing is taking importance due to its capacity to combine strategies and to take the best characteristics of each of them. The present paper approaches the bin packing problem (BPP) through a hybrid intelligent system that involves the fusion of Soft and Hard Computing.

2 Bin Packing Problem

Bin Packing Problem is a classic NP-Hard problem of optimization and combinatorial, in which there exist a limitless number of bins whose capacity is an integer c , the number of items is n , the size of each item is s_i ($0 \leq i < n$) that is limited to ($0 < s_i \leq c$) [1] and its objective is to determine the smallest number of bins in which the items can be packed. In order to evaluate the problem, Falkenauer proposed in [2] a fitness function that permits to maximize the well-filled of all bins, letting measure the bin efficiency and using the maximum capacity. Where N is the number of bins used, $fill_i$ is the sum of the items size in the bin i and k is a constant that expresses a concentration of a well-filled bin, $k > 1$ (expression 1).

$$\max z = \frac{\sum_{i=1..N} (fill_i / C)^k}{N} \quad (1)$$

3 Fusion of Hard Computing and Soft Computing

Ovaska defined in [3] the SC-HC fusion as: “An algorithm-level or system-level union of particular soft computing and hard computing methodologies, in which either methodology is signal (parameter) or structure-wise dependent upon and receives functional reinforcement from the other”.

The different fusion schemes were classified in 12 core categories [3]. The methodology proposed in this paper uses *HC assisted SC* (HC//SC). In HC//SC, some internal data are first extracted from the SC algorithm (master), thus they are processed by the specific HC algorithm (slave), and fed back into the SC algorithm, which completes the primary task (Fig. 1). An example of this kind of fusion is a hybrid genetic algorithm with local search strategies.

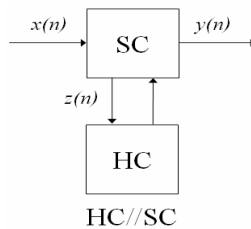


Fig. 1. Hard Computing assisted Soft Computing

4 State of Art

Diverse researchers in the computer sciences have been given to the task of providing hybrid algorithms to implementations related to BPP with promising results. However, given the nature of the problem, the study field still continues open.

In [4], Ducatelle developed a hybrid algorithm of Ant Colony Optimization with a local search algorithm. In large problems, the hybrid not always finds the best known

solution, besides it requires too much computational time, e.g. with 100 individuals demands between 1000 to 5000 generations depending of the size problem.

A hybrid heuristic was proposed in [5] by Alvim, with a fusion of hard and soft computing; the basic structure is composed by reduction techniques, use of limits, a greedy algorithm, redistributions and improvement by means of tabu search with $1/n$ iterations. More than 95% of the instances find a solution in the redistribution phase.

Falkenauer proposed to generate a hybrid with a grouping genetic algorithm (GGA) and a local optimization based on dominance criterion [2, 6]. He established as parameters 33% of mutation, 50% of crossover, population of 100 individuals and between 1000 to 5000 generations in order to solve each instance depending of its size. A GGA is used in grouping problems where the objective is to find a good clustering of the group members, such as the case of BPP. Falkenauer argued: "genetic algorithms configured with the classic strategies are insufficient for the solution of BPP". This affirmation is the basis of the algorithm proposed in this paper and was supported in [7] through experimental evidence with standard instances.

5 Solution Methodology

The present work is HC//SC fusion. SC is represented with the chromosomal strategy proposed by [2] where the complete bins are inherited in the *grouping crossover*, and complete bins are eliminated in the *grouping mutation*. HC is constituted by using specific determinist algorithms of BPP and limits criterion.

Fig. 2 describes the general algorithm. After the first generation, the probability to maintain the individual from the previous generation is proportional to its fitness and a random number (lines 3-8). Posteriori, an evolution process that includes evaluation, selection, crossover and mutation is developed, and the individual with the highest fitness is obtained (lines 9-12).

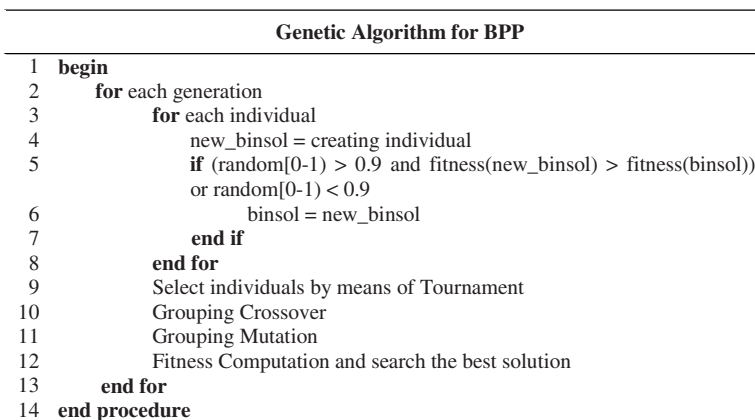


Fig. 2. General Procedure of AG

5.1 Problem Description

A fundamental part of the GA development is the creation of good individuals. Martello proposed in [8] the LBFD method that can be considerate as a HC. In LBFD, the starting point computes the L_2 limit (line 2). The L_2 biggest items are collocated in independent bins (lines 4-6), the rest of the items are packed using the BFD strategy. However, the method has been modified in order to use FF and apply randomness in the accommodation of the items residuals of L_2 (line 7-9) (Fig. 3).

Creating Individual with L_2

```

1  begin
2    Compute  $L_2$ 
3    Sort items in decreasing way
4    for the first  $L_2$  items
5      Put each item in an empty bin
6    end for
7    while there are items without accommodation
8      Select randomly an item and put it by means of FF
9    end while
10 end procedure

```

Fig. 3. Procedure to create individuals using the L_2 criterion limit

Another method incorporated to create individuals takes as base the DB3FD algorithm (Dual Best 3-Fit Decreasing) proposed in [5], where given an item into a bin, a pair of free items that fill the bin are searched. This method was modified for the requirements in Fig. 4, L_2 is computed (line 2) in order to calculate an initial number of bins near to the optimal. In lines 4-9, the algorithm tries to fill the first L_2 bins with three items. The rest of the items are accommodated randomize with FF.

Creating Individuals with DB3 and L_2

```

1  begin
2    Compute  $L_2$ 
3    Sort items in decreasing way
4    for the first  $L_2$  bins
5      Select randomly an item and insert it into the bin
6      if there are two free items that fill the bin
7        Insert the items into the bin
8      end if
9    end for
10   while there are items without accommodation
12     Select randomly an item and put it by means of FF
13   end while
14 end procedure

```

Fig. 4. Procedure to create individuals using the L_2 and fitting of three items (DB3)

The use of L_2 permits from the beginning the use of the Dominance Criterion approach described in [6], due that more dominants items are located in empty bins which are translated in a HC and time optimization. The expression 2 shows the

compute of L_2 , for an instance of BPP, where α is an integer, $0 \leq \alpha \leq c/2$, $N=\{0,1,\dots,n\}$, $J_1=\{j \in N : s_j > c-\alpha\}$, $J_2=\{j \in N : c-\alpha \geq s_j > c/2\}$ and $J_3=\{j \in N : c/2 \geq s_j > \alpha\}$. The L_2 value represents a low limit in order to search an optimal solution. An example of this can be seen in [8].

$$L(\alpha) = |J_1| + |J_2| + \max \left(0, \left\lceil \frac{\sum_{j \in J_3} s_j - (|J_2|c - \sum_{j \in J_2} s_j)}{c} \right\rceil \right) \quad (2)$$

$$L_2 = \max \{L(\alpha) : 0 \leq \alpha \leq c/2, \alpha \text{ int}\}$$

5.2 Grouping Crossover Procedure

The two-point grouping crossover was proposed in [2], but in the implementation of this algorithm some opportunity fields were analyzed and a second crossover was developed. Fig. 5 shows the new procedure, where the first segment of the 1st father is incorporated in a child (line 2). The fitness in each segment of the 2nd father is evaluated and the best segment is taken and incorporated in the child (lines 3-9). After that, the rest of the segments of the 1st father are added in the solution (lines 10-16); however, in the processes of lines 2-9 and 10-16 some items can already belong to the bins of the child, whereby these bins are discarded and the free items are reaccommodated with the HC: BFD (lines 17-19).

Grouping Crossover

```

1  begin
2  Copy in the child the bins of the 1st Segment of the 1st father
3  for each bin of the best segment in the 2nd father
4      if items of the actual bin is not in some bin of the child
5          Copy the bin in the child
6      else
7          Save as free_items, those that not belong to the solution
8      end if
9  end for
10 for each bin of the 1st father in the two and three segments
11     if items of the actual bin is not in some bin of the child
12         Copy the bin in the child
13     else
14         Save as free_items, those that not belong to the solution
15     end if
16 end for
17 if there are free_items
18     Packed free_items with BFD
19 end if
20 end procedure

```

Fig. 5. Grouping Crossover Procedure

5.3 Grouping Mutation Procedure

Taking up again the methodology of [2], which is contrary to normal mutation, the algorithm in Fig. 6 details the grouping mutation that eliminates a percentage of the bins more empty (lines 2-5). The free items are distributed with the HC: BFD (line 6).

Grouping Mutation	
1	Begin
2	for the number of bins to empty
3	Search the bin with the less weight accumulated and eliminate
4	Save as free_items those that belong to the bin
5	end for
6	Packed free_items with BFD
7	end procedure

Fig. 6. Grouping Mutation Procedure

5.4 Genetic Algorithm Hybridization – An Example

Fig. 7 shows the HC//SC procedure step by step with an example where: a population is generated and two individuals are selected (a); two cut points are selected in both fathers (b); 1st child is generated by injecting in the 1st father the best segment of the 2nd father (c); bins that duplicate items are eliminated and saved as free items not duplicated (d); so, with BFD the free items are reaccommodated (e); the generated solution, has better fitness than its progenitors, however, the mutation could improve it (f), for this, the emptiest bin is eliminated (4th) and the items are distributed by means of BFD and an optimal solution with three full bins is obtained (g).

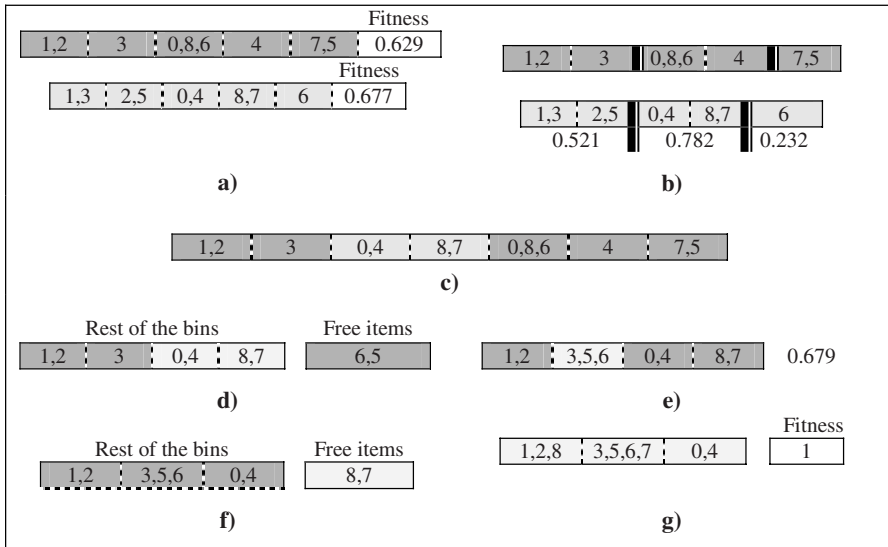


Fig. 7. Example of HC//SC

6 Experimentation and Results

In order to evaluate the methodology; a population of 30 individuals, 100 generations, 50% crossover and 20% mutation were established. These values are extremely lower than those reported by specialized literature which bears to a faster algorithm. The algorithm was developed in C language; 1370 standard instances divided in five sets

were taken from [9,10]. Table 1 shows: the number of instances where the best known solution was found for each set of instances (column 2); the theoretical ratios permit to see a very low error because in the instances where the best known solution is not reached, only one or two additional bins are used (column 3); the average time required in each instance is also observed (column 4); and the average generation where that solution was reached (column 5).

Table 1. Experimental Results per each set of instances

Instances	Optimum Instances	Theoretical Ratios	Time	Average Generation
Hard (10)	0	1.0214	1.4037	67.1
NCW (720)	661	1.0007	4.1411	47.5
NWBR (480)	442	1.0024	1.8451	42.1
T (80)	0	1.0235	1.0959	49.3
U (80)	44	1.0023	9.5270	75.9
Total (1370)	1147	1.0029	3.4534	47.5

Many researchers limit the experimentation in the generation of the best solutions without an exhaustive analysis of the algorithm performance which is a preponderant factor in order to evaluate the quality of the solution. Table 2 shows a sample of instances taken randomly from the total solved cases. The experimentation was repeated 30 times in order to verify the results consistency. As it can be seen in the low values of Standard Deviation, Variation Coefficient, Average Generation as well as the almost null difference among the maximum and minimum number of bins; the algorithm is consistent and robust.

Table 2. Statistical Analysis of the best solutions for configuration 6

Instance	Better Known Solution	Min.	Max.	Avg.	Std. Dev.	Variation Coef.	Time	Avg. Gen.	Theoretical Ratio
Hard9	56	57	58	57	0.489	0.008	1.426	67	1.018
N1c1w1_a	25	25	25	25	0.000	0.000	0.180	38	1.000
N4c1w1_m	246	246	246	246	0.000	0.000	11.480	56	1.000
N2w2b1r6	21	21	21	21	0.000	0.000	0.327	39	1.000
N3w4b3r1	22	22	22	22	0.000	0.000	0.756	40	1.000
t60_00	20	21	21	21	0.000	0.000	0.192	38	1.050
t501_10	167	168	169	168	0.179	0.001	2.588	53	1.005
u120_10	52	52	52	52	0.000	0.000	0.470	60	1.000
u1000_16	404	404	406	405	0.314	0.000	32.760	92	1.000
All Instances	---	---	---	---	0.082	0.001	3.453	48	1.002

7 Conclusions and Future Works

Through the use of hard-soft computing hybridization, it is possible to improve the results obtained with pure algorithms. Diverse deterministic methods have been

performed for the BPP, such as FFD and BFD. These algorithms permit to generate easily new methods that can be applied in diverse genetic hybridization processes. This paper shows the hard-soft fusion of a Genetic Algorithm with those deterministic algorithms and a strategy based on dominance criterion.

The dominance criterion was applied to the population generation process: the dominance items are determined using the limit L_2 , additionally that the best individuals are rescued from previous generations. This strategy implies high probability to get good progenitors and the fast algorithm convergence (100 generations); in the state of art, the analyzed researchers require a high number of generations in order to achieve their results. Regarding to the best known solution, the experiments revealed that the proposed hybrid intelligent system is able of reaching this value in 83.72% of the evaluated instances, which means a number not too far away, taking in account an error almost null represented by a theoretical ratio of 1.0029 (one or two bins of difference).

Therefore, several future works can be developed in order to improve the algorithm performance, such as: analysis of other limits; implementation of new methods that take knowledge from previous generations; a very deep revision of instances where the main differences were observed, having as a goal to reach the exhaustive analysis of their characteristics in order to create hybrid methods related to them; the algorithm tuning for concrete and real problems as loading in the bottled products distribution.

References

1. Baase, S.: Computer Algorithms, Introduction to Design and Analysis, Editorial Addison-Wesley Publishing Company, (1998)
2. Falkenauer E.: A Hybrid Grouping Genetic Algorithm for Bin Packing, CRIF - Research Centre for Belgian Metalworking Industry Industrial Management and Automation, (1996)
3. Ovaska S. J., Kamiya A., Chen YQ.: Fusion of Soft Computing and Hard Computing: Computational Structures and Characteristic Features, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, VOL. 36, NO. 3, MAY 2006
4. Ducatelle F.: Ant Colony Optimisation for Bin Packing and Cutting Stock Problems, School of Artificial Intelligence, University of Edinburgh, (2001)
5. Alvim A., Ribeiro C., Glover F., Aloise D.: A Hybrid Improvement Heuristic for the One-Dimensional Bin Packing Problem. Research report, Catholic University of Rio de Janeiro, Department of Computer Science, Rio de Janeiro, Brazil, (2003)
6. Martello S., Toth P.: Lower Bounds and Reduction Procedures for the Bin Packing Problem, in Discrete Applied Mathematics, Holland, Elsevier Science Publishers, (1990).
7. Nieto, D., Tomas, P., Castilla, G., Fraire, H., Cruz, L.: Configuración de Algoritmos Genéticos para el Problema de Empacado de Objetos, 13th International Congress on Computer Science Research (CIICC), Cd. Madero Tamaulipas México, 2006.
8. Martello S., Toth P.: Knapsack Problems: Algorithms and Computer Implementations. Wiley & Sons Ltd. (1990)
9. Beasley's OR-Library, <http://www.ms.ic.ac.uk/info.html>
10. Operational Research Library, <http://www.bwl.tu-darmstadt.de/bwl3/forsch/projekte/binpp>

Design of Artificial Neural Networks Based on Genetic Algorithms to Forecast Time Series

Juan Peralta, German Gutierrez, and Araceli Sanchis

CAOS Group, Computer Science Department, University Carlos III of Madrid
{jperalta,ggutierr,masm}@inf.uc3m.es

Abstract. In this work an initial approach to design Artificial Neural Networks to forecast time series is tackle, and the automatic process to design is carried out by a Genetic Algorithm. A key issue for these kinds of approaches is what information is included in the chromosome that represents an Artificial Neural Network. There are two principal ideas about this question: first, the chromosome contains information about parameters of the topology, architecture, learning parameters, etc. of the Artificial Neural Network, i.e. Direct Encoding Scheme; second, the chromosome contains the necessary information so that a constructive method gives rise to an Artificial Neural Network topology (or architecture), i.e. Indirect Encoding Scheme. The results for a Direct Encoding Scheme (in order to compare with Indirect Encoding Schemes developed in future works) to design Artificial Neural Networks for NN3 Forecasting Time Series Competition are shown.

Keywords: Time Series Forecasting, Artificial Neural Networks Desing, Genetic Algorithms, Direct Encoding Scheme.

1 Introduction

In order to acquire knowledge, it is so interesting to know what the future will look like, i.e. forecast the future from the observed past. The forecasting task can be performed by several techniques as Statistical methods, Immune Systems, and Artificial Neural Networks (ANN).

This contribution reports the methodology to carry out the automatic design of ANN that tackles the Forecasting Time Series problem taken from NN3 Forecasting Time Series Competition [1].

NN3 suggest a forecasting time series competition to evaluate automatic methodologies for design ANN by means of several time series. Two important research questions are pointed out: the performance of ANN compared with others forecasting methods; and “best practice” methodologies carried out by researchers to model ANN for time series forecasting. The task in NN3 Forecasting Competition is forecast several time series, not all of them with the same ANN, but the participants have to develop just one methodology to obtain a different ANN to forecast each time series.

The first step in order to design an ANN is setting the kind of ANN with which solve the forecasting task, and the learning algorithm used. According to [2], that show the approximation capability of Multilayer Perceptron (MLP), we have focused

on MLP with only one hidden layer and Backpropagation (BP) as learning algorithm to forecasting time series.

The design of ANN consists in setting its parameter values. In the case of MLP with only one hidden layer and BP the parameters are: number of inputs nodes, number of hidden neurons (number of output neurons is placed by the problem), which is the connection pattern (how the nodes are connected), the whole set of connection weights (or learning rate, minimum error to reach and the maximum training cycles as parameters that allow together with BP obtain it).

The process of designing ANN could be considered as a search problem within all possible designs. And this search could be done by Genetic Algorithms (GA).

Several works approach the design of ANN using Evolutionary Techniques. Some of them use Direct Encoding Schemes (DES) [3,4], the others using Indirect Encoding Scheme (IES) [5,6,7,8,9]. For DES the chromosome contains information about parameters of the topology, architecture, learning parameters, etc. of the Artificial Neural Network. In IES the chromosome contains the necessary information so that a constructive method gives rise to an Artificial Neural Network topology (or architecture). In [10] evidence about performance of ANN obtained from GA are reported. Ajith Abraham [11] shows an automatic framework for optimization ANN in an adaptive way, and Xin Yao et. al. [12] try to spell out the future trends of the field.

The work reported in this contribution is based on a “Research Paper” of first author (and supervised by the later authors) for Master in Computer Science and Technology at Carlos III of Madrid University¹.

The final object of our approach is consider Sparsely Connected MLP to forecast time series, and use both Indirect Encoding Scheme, one based on Cellular Automata [8] and other based on Bidimensional Grammar [9], to design ANN using GA. A previous work for designing ANN to forecast using GA based on Cellular Automata is developed by Salah and Al-Salqan in [13].

The paper is organized as follows. Sec 2 reviews questions about how to tackle forecast task with ANNs. Sec 3 approaches design of ANN to forecast with GA. In Sec 4 experimental setup and results are shown. And finally, conclusions and future works are described in Sec 5.

2 Forecasting Time Series with ANN

There are several works that tackles the forecasting time series task with ANN, not only computer science researchers, but statisticians as well. This reveals the full consideration of ANN (as a data driven learning machine) into forecasting theory.

Before using an ANN to forecast, it has to be designed, i.e. establishing the suitable value for each degree of freedom of the ANN [14] (kind of net, number of input nodes, number of outputs neurons, number of hidden layer, number of hidden neurons, the connections from one node to another , etc). The design process is more an “art” based on test and error and the experience of human designer, than an algorithm. In [15] Crone proposed an “extensive modeling approach” to review

¹ <http://www.uc3m.es/uc3m/gral/TC/ESMAOF/CTII/ctii.html>

several designs of ANNs, and finally Crone and Preßmar show in [16] an evaluation framework to NN modeling in forecasting, and literature review.

In order that a single ANN could work, and forecasting, with time series values, an initial step from original values of time series have to be done, i.e. normalize the data. And, once the ANN gives those values, the inverse process is done. This step is important as the ANN will learn just the normalized values.

The problem of forecasting time series with ANN is considered as obtaining the relationship of the value of period "t" (i.e. the net has only one output neuron) and the values of previous periods, i.e the function (1):

$$a_t = f(a_{t-1}, a_{t-2}, \dots, a_{t-k}). \quad (1)$$

Therefore, the time series will be transform into a pattern set, it depend on the k inputs nodes of a particular ANN. Each pattern consists in:

- "k" inputs values, that correspond to "k" normalized previous values of period t: $a_{t-1}, a_{t-2}, \dots, a_{t-k}$
- One output value, that corresponds to normalized time series value of period t.

The complete patterns set are ordered into the same way the time series is. This patterns set will be used to train and test the ANN, then it will be split into two sets, train and test sets. The train set will be obtained from the first $m\%$ (e.g 70%) and the test set will be obtained from the rest of the complete patterns set.

If hand design of ANN is carry out, several topologies (i.e. different number of inputs nodes and number of different number of neurons of only one hidden layer), with different learning rates are trained. For each of them, train and test error are obtained, and one with better generalization capability (i.e. less test error and a good train error) is selected to generate forecast values.

3 ANN Design with Genetic Algorithms

The problem of design ANN could be seen as a search problem into the space of all possible ANN. And that search can be done by a GA [17] using exploitation and exploration. There three crucial issues: the solution's space; how each solutions is codified into a chromosome, i.e. Encoding Scheme; and what is looking for, translated into the function fitness.

As a first approach to design ANN to forecasting time series, a Direct Encoding Scheme for Full Connected MLP has been considered.

For this Direct Encoding Scheme the information placed into the chromosome is:

- Number of inputs nodes (i).
- Number of hidden nodes (h).
- Learning rate, for BP learning algorithm (α)

The value of learning rate, " α ", is between 0 and 1, and the value of " i " and " h " is limited by a maximum number of inputs nodes (max_inputs) and a maximum number

of hidden nodes (*max_hidden*), respectively. These maximum values are related, by a factor "a", with the number of periods of time series what values are known (*nts*), see ec. (2)

$$\begin{aligned} \max_inputs &= a \times nts . \\ \max_hidden &= 2 \times \max_inputs . \\ a &= 0.3 . \end{aligned} \tag{2}$$

Into the chromosome, two decimal digits, i.e two genes, are used to codify the value "i", other two for "h" and the last two for "α". This way, the values of "i", "h" and "α" are obtained, from the chromosome, as it can see in (3)

$$\begin{aligned} \text{chrom: } &g_{i1} \ g_{i2} \ g_{h1} \ g_{h2} \ g_{\alpha 1} \ g_{\alpha 2} . \\ i &= \max_inputs \times ((g_{i1} \cdot 10 + g_{i2}) / 100) . \\ h &= \max_hidden \times ((g_{h1} \cdot 10 + g_{h2}) / 100) . \\ \alpha &= ((g_{\alpha 1} \cdot 10 + g_{\alpha 2}) / 100) . \end{aligned} \tag{3}$$

The search process (GA) begins with a random population, i.e set of randomly generated chromosomes. Later, the fitness value for each one of the individual of the population is obtained (a). Once that it is already done the GA operators as Elitism, Selection, Crossover and Mutation are applied in order to generate the population of next generation, i.e. set of chromosomes (b). The steps (a) and (b) are iteratively executed till a maximum number of generations is reached.

To obtained the fitness value of a chromosome:

1. The train patterns and test patterns sets are obtained, depending on the number of inputs nodes of the net, as was said above (Sec.2).
2. Then, the connection weights are randomly initialized, and the net is trained (using Stuttgart Neural Network Simulator (SNNS) binary tools [19]) a maximum training cycles.

The fitness value will be the minimum error test reached in training process; it doesn't have to be in the last training cycle. The architecture of the net (topology + connections weights) when the test error is minimum in training process is saved to be used later for forecasting time series values.

Once that GA reaches the last generation, the best individual from all generations is used to forecast the time series.

4 Experimental Setup and Results

The experiments related to our approach have been carried out on the Reduced Database of NN3 Competition. Nowadays, the results of time series, i.e. the time series values to be forecasted still have not been published till the moment. Therefore, the forecasted values obtained by our system cannot be evaluated or compared with real values. Nevertheless, the forecast results for ANN will be shown together forecast obtained with statistical techniques.

To normalize the time series values (Sec 2), so the nets will be able to work with them, it has to be considered the shape that time series has [16], specially trend (upward or downward) component. As the time series values have to be rescale, into

the numerical range value $[0, I]$, considering not only the known values, but the future values (those to be forecasted), the maximum and minimum limits for normalizing ($max2norm$, $min2norm$ respectively) cannot be just the maximum (max) and minimum (min) known time series values. A margin from max and min has to be set if future values are higher or lower than they already are. Equation (4) show how are obtained $max2norm$ and $min2norm$.

$$\begin{aligned} max2norm &= max + (0.1 \times (max - min)) . \\ min2norm &= min - (0.1 \times (max - min)) . \end{aligned} \quad (4)$$

The parameters of GA described in Sec 3 are: population size, 50; maximum number of generations, 100; percentage of the best individual that stay unchangeable to the next generation (percentage of elitism), 10%; crossover method, mutation probability, $(1/length_chrom) = 0.17$.

The parameter for learning process and getting fitness values are: percentage from whole pattern set to obtain train and test subsets (Sec 2), 70 % and 30% respectively; and maximum training cycles, 10^4 . Remember that number of inputs nodes, hidden neurons and learning rate values are obtained from chromosome.

And, as it was said before, the fitness function is the minimum error test reached in training process (Sec 3). At the end of this process, the topology obtained will be the one described by the information inside the chromosome with the best fitness, plus the connection weights obtained at the end of the training process of the net.

4.1 Results of Genetic Algorithm

The GA to design an ANN has been carried out for the 11 time series of NN3 Reduced Database. Nevertheless, because of the limited space, the results for only two of them will be shown, those that seem especially significant, nn3-101 and nn3-108 time series.

For each of them, it will be shown on one side the evolution fitness graph along the generations (Fig 1), and on the other hand the known time series values together with the forecasted ones obtained by two methods (Fig 2 & 3). The two methods are: our GA method and a Statistical one (ARIMA [18], implemented in AUTOBOX tool).

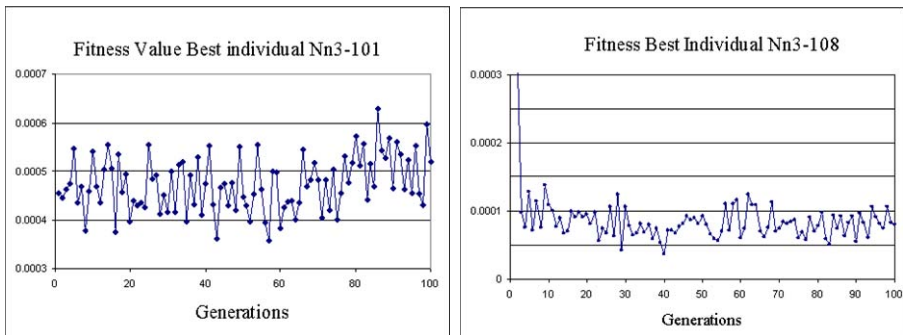


Fig. 1. Fitness along generations for nn3-101 and nn3-108

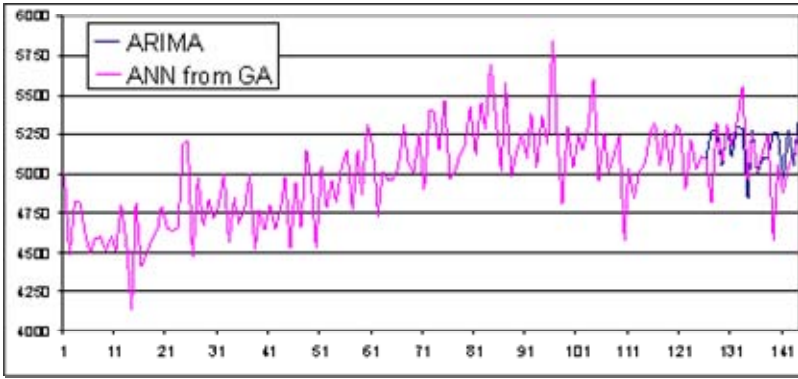


Fig. 2. Forecasted values for nn3-101

For our method, after GA runs, the best ANN model is selected and used to forecast just the 18 values asked by the competition.

As far as we check, the GA for nn3-101 does not improve the fitness value along 100 generations, i.e. GA is not better than a random search for this time series and these GA parameters. However, for nn3-108 GA decreases the error test from the beginning. On the other hand it is interesting to emphasize the saw shape which has the graph of the fitness evolution (Fig 1), this happens because in each fitness evaluation of an individual along the GA the weights are randomly initialized for learning process. The set of connection weights are not included into the chromosome, though the search process generates an individual not only with an appropriate topology but also with a right set of free parameters (i.e. set of connection weights).

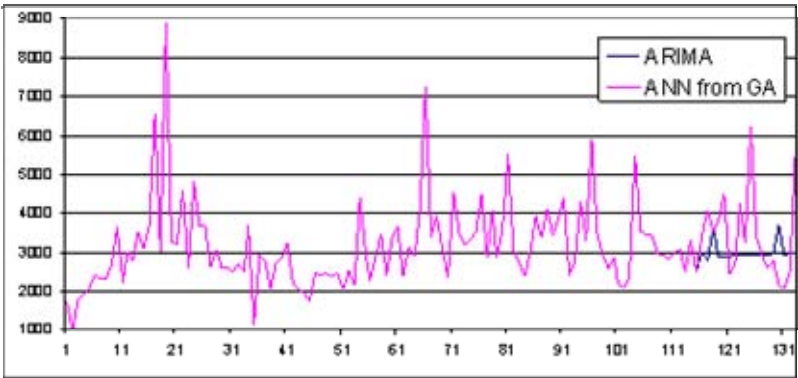


Fig. 3. Forecasted values for nn3-108

In relation to forecasting, it seems that, for both time series, with ANN obtained by GA goes on with the trend of time series.

5 Conclusions and Future Works

As we said before the set of connection weights are not included into the chromosome, though the search process generates an individual not only with an appropriate topology but also with a right set of free parameters (i.e. set of connection weights).

Furthermore, the Partial Autocorrelation Function (PACF) [18] indicates that the optimal value of previous elements of the Time Series (i.e. input nodes to be considered for ANN) to forecast a new one is 15. Nevertheless, the results of GA are different for both Time Series, nn3-101 and nn3-108, being the optimal previous values (i.e. input nodes) 30 and 34 respectively to forecast a new one.

The system reported here is a first approximation with Direct Encoding to design ANN. In Future works our approach will consider, not full connected nets, but Sparsely Connected MLP to obtain a better forecasting, using both Indirect Encoding Scheme, one based on Cellular Automata [8] and other based on Bidimensional Grammar [9], to design ANN using GA.

References

1. 2006/07 Forecasting Competition for Neural Networks & Computational Intelligence. <http://www.neural-forecasting-competition.com/>
2. G. Cybenko. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2, 303-314, 1989
3. T. Ash. Dynamic Node Creation in Backpropagation Networks ICS Report 8901, The Institute for Cognitive Science, University of California, San Diego (Saiensu-sh, 1988), 1988
4. D.B. Fogel, Fogel L.J. and Porto V.W. Evolving Neural Network, *Biological Cybernetics*, 63, 487-493, 1990.
5. Gruau F. "Genetic Synthesis of Boolean Neural Networks with a Cell Rewriting Developmental Process". *Proceedings of COGANN-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pp. 55-74, IEEE Computer Society Press, 1992.
6. Yao, X. and Lin, Y. A new evolutionary system for evolving artificial neural networks, *Transactions on Neural Networks*, 8(3): 694-713, 1997.
7. Kitano, H.: Designing Neural Networks using Genetic Algorithms with Graph Generation System, *Complex Systems*, 4, 461-476, 1990.
8. G. Gutiérrez, A. Sanchis, P. Isasi, J.M. Molina and I. M. Galván. Non-direct Encoding method based on Cellular Automata to design Neural Network Architectures. *Computer and Informatics*, 24 2005, No3.
9. Studying the Capacity of Grammatical Encoding to Generate FNN Architectures. Germán Gutiérrez, Beatriz Garcá, JosM. Molina and Araceli Sanchis. *Computational Methods in Neural Modeling, Lecture Notes in Computer Science, Volume 2686/2003. IWANN 2003*.
10. James V Hansen, James B McDonald, Ray D Nelson. Time Series Prediction With Genetic-Algorithm Designed Neural Networks: An Empirical Comparison With Modern Statistical Models. *Computational Intelligence Journal*, Vol. 15, 171-184, 1999.
11. Ajith Abraham, Meta-Learning Evolutionary Artificial Neural Networks, *Neurocomputing Journal, Elsevier Science, Netherlands*, Vol. 56c, pp. 1-38, 2004.

12. H. Garis, D. Fogel, M. Conrad and X. Yao. Special Issue on evolutionary neural systems. *Neurocomputing*, 42, 1-8, 2002
13. Asma Abu Salah, Y.A. Meta-Learning Evolutionary Artificial Neural Networks Using Cellular Configurations: Experimental Works *Intelligent Computing*, 2006, Volume 4113/2006, pages 178-193
14. Haykin, S. Simon & Schuster (ed.) *Neural Networks. A Comprehensive Foundation* Prentice Hall, 1999
15. Crone, S. F. Stepwise Selection of Artificial Neural Networks Models for Time Series Prediction *Journal of Intelligent Systems*, Department of Management Science Lancaster University Management School Lancaster, United Kingdom, 2005,
16. Crone, S. F. and Preßmar, D. B. 2006. An extended evaluation framework for neural network publications in sales forecasting. *Proceeding of International Conference on Artificial intelligence and Applications*, 2006
17. Fogel, D. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Wiley-IEEE Press, 1998.
18. ARIMA and PACF: <http://www.statsoft.com/textbook/sttimser.html>
19. Prof. Dr. Andreas Zell, WSI Computer Science Department, Computer Architecture, Software, Artificial Neural Networks <http://www-ra.informatik.uni-tuebingen.de/SNNS/>

Appendix

SNNS binary tools (`ff_bignet` and `batchman`) [19], has been used for generate and training ANN. The whole source code is available in `{gguatierr,jperalta}@inf.uc3m.es`.

Experimental Analysis for the Lennard-Jones Problem Solution

Héctor J. Fraire Huacuja¹, David Romero Vargas², Guadalupe Castilla Valdez¹,
Carlos A. Camacho Andrade³, Georgina Castillo Valdez¹,
and JoséA. Martínez Flores¹

¹ Instituto Tecnológico de Ciudad Madero,
1o. de Mayo y Sor Juana I. de la Cruz S/N, C.P. 89440
Cd. Madero, Tamaulipas, México
hfraire@prodigy.net.mx, gpe_cas@yahoo.com.mx,
ginacastillo78@hotmail.com, jose.mtz@gmail.com

² Instituto de Matemáticas,
Universidad Nacional Autónoma de México.
Av. Universidad s/n, Col. Lomas de Chamilpa,
62210 Cuernavaca, Mor., Mexico
davidr@matcuer.unam.mx

³ Instituto Tecnológico de Costa Grande,
Av. Amado Nervo, Manzana 30, Lote 1, Col. El Limó C.P. 40880
Costa Grande, Guerrero, México
camachoaca@hotmail.com

Abstract. In this paper the problem of determining the atomic cluster configurations that minimize the Lennard-Jones potential energy is approached. Traditional studies are oriented to improve the quality of the solution and practically do not present statistical information to support the efficiency of the reported solution methods. Without this type of evidence the effectiveness of these methods might highly be dependent only on the capacity of the available computing resources. In this work it is proposed to incorporate statistical information on the performance of the solution methods. An advantage of this approach is that when the performance tests are standardized and statistically supported, we can take advantage of efficient solution methods that have been tested only in conditions of modest computing resources. An experimental study of the problem is presented in which the generated statistical information is used to identify two potential areas to improve the performance of the evaluated method.

Keywords: Analysis of algorithms, Lennard-Jones potential, Genetic algorithms.

1 Introduction

In this work the problem of determining the atomic cluster configurations that minimize the Lennard-Jones potential energy is approached. The problem consists in determining the positions, $p_1, \dots, p_n \in R^3$ of an n atoms cluster, in such a way that the potential energy $\sum_{i=1..n-1} \sum_{j=i+1..n} (r_{ij}^{-12} - 2r_{ij}^{-6})$ generated by the atomic interactions is minimized. Here, r_{ij} is the Euclidian distance between points p_i and p_j .

Some characteristics of the problem are: the number of terms of the objective function grows as the combinations of n in 2, the objective function is not convex, the number of variables grows at the rate of $3n$, and the number of local minima grows up exponentially [1]. In order to discretize the problem space solution, reticular structures denominated *lattices* are used to define the sites in the space where the atoms of a cluster can be initially placed. In the method evaluated here, given a point $p \in R^3$ a *spherical slice* $S(p,n)$ of a lattice is the set of the n sites of the lattice nearest to p [2, 3]. The sites of a spherical slice are used to place the n atoms of a cluster and get a “good” initial configuration. By selecting distinct points p various initial configurations are obtained leading each to some local optimum.

Some of the most efficient solution methods use a hybrid approach that combines a local optimizer with a global search metaheuristic. The local optimizer permits moving the atoms from their initial configuration to the nearest local optima configuration. The metaheuristic generates neighbor configurations to diversify the search and avoid to get stuck in local optima. This technique allows the efficient determination of low Lennard-Jones energy configurations.

2 Related Work

Some relevant works related with the experimental analysis of algorithms are the following ones.

Barr [4] and Johnson [5] consider that the effectiveness of a methodology can be proved by means of theoretical analysis or with experimental tests. They also indicate that the theoretical analysis is often limited to analyze the performance of the algorithms in real applications. The approach of the worst case tends to produce too pessimistic levels, and it does not provide information about the performance of the algorithms on specific instances. The approach of the average case requires the determination of a suitable model of the distribution of the data associated to the used instances, which often is difficult to build. Johnson [5] considers that in recent years the community of the theoretical analysts has shown an increasing interest in the experimental analysis of algorithms. This interest is due to the recognition that the theoretical analysis is insufficient to analyze the performance of the algorithms of the real world [6].

Barr [4], Johnson [5], McGeoch [6, 7], Moret [8], and Fraire [9] propose a series of principles to design computational experiments that allow producing relevant scientific knowledge. They propose that the experimental analysis of algorithms must consider the following elements: to include relevant computational experiments, which must be related with other experiments reported in the literature, that can be reproduced and compared with those realized in other investigations, and they must use reasonably efficient implementations and include standard test cases and statistical support. In order to assure the conclusions solidity they recommend the incorporation of variance reduction techniques in the computational experiments.

Now we describe the most relevant works that use a hybrid approach combining a local optimizer with a global search metaheuristic to solve the Lennard-Jones problem.

In [2, 3] a solution method based on a genetic algorithm is proposed. It uses lattices to generate atomic initial configurations using spherical slices; each generated configuration is optimized locally. The best found solution for groups of 13 to 309 atoms, the number of evaluations of the objective function and the number of generations are reported. It does not specify the local optimizer used and it does not present statistical evidence of the efficiency of the solution.

In [10, 11, 12] a different approach is applied which consists in using lattices for the generation of configurations of atoms, an evaluation function and a genetic algorithm [10, 11] and a greedy algorithm [12] to improve the atomic configurations. Each improved configuration is locally optimized using L-BFGS, and the process is repeated during a certain number of iterations. For groups of 310 to 1000 atoms the best found solution is reported. For 20 instances in the range from 13 to 310 atoms they report the percentage of successes achieved in 50 attempts of solution, the time average used in seconds of CPU and the best found solution.

As we can observe, these works are fundamentally oriented to improve the quality of the solution for each instance since they do not present experimental evidence of the method's efficiency. For example, the best solution found is reported, without indicating neither the frequency with which this solution is obtained nor the observed variance. Without this kind of evidence the effectiveness of the proposed methods can be highly dependent of the capacity of the available resources. In this work we propose to orient the investigation of this problem towards the construction of efficient solution methods that guarantee high quality solutions with an efficient consumption of the available resources. An advantage of this approach is that when the performance tests are standardized and statistically supported, we can take advantage of efficient solutions that have been proven in limited conditions of computational resources.

To attain this goal we propose to adjust the experimental analysis of the performance of the Lennard-Jones solution methods, to the principles proposed to realize the experimental analysis of algorithms and to produce relevant scientific knowledge.

3 Solution Approach

3.1 Reference Solution Method

The solution method implemented is based on the method reported in [3]. The most important differences with respect to this proposal are that a trust region local optimizer is used [13, 14]. An icosahedral lattice is used for the generation of the initial atomic configurations [15].

The initial population contains 9 locally optimized configurations. Seven of them are generated from only one spherical slice and the other two from two additional spherical slices. The crossover process consists in randomly selecting six configurations to form pairs. For each pair the two optimized configurations (the

parents) are cut with a plane or a sphere, or applying a rotation to them. Then the four gotten sections are combined to generate two offspring configurations (the sons) which are locally optimized too. An additional make-up process is applied to six configurations generated in the crossover, to prevent computer overflow, and consists in swap atoms from high to low energy sites. The mutation process consists in rotating an optimized configuration and optimizes the resultant configuration. The selection process uses 9 initial configurations and 6 crossover and make-up generated configurations to get 9 configurations of the next generation. From 15 considered configurations, 7 are selected and two additional configurations are generated with the two additional spherical slices considered in the initial configurations generation process. Additional details of the genetic algorithm can be found in [3].

3.2 Proposed Experimental Analysis

In order to formalize the experimental analysis of the Lennard-Jones problem, 30 experiments with each of the considered instances was realized. Each experiment consisted in solving an instance applying the solution method whose performance was evaluated. Each experiment incorporated a variance reduction mechanism. The experiment outputs that correspond to the defined indicators of the performance were determined. At the end of all experiments the maximum, minimum, average, and variance of the sample generated for each considered indicator were determined. Finally this information can be used to identify weak areas, thus helping to improve the design of the evaluated algorithm. Without this type of evidence the effectiveness of the analyzed solution methods can be highly dependent of the capacity of the available computing resources. Currently many papers report experimental results with metaheuristics algorithms but they do not include statistical information about the solution process.

4 Experimentation Results

In this section some results of the performance experimental analysis of the Lennard Jones problem solution are shown. In order to realize the experimentation a 64 bits workstation Power Edge Dell with two 3.06 GHZ Xeon processors with 4 GB RAM, and 60 GB HD was used. The algorithms were implemented using Matlab 7.0 running over Microsoft Windows Server 2003. The test cases used are the 13-28 atomic clusters.

Table 1 shows the comparative results for the number of evaluations of the objective function realized when the instances are solved with the described method. The first column contains the atoms number of the corresponding instance, the second column contains the number of experiments where the best reported solution with respect to the 30 realized experiments was found, the third column contains the evaluation number of the objective function used to obtains the best solution reported

in [2], the next four columns contain the minimum, maximum, average, and standard deviation values of the evaluation number of the objective function.

Table 1. Analysis of the number of the objective function evaluations

n	Successes	Evaluations of the objective function				
		Min. [2]	Min.	Max.	Avg.	Std. Dev.
13	30	754	20	20	20	0
14	30	112	35	35	35	0
15	30	132	83	4,207	559	914
16	30	177	141	5,877	684	1,281
17	30	153	50	50	50	0
18	30	6,620	486	241,822	47,507	58,751
19	30	9,901	415	91,685	19,752	20,544
20	30	866	579	23,189	7,578	6,739
21	30	175	627	215,815	49,370	48,417
22	30	1,542	188	55,258	14,261	14,611
23	30	2,041	207	385,756	68,042	85,780
24	30	163	193	206,202	32,307	44,802
25	30	283	792	276,596	59,325	70,030
26	30	2,213	3,450	432,548	193,511	141,636
27	30	3,599	3,291	268,700	70,898	65,946
28	30	411	2,915	140,587	40,621	36,031
	Total	29,142	13,142			

Table 2 shows the comparative results for the number of generations required for the genetic algorithm to obtain the best solution. The first column contains the atoms number of the corresponding instance, the second column contains the number of experiments where the best reported solution with respect to the 30 realized experiments was found, the third column contains the generation number reported in [2], the next four columns contain the minimum, maximum, average, and standard deviation values of the generation number.

In Table 1 we observe that in all the 30 experiments the best known solution was obtained with the method proposed. Also we can see that with the exception of 21 and 24-28 atomic clusters, the method proposed shows a better performance than the reference method. Also the method proposed has a superior performance with respect to the total evaluations number (see totals). In Table 2 a similar behavior is observed with respect to the generation number used to obtain the best solution.

Based in the experimental evidence we conclude that the solution method proposed has a better performance than the reference method. From the statistical information we observe that the average values are greater than the minimum values in many cases. When this difference is too large we can consider reviewing the process to search an explanation, and get more knowledge about the solution process to improve the algorithm design. Additionally, Tables 1 and 2 show that the variance of the number of evaluations of the objective function and the number of generations is

high. This suggests the possibility of improving the algorithm design to reduce these variations. Since the method incorporates a technique of variance reduction, an additional reduction can be obtained modifying the design of the algorithm.

Table 2. Analysis of the number of generations realized

n	Successes	Generations				
		Min. [2]	Min.	Max.	Avg.	Std. Dev.
13	30	0	0	0	0	0
14	30	0	0	0	0	0
15	30	0	0	5	0.57	1.10
16	30	0	0	7	0.53	1.46
17	30	0	0	0	0	0
18	30	2	0	227	48.5	59.84
19	30	3	0	92	19.2	20.01
20	30	0	1	25	7.53	7.05
21	30	0	1	225	49.23	49.89
22	30	0	0	56	13.8	14.44
23	30	0	0	364	65.3	81.91
24	30	0	0	193	32.13	43.29
25	30	0	1	256	59.07	68.71
26	30	0	3	395	186.75	137.41
27	30	1	3	275	65.5	64.09
28	30	0	3	130	37.13	33.41
	Totals	6	12			

An advantage of this approach is that it permits to use the obtained statistical information so as to identify weak areas, thus helping to improve the design of the evaluated algorithm.

5 Conclusions

In this work the problem of determining the atomic cluster configurations that minimize the Lennard-Jones potential energy is approached. The most efficient solution methods use a hybrid approach that combine a local optimizer with a global search metaheuristic. Traditional approaches are fundamentally oriented to improve the quality of the solution for each instance and they do not present experimental evidence of the efficiency of the solution method. Without this kind of evidence the effectiveness of the proposed methods can be highly dependent of the capacity of the available resources. In this work we propose to orient the investigation of this problem towards the construction of efficient solution methods that guarantee high quality solutions with an efficient consumption of the available resources. An advantage of this approach is that using standardized and statistically supported performance tests we can take advantage of efficient solution methods that have been proved in conditions of reduced computational resources. With the

purpose of validating the feasibility of our approach, an experimental study of the Lennard-Jones problem is presented, in which, from the generated statistical information, two potential areas to improve the performance of the evaluated method are identified.

At the moment different design strategies are being evaluated which include modifications to the make-up function and the design of a new poly-geometric lattice.

Acknowledgments

We would like to thank CONACYT, COTACYT, and DGEST for their support to this research project.

References

1. Hoare M.: H. Structure and Dynamics of Simple Microclusters: Advances in Chemical Physics. No 40, pp. 49-135 (1979)
2. D. Romero, C. Barró, S. Góñez.: A Genetic Algorithm for Lennard-Jones Atomic Clusters. Applied Mathematics Letters, pp. 85-90 (1998)
3. D. Romero, C. Barró, S. Góñez.: The optimal geometry of Lennard-Jones clusters:148-309. Computer Physics Communications 123, pp. 87-96 (1999)
4. Barr, R.S., Golden, B.L., Kelly, J., Steward, W.R., Resende, M.: Guidelines for Designing and Reporting on Computational Experiments with Heuristic Methods. Proceedings of International Conference on Metaheuristics for Optimization. Kluwer Publishing, Norwell, MA, pp. 1-17 (2001)
5. Johnson, D.S., McGeoch, L.A.: Experimental Analysis of Heuristics for the STSP. In: Gutin, G., Punnen, A. (eds.): The Traveling Salesman Problem and its Variations. Kluwer Academic Publishers, Dordrecht, pp. 369-443 (2002)
6. McGeoch, C.: Analyzing Algorithms by Simulation: Variance Reduction Techniques and Simulation Speedups. ACM Computer Survey. Vol. 24 No. 2, pp. 1995-212 (1992)
7. McGeoch, C.: Experimental Analysis of Algorithms. In: Pardalos, P.M., Romeijn, H.E. (eds.): Handbook of Global Optimization, Vol. 2, pp. 489-513 (2002)
8. Moret, B.M.E.: "Toward a Discipline of Experimental Algorithmics". In: Goldwasser, M.H., Johnson, D.S., McGeoch, C. (eds.): Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges, Series DIMACS, Vol. 5, pp. 197-214 (2003)
9. Fraire H.: Una metodología para el diseño de la fragmentación y ubicación en grandes bases de datos distribuidas. PhD Thesis. CENIDET, Cuernavaca, México (2005)
10. X. Shao, Y. Xiang, H. Jiang, W. Cai.: An efficient method based on lattice construction and the genetic algorithm for optimization of large Lennard-Jones clusters. J. Phys. Chem. A.108, pp. 3586-3592 (2004)
11. X. Shao, Y. Xiang, L. Cheng, W. Cai.: Structural distribution of Lennard-Jones clusters containing 562 to 1000 atoms. J. Phys. Chem. A.108, pp. 9516-9520 (2004)
12. X. Shao, Y. Xiang, W. Cai.: Structural transition from icosahedra to decahedra of large Lennard-Jones clusters. J. Phys. Chem. A.109, pp. 5193-5197 (2005)

13. Coleman, T.F., Y. Li.: An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM Journal on Optimization*. Vol. 6, pp. 418-445 (1996)
14. Coleman, T.F., Y. Li: On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds. *Mathematical Programming*, Vol. 67, No. 2, pp. 189-224 (1994)
15. Xue, G.L., Maier, R.S., Rosen, J.B.: A discrete-continuous algorithm for molecular energy minimization. *IEEE Computer Society Press*, pp. 778-786 (1992)

Application of Genetic Algorithms to Strip Hot Rolling Scheduling

Carlos A. Hernández Carreón¹, Héctor J. Fraire Huacuja², Karla Espriella Fernandez³,
Guadalupe Castilla Valdez², and Juana E. Mancilla Tolama¹

¹ Instituto Politécnico Nacional, ESFM. Unidad Profesional Adolfo López Mateos.
Av. Instituto Politécnico Nacional, S/N. 9° piso, Zacatenco. 07738-México, D.F.
cahc05@yahoo.com.mx; emtolama@hotmail.com

² Instituto Tecnológico de Ciudad Madero, México.
1o. de Mayo y Sor Juana I. de la Cruz S/N.
89440-Cd. Madero, Tamaulipas, México
hfraire@prodigy.net.mx, gpe_cas@yahoo.com.mx

³ Instituto Politécnico Nacional, CICATA, U. Altamira.
km 14.5. Carretera Tampico – Puerto Industrial Altamira.
89600-Ciudad Altamira, Tamaulipas, México
karla_rouge@yahoo.com

Abstract. This paper presents an application of a genetic algorithm (GA) to the scheduling of hot rolling mills. The objective function used is based on earlier developments on flow stress modeling of steels. A hybrid two-phase procedure was applied in order to calculate the optimal pass reductions, in terms of minimum total rolling time. In the first phase, a non-linear optimization function was applied to evaluate the computational cost to the problem solution. For the second phase, a GA was applied. A comparison with two-point and simulated binary (SBX) crossover operators was established. The results were validated with data of industrial schedules. A GA with SBX crossover operator is shown to be an efficient method to calculate the multi-pass schedules at reduced processing time.

Keywords: Hot Rolling Scheduling, Genetic Algorithms, Simulated Binary Crossover, Optimization.

1 Introduction

Hot rolling scheduling is an empirical procedure aimed to calculate reductions and rolling speeds in a succession of individual passes for which the optimum must be found. For a given steel composition and initial thickness, the strip deformed in each roll pass has a specific temperature and microstructural evolution. This is a result of the strip initial conditions before the previous deformation, and the time elapsed to the end of it. Furthermore, factors as roll properties (bending) and motor specifications (back-up roll bearing capacity, torque, voltage, current) have influence on the optimum values of reductions [1]. These factors are acting as restrictions to the problem of optimization [2]. The problem of optimizing rolling schedule is solved by using motor power and thickness constrains for each deformation pass.

When hot working is done in a tandem mill, exit parameter values are the input at the entrance of next step. Such type of situations are difficult to analyze mathematically because of the great number of involved variables and the sequential relations that exist between any two passes; also because of the restrictions that must be applied not to exceed the nominal capacities of the rolling stands [3].

In this paper, we present the results to optimize and compare the exit thickness in each rolling stand, for a specific number of stands, using genetic algorithm (GA). To optimize the thickness, we applied a hybrid two-phase strategy. In Phase 1 we present an initial identification and assessment of the important parameters and constrain for modelling hot rolling to evaluate the computational cost of the problem solution. In the second phase we use a GAs-based heuristic technique for optimization. All rolling times calculated were compared with data from an industrial hot rolling mill.

2 Related Works

2.1 Hot Rolling Model

We used a more realistic approach to modeling hot rolling steel. The model was constructed in two steps. In the first step, a stress-strain curve model is used to calculate the flow stress [4, 5]. Taking into account the influence of the main alloying (C, Mn, Si, Mo) and micro alloying (Ti, V, Nb) elements in steels, process variables (strain ε , temperature T , strain rate $\dot{\varepsilon}$) and the initial grain size D_0 , the flow curves was calculated by constitutive equations of the general form $\sigma = f(Z, \varepsilon, D_0)$, where Z is the Zener-Hollomon parameter given by $Z = \dot{\varepsilon} \exp(Q_{\text{def}}/R T)$, where $\dot{\varepsilon}$ is the strain rate, T is the absolute temperature of the deformation, R is the Gas constant, and Q_{def} is the apparent activation energy for deformation of steel [6].

The resistance to deformation $\bar{\sigma}$ can be calculated as follows:

$$\bar{\sigma} = \frac{1}{\varepsilon_0} \int_0^{\varepsilon_0} \sigma \, d\varepsilon \quad (1)$$

The roll-separating force and torque have been derived with the Ford & Alexander model [7]. The power requirement for a given pass can be established by an empirical expression based on rolling torque. Thus, overloading of the motor can be assessed.

2.2 Genetic Algorithms

The problem of minimizing the hot rolling time in a reversing mill stand was studied by Chakraborti [8], using a simplified hot rolling model. In comparison to other traditional optimization methods, this author showed the efficiency of GAs to calculate the hot rolling schedule [9].

A GA is a heuristic search and optimization method based on the principles of evolution and natural selection. GA explores as many as possible parallel solutions,

rather than doing many calculations around a local solution. A GA manipulates a large population to evolve under certain rules to optimize the function to fit [10]. The fitness stage evaluates the function for each chromosome. In the reproduction stage, the members of the new solution are selected based on their fitness. The next step generates new solutions applying reproduction operators, selecting genes from parent chromosomes, creating a new offspring (crossover), and introducing random changes in the new offspring (mutation). The next generation is selected evaluating the cost associated with the offspring and mutated chromosomes. In order to improve the estimates of each solution a method to reduce the variance was applied [11].

3 Solution Approach

The aim of the present problem is to find the optimal reductions to be applied in the different passes which give a minimum total finishing rolling time. The time necessary to deform steel between the n rolls along with the inter-pass time are determined for various initial and final thicknesses. For an n -pass rolling mill, the chromosome is represented by a collection of $n+1$ thicknesses (genes), from h_0 to h_f corresponding to $n+1$ passes. The first n element corresponds to the thickness at the entrance of the finish mill. The last n element corresponds to the final thickness. Both are input data in this problem. The procedure of hot rolling optimization algorithm is described as follows.

3.1 Genetic Algorithm Phase

3.1.1 Initial Population

The initial population consists of a set of randomly generated thicknesses. All the elements in each population are generated randomly such that they lie in the range of the initial and final thicknesses, given by the user.

Applying equation (2), the first generation was determined. For each computation, the thicknesses constrain ($h_0 > h_1 > \dots > h_f$) was respected.

3.1.2 Evaluation of Fitness Function

The objective function was computed for each individual in the population as the sum of the times corresponding to each pass, $t = t_0 + t_1 + t_2 + t_3 + t_4 + t_f$.

The fitness values of each population were calculated and the expected rolling time of each element was obtained. The optimal solution is that of minimizing the total rolling time. To guarantee the minimum value, it is important to know industrial hot rolling schedules. The optimum time calculated can not be greater than the corresponding value given by industrial schedules.

The selection of the best elements was elitist. The elitist process is to select two individuals with the highest fitness and pass them onto the next generation without crossover and mutation. The new generation is formed with the best individuals, and will be used for the next operator, crossover.

3.1.3 Crossover and Mutation

The chromosomes of parents are recombined by random selection during crossover. The process is repeated with different parents until there are a suitable number of solutions in the next generation population.

A well-known two-point crossover operation was first worked and a second method, the simulated binary crossover operator (SBX), was employed later on. In the SBX operator, if two parents $x_1 = [h_0, h_1, h_2, h_3, h_4, h_5]$ and $x_2 = [h_0, h_1, h_2, h_3, h_4, h_5]$ are selected for crossover, then the $x'_1 = [h'_0, h'_1, h'_2, h'_3, h'_4, h'_5]$ and $x'_2 = [h'_0, h'_1, h'_2, h'_3, h'_4, h'_5]$ offspring can be computed by mean of a probability distribution function [12].

The mutation operator force to have solutions by randomly introducing changes in some of the chromosomes. Mutation is done to find a resulting population with a better fitness than the crossover operator.

4 Results and Discussion

The data for the problem case is presented in Table 1. The data is representative of a finishing hot rolling mill. In the first phase of this study, we evaluated the computational cost to the problem solution, calculating initially the total rolling time with a non-linear optimization method. All the computational experiments were done in Intel Celeron equipment (2.4 GHz, 512 RAM). For a hot rolling mill formed with 2 until 6 stands, given an initial and final thickness (Table 1), a set of thickness for each intermediate rolling pass can be computed, such that minimizes the total rolling time.

In each case, the initial guess value of the intermediate thicknesses for any given n number of roll-pass was calculated using a mean reduction r , given as [13]:

$$r = 1 - n \sqrt[n]{\frac{h_0}{h_f}} \quad (2)$$

where n is the number of deformation passes. A Matlab program was developed to calculate the thicknesses using a numerical optimization function with a non-linear constrains regression algorithm. This calculation takes into account the restrictions established for thickness ($h_0 > h_1 > \dots > h_f$) and rolling power ($P_{i, \text{calc}} < P_{i, \text{motor}}$), where $P_{i, \text{motor}}$ is the nominal motor power for any stand i .

Table 2 presents the results obtained with 2 to 6 stands rolling mills using the non-linear constrain optimization Matlab function. The rolling time has increased when the number of stands increased. For the last stand, the execution time was longer than 160 min, so we finished the computation by convenience. The computational cost in terms of execution times was in any case in the order of minutes.

Table 1. Numerical data for the scheduling problem, corresponding to a hot rolling finish mill

(1) Work piece		
Material	Very Low Carbon Steel:	
Chemical composition (wt. pct.)	C=0.053; Mn=0.784; Si=0.017, Ti=0.01; Nb=0.042	
Initial thickness of strip h_0 (mm)	48	
Final thickness of strip h_f (mm)	30	
Width of strip w (mm)	991	
(2) Roll		
Roll diameter D (mm)	752.65	
Peripheral speed v (mm/s)	810	
(3) Parameters and constrains		
Inter-stands length d (mm)	1,500	
Motor power:	--	
Stand 1 $P1$ (kW)	2,800	
Stand 2 $P2$ (kW)	2,000	
Stand 3 $P3$ (kW)	1,000	
Stand 4 $P4$ (kW)	800	
Stand 5 $P5$ (kW)	600	
Stand 6 $P6$ (kW)	300	

Table 2. The computational cost using a non-linear constrains optimization

Pass no.	h_1 (mm)	h_2 (mm)	h_3 (mm)	h_4 (mm)	h_5 (mm)	Rolling time, s	Execution time, min
2	25					2.079	4 – 6
3	25	15				3.977	14.6
4	30	19.9	9.9			5.872	20
5	34.8	24.7	14.3	10		7.761	16
6	—	—	—	—	—	—	> 160

The GA proposed in this work was implemented in C++ language. Mutation and crossover operations were performed in 80 % and 40 % of the population, respectively. Using the two-point crossover, the computational cost was estimated and compared with respect to the Matlab optimization function, for a 4-pass rolling mill case. Ten computational experiments were run using different initial and final thicknesses up to 30 generations (Table 3). From Table 3, it is clear that all

experiments computed by using GA are better than those using non-linear optimization. In every computational experiment, the variance associated to the GA execution time was calculated and included in Table 3. The great variation from 1 to 4 orders of magnitude in variance was the reason to apply a variance reduction technique [9].

Table 3. Computational cost for Matlab and GA optimization methods

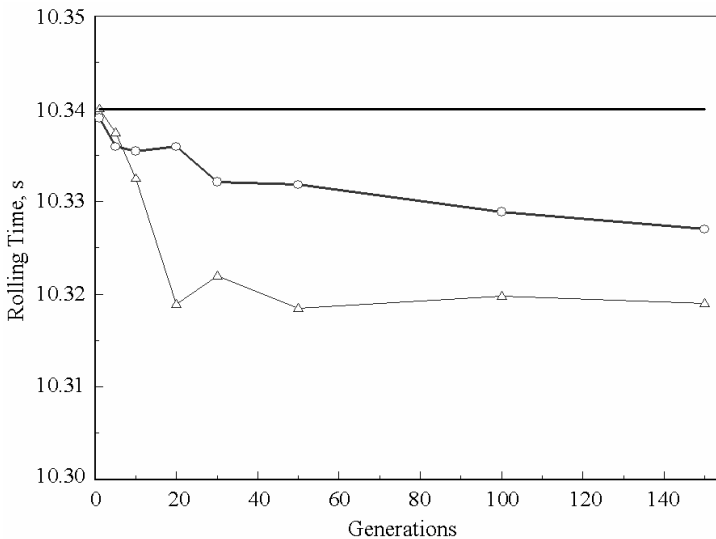
Computational experiment	h_0-h_f (cm)	Execution time, minutes		
		Matlab	GAs	variance
1	4.80-0.30	783	218.57	21,529.0
2	5.00-0.38	415	51.80	1,466.7
3	4.5-0.30	1072	19.06	23.51
4	4.0-0.20	879	31.56	46.59
5	3.9-0.10	129	87.73	609.35
6	4.6-0.50	501	21.53	36.53
7	4.7-0.20	497	40.46	73.36
8	4.3-0.30	735	29.76	99.97
9	4.2-0.30	1,291	25.70	67.37
10	4.4-0.40	740	18.93	40.75

Table 4 presents the results obtained, where it is shown that the variance was reduced from 7 to 10 orders of magnitude. In all experiments, the execution times were reduced significantly with respect to the values in Table 3. The cost of computing has significantly reduced to 5.8 s. The reason for this may be that GA maintains a better initial population diversity, which can help to search more efficiently a set of local suboptimal estimations than the Matlab non-linear restrictions algorithm, whose solution depends on the quality of the initial guess.

Finally, the fitness using a crossover SBX operator function was calculated after 150 generations for the average of ten tests. At the beginning (less than 10 generations), the figure 1 show that the optimum rolling time calculated with a two-point crossover method were smaller than those for SBX crossover method. It means that the GA using two-point crossover method found good results in short times, because it maintains better population diversity at the initial stage, finding better optimums than with the SBX crossover method. With more than 10 generations, the SBX recombination scheme found smaller minima optimums than with the two-point scheme. The comparison of both crossover methods with the industrial scheduled hot rolling time is shown in figure 1. It can be seen that solutions with SBX are better than with two-point crossover.

Table 4. Computational cost for Matlab and GA optimization methods using variance reduction

Computational experiment	Thickness h_0-h_f (cm)	Execution time, seconds		
		Matlab	GAs	variance
1	4.80-0.30	5.866	5.867	1.23 e-05
2	5.00-0.38	5.8562	5.8575	1.31 e-05
3	4.5-0.30	5.841	5.8486	1.80 e-05
4	4.0-0.20	5.81585	5.8299	2.74 e-05
5	3.9-0.10	5.8	5.8179	1.54 e-05
6	4.6-0.50	5.827	5.8298	3.85 e-05
7	4.7-0.20	5.827	5.854	1.25 e-05
8	4.3-0.30	5.833	5.844	7.27 e-05
9	4.2-0.30	5.831	5.84	7.58 e-05
10	4.4-0.40	5.8334	5.8447	1.82 e-05

**Fig. 1.** Evolution if the rolling time using two-point crossover (○), SBX crossover (Δ), in comparison with industrial rolling time (—)

5 Conclusions

The method proposed in the present work based on a GA is aimed to study hot rolling scheduling problem. The computational cost was higher with Matlab numerical optimization method than with GA. Based on the total rolling times calculated, it can be seen that GA is more effective for optimizing a hot rolling schedule than numerical

optimization functions. The application of a variance reduction method reduces significantly the computational cost on both approaches. Using two different methods of crossover, the crossover SBX operator gives lower rolling times than using the two-point crossover operator on 150 generations. With both methods, the GA calculates smaller rolling times than the industrial one.

Acknowledgments. Thanks to the following companies for the generous supply of material and technical information. HYLISA, Monterrey, N.L., Mexico; INTEG Process Group, Wexford PA, USA.

Supported by IPN grant CGPI 20061402.

References

1. Boubel, D., Fazan, B., Ratte, P.: Optimum Computer Control of the Plate Mill. Results of Industrial Operation. IRSID Report. No. RE501 (1978)
2. Akgerman, N., Lahoti, G., Altan, T.: Computer Aided Roll Pass Design. *J. Appl. Metalwork.* 30--40 (1980)
3. Tamura, R., Nagai, M., Nakagawa, Y., Tanizaki, T., Nakajima, H.: Synchronized Scheduling Method in Manufacturing Steel Sheets *Int. Trans. Opl. Res.* 5, 189--199 (1998)
4. Medina, S. F., Hernández, C. A.: General Expression of the Zener-Hollomon Parameter as a Function of the Chemical Composition of Low Alloy and the Microalloyed Steels. *Acta Met. Mat.* 44, 137--148 (1996)
5. Medina, S. F., Hernández, C. A.: Influence of Chemical Compo on Peak Strain of Deformed Austenite in Low Alloy and Microalloyed Steels. *Acta Met. Mat.* 44, 149--154 (1996)
6. Hernández, C. A., Medina, S. F., Ruiz, J.: Modelling Austenite Flow Curves in Low Alloy and Microalloyed Steels. *Acta Met. Mat.* 44, 155--163 (1996)
7. Ford, H., Alexander, J.: Simplified Hot-Rolling Calculations. *J. Inst. Met.* 92, 397--404 (1963)
8. Haupt, R., Haupt, S.: *Practical Genetic Algorithms.* John Wiley & Sons, New York (1998)
9. Fraire, H., Castilla, G., Hernández, A., Aguilar, S., Castillo, G., Camacho, C.: Reducción de la Varianza en la Evaluación Experimental de Algoritmos Metaheurísticos. In: *Proc. 13th Int. Conf. on Computer Sci. Res. CIICC'06*, pp. 14--22 (2006)
10. Deb, K., Beyer, H.-G.: Self-Adaptive Genetic Algorithms with Simulated Binary Crossover. Technical Report No. CI-61/99, Dept. Computer Science, University of Dortmund, (1999)
11. Wright, H., Hope, T.: Rolling of stainless steel in wide hot strip mills. *Metals Tech.* 9, 565--576 (1975)
12. Chakraborti, N., Kumar, A.: The Optimal Scheduling of a Reversing Strip Mill: Studies Using Multipopulation Genetic Algorithms and Differential Evolution. *Mat. and Manuf. Proc.* 18, 433--445 (2003)
13. Chakraborti, N., Kumar, B., Babu, S., Moitra, S., Mukhopadhyay, A.: Optimizing Surface Profiles during Hot Rolling: A Genetic Algorithms Based Multi-Objective Optimization. *Comp. Mat. Sci.* 37, 159--165 (2006)

Synergy of PSO and Bacterial Foraging Optimization – A Comparative Study on Numerical Benchmarks

Arijit Biswas¹, Sambarta Dasgupta¹, Swagatam Das¹, and Ajith Abraham²

¹ Dept. of Electronics and Telecommunication Engg,
Jadavpur University, Kolkata, India

² Norwegian University of Science and Technology, Norway
arijitbiswas87@gmail.com, sambartadg@gmail.com,
swagatamdass19@yahoo.co.in, ajith.abraham@ieee.org

Abstract. Social foraging behavior of *Escherichia coli* bacteria has recently been explored to develop a novel algorithm for distributed optimization and control. The Bacterial Foraging Optimization Algorithm (BFOA), as it is called now, is currently gaining popularity in the community of researchers, for its effectiveness in solving certain difficult real-world optimization problems. Until now, very little research work has been undertaken to improve the convergence speed and accuracy of the basic BFOA over multi-modal fitness landscapes. This article comes up with a hybrid approach involving Particle Swarm Optimization (PSO) and BFOA algorithm for optimizing multi-modal and high dimensional functions. The proposed hybrid algorithm has been extensively compared with the original BFOA algorithm, the classical *g*-best PSO algorithm and a state of the art version of the PSO. The new method is shown to be statistically significantly better on a five-function test-bed and one difficult engineering optimization problem of spread spectrum radar poly-phase code design.

Keywords: Bacterial Foraging, hybrid optimization, particle swarm optimization, Radar poly-phase code design.

1 Introduction

In 2001, Prof. K. M. Passino proposed an optimization technique known as Bacterial Foraging Optimization Algorithm (BFOA) based on the foraging strategies of the *E. Coli* bacterium cells [1]. Until date there have been a few successful applications of the said algorithm in optimal control engineering, harmonic estimation [2], transmission loss reduction [3], machine learning [4] and so on. Experimentation with several benchmark functions reveal that BFOA possesses a poor convergence behavior over multi-modal and rough fitness landscapes as compared to other naturally inspired optimization techniques like the Genetic Algorithm (GA) [5] Particle Swarm Optimization (PSO) [6] and Differential Evolution (DE)[7]. Its performance is also heavily affected with the growth of search space dimensionality. In 2007, Kim *et al.* proposed a hybrid approach involving GA and BFOA for function optimization [8]. The proposed algorithm outperformed both GA and BFOA over a few numerical benchmarks and a practical PID tuner design problem.

In this article we come up with a hybrid optimization technique, which synergistically couples the BFOA with the PSO. The later is a very popular optimization algorithm these days and it draws inspiration from the group behavior of a bird flock or school of fish etc. The proposed algorithm performs local search through the chemotactic movement operation of BFOA whereas the global search over the entire search space is accomplished by a PSO operator. In this way it balances between *exploration and exploitation* enjoying best of both the worlds.

The proposed algorithm, referred to as Bacterial Swarm Optimization (BSO) has been extensively compared with the classical PSO, a state-of-the-art variant of PSO and the original BFOA over a test suit of five well-known benchmark functions and also on a practical optimization problem of spread spectrum radar poly-phase code design [9]. The following performance metrics were used in the comparative study (i) quality of the final solution, (ii) convergence speed, (iii) robustness and (iv) scalability. Such comparison reflects the superiority of the proposed approach.

2 The Bacterial Swarm Optimization Algorithm

Particle swarm optimization (PSO) [6] is a stochastic optimization technique that draws inspiration from the behavior of a flock of birds or the collective intelligence of a group of social insects with limited individual capabilities. In PSO a population of particles is initialized with random positions \vec{X}_i and velocities \vec{V}_i , and a fitness function, f , is evaluated, using the particle’s positional coordinates as input values. In an n-dimensional search space, $\vec{X}_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})$ and $\vec{V}_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{in})$. Positions and velocities are adjusted, and the function is evaluated with the new coordinates at each time-step. The velocity and position update equations for the d-th dimension of the i-th particle in the swarm may be given as follows:

$$\left. \begin{aligned} V_{id}(t+1) &= \omega \cdot V_{id}(t) + C_1 \cdot \varphi_1 \cdot (P_{lid} - X_{id}(t)) + C_2 \cdot \varphi_2 \cdot (P_{gd} - X_{id}(t)) \\ X_{id}(t+1) &= X_{id}(t) + V_{id}(t+1) \end{aligned} \right\} \quad (1)$$

The BFOA is on the other hand is based upon search and optimal foraging decision making capabilities of the *E.Coli* bacteria [10]. The coordinates of a bacterium here represent an individual solution of the optimization problem. Such a set of trial solutions converges towards the optimal solution following the foraging group dynamics of the bacteria population. Chemo-tactic movement is continued until a bacterium goes in the direction of positive nutrient gradient (i. e. increasing fitness). After a certain number of complete swims the best half of the population undergoes reproduction, eliminating the rest of the population. In order to escape local optima, an elimination-dispersion event is carried out where, some bacteria are liquidated at random with a very small probability and the new replacements are initialized at random locations of the search space. A detailed description of the complete algorithm can be traced in [1].

In the proposed approach, after undergoing a chemo-tactic step, each bacterium also gets mutated by a PSO operator. In this phase, the bacterium is stochastically attracted towards the globally best position found so far in the entire population at

current time and also towards its previous heading direction. The PSO operator uses only the ‘social’ component and eliminates the ‘cognitive’ component as the local search in different regions of the search space is already taken care of by the chemo-tactic steps of the BFOA algorithm. In what follows we briefly outline the new BSO algorithm step by step.

[Step 1] Initialize parameters $n, N, N_C, N_S, N_{re}, N_{ed}, P_{ed}, C(i) (i=1,2,\dots,N), \phi^i$.

Where,

n : Dimension of the search space,

N : The number of bacteria in the population,

N_C : No. of Chemo-tactic steps,

N_{re} : The number of reproduction steps,

N_{ed} : The number of elimination-dispersal events,

P_{ed} : Elimination-dispersal with probability,

$C(i)$: The size of the step taken in the random direction specified by the tumble.

ω : The inertia weight.

C_1 : Swarm Confidence.

$\vec{\theta}(i, j, k)$: Position vector of the i -th bacterium, in j -th chemotactic step, and k -th reproduction.

\vec{V}_i : Velocity vector of the i -th bacterium.

[Step 2] Update the following:

$J(i, j, k)$: Cost or fitness value of the i -th bacterium in the j th chemo-taxis, and k -th reproduction loop.

$\vec{\theta}_{g_best}$: Position vector of the best position found by all bacteria.

$J_{best}(i, j, k)$: Fitness of the best position found so far.

[Step 3] Reproduction loop: $k=k+1$

[Step 4] Chemotaxis loop: $j=j+1$

[substep a] For $i=1,2,\dots,N$, take a chemotactic step for bacterium i as follows.

[substep b] Compute fitness function, $J(i, j, k)$.

[substep c] Let $J_{last}=J(i, j, k)$ to save this value since we may find a better cost via a run.

[substep d] Tumble: generate a random vector $\Delta(i) \in R^n$ with each element

$$\Delta_m(i), m = 1, 2, \dots, p, \text{ a random number on } [-1, 1].$$

[substep e] Move: Let $\theta(i, j+1, k) = \theta(i, j, k) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$

[substep f] Compute $J(i, j+1, k)$.

[substep g] Swim: we consider only the i -th bacterium is swimming while the others are not moving then.

- i) Let $m=0$ (counter for swim length).
 ii) While $m < N_s$ (if have not climbed down too long).

- Let $m=m+1$.
- If $J(i, j+1, k) < J_{last}$ (if doing better),
 let $J_{last} = J(i, j+1, k)$ and let

$$\theta(i, j+1, k) = \theta(i, j, k) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

and use this $\theta(i, j+1, k)$ to compute the new $J(i, j+1, k)$ as we did in [sub step f]

- Else, let $m = N_s$. This is the end of the while statement.

[Substep 5] Mutation with PSO Operator

For $i = 1, 2, \dots, S$

- Update the $\vec{\theta}_{g_best}$ and $J_{best}(i, j, k)$
- Update position and velocity of the d -th coordinate of the i -th bacterium according to the following rule:

$$V_{id}^{new} = \omega V_{id}^{old} + C_1 \cdot \phi_1 \cdot (\theta_{g_best_d} - \theta_{id}^{old}(i, j+1, k))$$

$$\theta_{id}^{new}(i, j+1, k) = \theta_{id}^{old}(i, j+1, k) + V_{id}^{new}$$

[Step 6] Let $S_r = S/2$.

The S_r bacteria with highest cost function (J) values die and the other half of bacteria population with the best values split (and the copies that are made are placed at the same location as their parent).

[Step 7] If $k < N_{re}$, go to step 1. We have not reached the specified number of reproduction steps. So we start the next generation in the chemo-taxis loop.

3 Experimental Setup

3.1 Benchmark Functions Used

The performance of the BSO algorithm has been evaluated on a test bed of 5 well-known benchmark functions [10] as shown in table 1. In table 1 n represents the number of dimensions and we used $n=15, 30, 45$ and 60 . All the benchmark functions except f_5 have their global minima at the origin or very near to the origin. For Shekel's Foxholes the global minimum is at $(-31.95, -31.95)$ and its value is 0.998 . An asymmetrical initialization procedure has been used here following the work reported in [12]. A famous NP-hard problem of optimal design arises in the field of spread spectrum radar poly-phase codes [9]. The four competitor algorithms have been applied on this problem. We omit the detailed description of the associated fitness function in order to save space.

Table 1. Benchmark Functions Used

Function	Mathematical Representation
Rosenbrock	$f_1(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Rastrigin	$f_2(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Ackley	$f_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$
Shekel's Foxholes	$f_5(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$

3.2 Simulation Strategy

The proposed BSO algorithm has been compared with the classical PSO, original BFOA and a recently developed variant of PSO known as MPSO-TVAC [13]. In the later version velocity of a randomly selected particle is perturbed by a random mutation step size if the global best-sofar solution does not improve for a predetermined number of generations. Following [13] we keep the mutation step size proportional to the maximum allowable velocity. For all the competitive algorithms we use same population size which amounts to 40 particles or bacteria in corresponding algorithm. To make the comparison fair population all the competitor algorithms (all problems tested) were initialized using the same random seed.

We choose the number of fitness function evaluations (FEs) as a measure of the computational time instead of 'iterations' or 'generations'. Twenty-five independent runs of the four competitor algorithms were carried out on each problem and the average of the best -of- run solutions and standard deviations were noted. Each run was continued for different maximum number of FEs depending on the complexity of the problem. The spread spectrum radar poly-phase code design problem was tested varying n from 2 to 20. We, however, report result of just two of the most difficult problem instances (for dimensions 19 and 20) owing to the space limitations. Standard set of parameters was used for the PSO algorithm and the original BFOA. In case of the BSO algorithm we have chosen the best-suited set of parameters after a series of hand tuning experiments. We take $N_{re}=4$, $N_c=50$, $\omega=0.8$, $C_1=C_2=1.494$. The same set of parameters was used for all algorithms. For BFOA and MPSO-TVAC, we have employed the standard set of parameter values as recommended in [1] and [13] respectively.

4 Results

Table 2 compares the algorithms on quality of the optimum solution over five benchmarks. The mean and the standard deviation (within parenthesis) of the

Table 2. Mean and Standard Deviation over five benchmarks

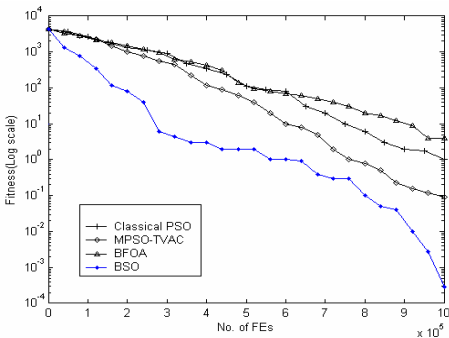
Fun	Dim	Max ^m FE	Mean Best Value (Standard Deviation)			
			BFOA	Classical PSO	MPSO- TVAC	BSO
f ₁	15	50,000	26.705 (2.162)	14.225 (3.573)	4.217 (1.332)	0.483 (0.074)
	30	1×10 ⁵	58.216 (14.32)	46.139 (9.649)	22.432 (7.178)	15.471 (2.655)
	45	5×10 ⁵	96.873 (26.136)	83.630 (14.536)	43.258 (16.944)	27.986 (4.338)
	60	1×10 ⁶	154.705 (40.162)	122.239 (67.728)	97.537 (24.379)	52.263 (8.341)
f ₂	15	50,000	6.9285 (2.092)	3.3484 (0.297)	1.1545 (0.321)	0.082 (0.00928)
	30	1×10 ⁵	17.0388 (4.821)	12.7374 (0.781)	9.8824 (0.931)	10.2266 (0.238)
	45	5×10 ⁵	30.9925 (7.829)	24.8286 (1.818)	17.0656 (1.352)	13.5034 (3.923)
	60	1×10 ⁶	45.8234 (9.621)	36.3343 (6.291)	22.253 (4.889)	18.3621 (5.773)
f ₃	15	50,000	0.2812 (0.0216)	0.0361 (0.00524)	0.1613 (0.097)	0.0541 (0.0287)
	30	1×10 ⁵	0.3729 (0.046)	0.1348 (0.107)	0.2583 (0.1232)	0.0792 (0.0113)
	45	5×10 ⁵	0.6351 (0.052)	0.1969 (0.116)	0.5678 (0.236)	0.1352 (0.0135)
	60	1×10 ⁶	0.8324 (0.076)	0.7584 (0.342)	0.6113 (0.097)	0.2547 (0.0287)
f ₄	15	50,000	0.9332 (0.0287)	0.5821 (0.0542)	0.1696 (0.0026)	0.0825 (0.0007)
	30	1×10 ⁵	4.3243 (1.883)	0.8578 (0.042)	0.7372 (0.0415)	0.5921 (0.036)
	45	5×10 ⁵	12.4564 (3.434)	1.8981 (0.195)	0.8922 (0.1453)	0.9383 (0.1327)
	60	1×10 ⁶	8.3247 (1.613)	2.4062 (0.451)	2.1692 (0.418)	1.8766 (0.536)
f ₆	2	50,000	0.999868 (0.00217)	0.999832 (0.00167)	0.999805 (0.00485)	0.999800 (0.0000)

best-of-run values of 25 independent runs for each of the four algorithms were presented. Each algorithm is predetermined maximum number of FEs. The best solution in each case has been marked in bold. Table 3 presents results of the unpaired *t*-tests between BSO and best of the three competitive algorithms in each case (standard error of difference of the two means, 95% confidence interval of this difference, the *t* and the two tailed P value). In table 3 for all cases the sample size =25 and degrees of freedom = 48. It is interesting to note from table 2 and 3 that for most of the cases the BSO algorithm meets or beats its nearest competitor in a statistically meaningful way. Table 2 shows that in three cases (f₂ (30), f₃ (15), f₄ (45)) the mean of the BSO algorithm is greater than that of the classical PSO or MPSO-TVAC. But table 3 reveals that in two cases i.e. f₂ (30) and f₄ (45) this difference is not statistically significant. Also one may perceive that incorporating the *g*_best PSO

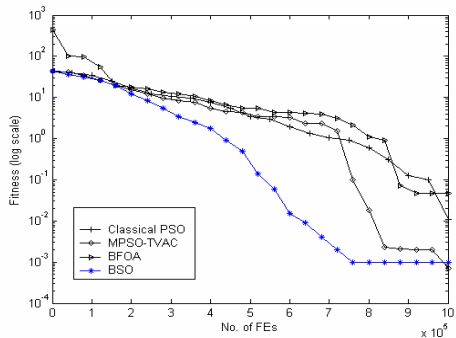
operator besides the computational chemotaxis has significantly improved the performance of BSO as compared to the original BFOA algorithm. Tables 4 and 5 present the corresponding results for the radar poly-phase code design problem for $n=19$ and $n=20$. In figure 1 we have graphically presented the rate of convergence of the competitor algorithms for all the functions in 30 dimensions. The graphs have been drawn for the median of the run for all cases. Figure 2 shows the scalability of the four methods on two tests functions-how the average time of convergence varies with the dimensionality of the search space. We omit rest of test functions for the sake of space economy. The graphs suggest that the effects of the *curse of dimensionality* are comparable on BSO and MPSO-TVAC and at least far better than classical PSO and BFOA.

Table 3. Results of unpaired t-tests on the data of Table 3

Fn, Dim	Std. Err	<i>t</i>	95% Conf. Intvl	Two-tailed <i>P</i>	Significance
$f_1, 15$	0.267	13.9949	-4.27046 to -3.19754	< 0.0001	Extremely significant
$f_1, 30$	1.531	4.5477	-10.03859 to -3.88341	< 0.0001	Extremely significant
$f_1, 45$	3.498	4.3658	-22.30540 to -8.23860	< 0.0001	Extremely significant
$f_1, 60$	5.153	8.7855	-55.63537 to -34.91263	< 0.0001	Extremely significant
$f_2, 15$	0.064	16.6908	-1.2011367 to -0.9428633	<0.0001	Extremely Significant
$f_2, 30$	0.192	1.7899	-0.04242 to 0.73042	0.0798	Not quite Significant
$f_2, 45$	1.513	14.4685	-24.9330 to -18.8487	< 0.0001	Extremely significant
$f_2, 60$	1.513	2.5724	-6.932087 to -0.849713	0.0132	Significant
$f_3, 15$	0.006	3.0849	0.0062682 to 0.0297318	0.0034	Very Significant
$f_3, 30$	0.022	2.5838	0.012333 to 0.098867	0.0129	Significant
$f_3, 45$	0.023	2.6417	-0.108662 to -0.014738	0.0111	Significant
$f_3, 60$	0.020	17.6261	-0.3972779 to -0.3159221	<0.0001	Extremely Significant
$f_4, 15$	0.001	161.740	-0.0881827 to -0.086017	<0.0001	Extremely Significant
$f_4, 30$	0.011	13.2057	-0.1671923 to -0.1230077	<0.0001	Extremely Significant
$f_4, 45$	0.030	0.8735	-0.033979 to 0.086179	0.3867	Not Significant
$f_4, 60$	0.136	2.1524	-0.565934 to -0.019266	0.0364	Significant
$f_5, 2$	0.001	0.0052	-0.0019553 to 0.0019453	0.9959	Not Significant



(a) Rastrigin (f_2)



(b) Ackley (f_4)

Fig. 1. Progress towards the optima for two benchmark functions

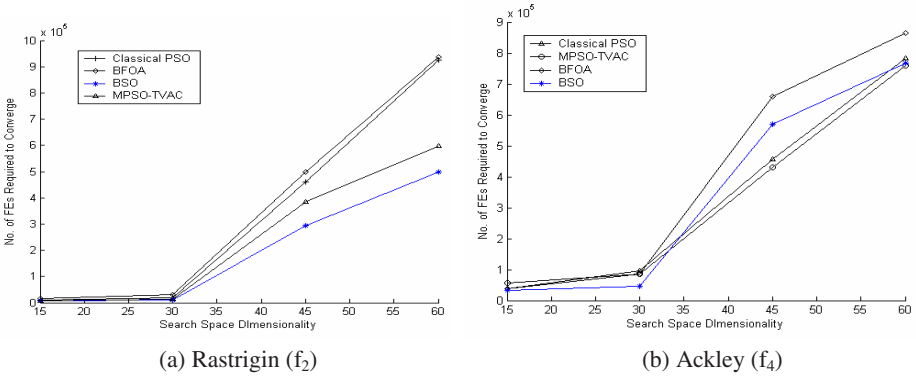


Fig. 2. Variation of computational cost with search space dimensionality

Table 4. Average and the standard deviation of the best-of-run solution for 25 runs for spread spectrum radar poly-phase code design problem (number of dimensions $n = 19$ and $n = 20$). For all cases each algorithm was run for 50,000 FEs.

N	Mean best-of-run solution (Std Dev)			
	BFOA	Classical PSO	MPSO-TVAC	BSO
19	0.7974 (0.0323)	0.7524 (0.00493)	0.7832 (0.00038)	0.7519 (0.0392)
20	0.8577 (0.0283)	0.8693 (0.0048)	0.8398 (0.0482)	0.8134 (0.0482)

Table 5. Results of unpaired t-tests on the data of Table 4

n	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
19	0.002	2.5024	-0.00734 to -0.00081	0.0152	Significant
20	0.010	3.5880	-0.05792 to -0.01644	0.0007	Extremely significant

5 Conclusions

The paper has presented an improved variant of the BFOA algorithm by combining the PSO based mutation operator with bacterial chemotaxis. The present scheme attempts to make a judicious use of exploration and exploitation abilities of the search space and therefore likely to avoid false and premature convergence in many cases. The overall performance of the proposed algorithm is definitely better than a standalone BFOA at least on the numerical benchmarks tested. The performance also appears to be at least comparable with PSO and its variants. The future research effort should focus on reducing the number of user-defined parameters for BFOA and its variants. Also an empirical study on the effects of these parameters on the convergence behavior of the hybrid algorithms may be worthy to undertake.

References

1. Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control, *IEEE Control Systems Magazine*, 52-67, (2002).
2. Mishra, S.: A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Trans. on Evolutionary Computation*, vol. 9(1): 61-73, (2005).
3. Tripathy, M., Mishra, S., Lai, L.L. and Zhang, Q.P.: Transmission Loss Reduction Based on FACTS and Bacteria Foraging Algorithm. *PPSN*, 222-231, (2006).
4. Kim, D.H., Cho, C. H.: Bacterial Foraging Based Neural Network Fuzzy Learning. *IICAI 2005*, 2030-2036.
5. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Harbor (1975).
6. Kennedy, J, Eberhart, R.: Particle swarm optimization, In *Proceedings of IEEE International Conference on Neural Networks*, (1995) 1942-1948.
7. Storn, R., Price, K.: Differential evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, 11(4) 341–359, (1997).
8. Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization, *Information Sciences*, Vol. 177 (18), 3918-3937, (2007).
9. Mladenovic, P., Kovacevic-Vujicic, C.: Solving spread-spectrum radar polyphase code design problem by tabu search and variable neighborhood search, *European Journal of Operational Research*, 153(2003) 389-399.
10. Stephens, D.W., Krebs, J.R., *Foraging Theory*, Princeton University Press, Princeton, New Jersey, (1986).
11. Yao, X., Liu, Y., Lin, G. Evolutionary programming made faster, *IEEE Transactions on Evolutionary Computation*, vol 3, No 2, 82-102, (1999).
12. Angeline, P. J.: Evolutionary optimization versus particle swarm optimization: Philosophy and the performance difference, *Lecture Notes in Computer Science* (vol. 1447), *Proceedings of 7th International Conference on Evolutionary Programming – Evolutionary Programming VII* (1998) 84-89.
13. Ratnaweera, A., Halgamuge, K.S.: Self organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, In *IEEE Transactions on Evolutionary Computation* 8(3): 240-254, (2004).

Bayes-Based Relevance Feedback Method for CBIR*

Zhiping Shi, Qing He, and Zhongzhi Shi

Key Laboratory of Intelligent Information Processing, CAS, Beijing 100080 China
{shizp, heq, shizz}@ics.ict.ac.cn

Abstract. The paper proposes a Bayes-based relevance feedback approach integrating visual features and semantics for content-based image retrieval systems. The data of the image database are divided into small clusters by semantic supervised clustering algorithm. The cluster here is called as index cluster. So the data of each index cluster are similar both in visual features and in semantics. During relevance feedback process, users sign the positive and negative examples regarded a cluster as unit rather than a single image, and each feedback cluster construct a Bayesian classifier on visual features; and the semantic classes of the feedback examples construct the Bayesian classifier on semantics. At last, we use Bayesian classifiers on visual features and semantics respectively to adjust retrieval similarity distance. Our experiments on an image database show that a few cycles of relevance feedback by the proposed approach can significantly improve the retrieval precision.

Keywords: Relevance feedback; image retrieval; cluster; Bayes.

1 Introduction

The Content-Based Image Retrieval (CBIR) systems query images by example image commonly, and return k -nearest neighbors (k -NN) in terms of the visual feature similarity. The visual features of the example image are used to represent the retrieval semantics actually. It is supposed that the more similar are the visual features of the images, the closer are the semantics of the images. However, because of subjectivity of the users' perception and semantic gap, it is difficult to construct the direct map between the visual features and the semantics of images.

Relevance feedback is a hybrid technology of information retrieval and machine learning to deal with the relation between the visual features and the semantics of images through human-machine interaction. On relevance feedback, one side query vector is moved close to the center of the positive examples and far from negative examples, on the other side, the parameters of the Gaussian models are optimized to represent the distribution of positive examples more exactly [1-4]. The common default of these methods is that they are based on an idealist hypothesis: positive examples distribution follows a Gaussian model [4]. The real distribution of the visual

* This paper is supported by the National Science Foundation of China (No. 60435010, 90604017, 60675010), 863 National High-Tech Program of China (No.2006AA01Z128), National Basic Research Priorities Programme of China (No. 2003CB317004, 2007CB311004) and the Nature Science Foundation of Beijing, China (No. 4052025).

features in a semantic class in an image database is very complicated. The visual features in a semantic class maybe not follow a single Gaussian distribution but a mixture Gaussian distribution with several Gaussian components [5]. In most of the traditional relevance feedback techniques, only analysis in the visual feature space is considered, but the semantic class information in image databases is ignored. In actually, the semantic class information plays an unfungible role in image retrieval. The images with similar visual features may be different semantic classes. At this station, relevance feedback on the visual features can't classify them but relevance feedback on the semantic class information can do easily.

The paper proposes a novel Bayes-based relevance feedback approach integrating the visual features and the semantics for CBIR. We assume that both the positive feedback space and the negative feedback space follow the Gaussian mixture distribution so we can estimate the complexity distribution of query space more accurately. The feedback examples are expanded to their index clusters. The Bayesian classifiers created by the index clusters are applied to rectify similarity distance. On the other hand, the positive and negative semantic space can be estimated by feedback examples' semantic information. The probability of each database image belonging to the positive or negative semantic space can be estimated to rectify the similarity distance. As a result, the retrieval results by feedback are more consistent with the semantics.

2 Semantics Supervised Cluster Based Index and Retrieval

In practice, there are two kinds of indexes in the image databases: 1) the semantic class, 2) the visual features. Intuitively, it is reasonable to develop techniques that combine the advantages of both semantics and visual feature index. We have presented a semantics supervised cluster based index approach (briefly as SSCI) and the corresponding retrieval approach [5]. The outline of our SSCI is as follows: The entire data set is divided hierarchically by a modified clustering technique into many clusters until the objects within a cluster are not only close in the visual feature space but also within the same semantic class, and then an index entry including semantic clue and visual feature clue is built for each cluster. So the SSCI-based nearest-neighbor (NN) search can be divided into two phases: the first phase computes the distances between the query example and each cluster index and returns the clusters with the smallest distance, here namely candidate clusters; then the second phase retrieves the original feature vectors within the candidate clusters to gain the approximate nearest neighbors. The main character of the technique is that it distinctly improves the semantic precision as increasing the search speed for CBIR.

Because the cluster algorithm performs in the whole image database, the clusters reflect the distribution of the visual features in the image database. Each cluster represents a narrow Gaussian distribution in which the images with the similar visual features and the same semantic class lie. Each semantic class follows the mixture Gaussian distribution with the several cluster components with the same semantic class.

3 Relevance Feedback Based on the Bayesian Classifier

In this section, we propose the relevance feedback in visual features and semantics respectively.

3.1 Relevance Feedback in Visual Features

During the relevance feedback process, all images in a cluster are regarded as feedback examples if one image in the cluster is marked feedback example. This can make up the insufficiency of feedback examples. The index clusters are the Gaussian components with the small variances. To say, there are many positive Gaussian components and negative Gaussian components near the query example in the visual feature space.

The posterior probability for the positive and/or negative Gaussian components membership of each image can be estimated and the distance of the image is amended by the margin between the posterior probability for the positive Gaussian components membership and that for the negative Gaussian components membership. Let \mathbf{x} be a sample of the image database, and c_j be an index cluster in which a feedback example has been marked. Because c_j follows a narrow Gaussian distribution, the likelihood of \mathbf{x} is given by

$$p(\mathbf{x} | c_j) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_j|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)}. \quad (1)$$

Under the Bayes rules, the determining function for \mathbf{x} belongs to c_j is given by

$$\begin{aligned} g_j(\mathbf{x}) &= \log p(c_j | \mathbf{x}) \\ &\propto \log p(\mathbf{x} | c_j) + \log p(c_j) \\ &= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j) - \frac{d}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}_j| + \log p(c_j). \end{aligned} \quad (2)$$

Let \mathbf{q} be the query example, \mathbf{x} be a database sample. Allowing for both the positive Gaussian distribution and the negative Gaussian distribution, the feedback function is given by

$$\text{sim}^*(\mathbf{q}, \mathbf{x}) = \text{sim}(\mathbf{q}, \mathbf{x}) + \sum_{j \in +} g_j(\mathbf{x}) - \sum_{j \in -} g_j(\mathbf{x}). \quad (3)$$

Wherein +, - stand for positive example set and negative example set respectively, $\text{sim}(\mathbf{q}, \mathbf{x})$, which maybe the Euclidean distance or the intersection of the histograms commonly, stands for the similarity distance between \mathbf{x} and \mathbf{q} .

Allow for the feedback examples marked by users are few, and the number of the positive examples is not equal to that of the negative ones, they have only local representative. It maybe bring out uncertain errors to mix all the feedback examples to represents the distribution of the positive or negative samples. As shown in fig.1, “+”, “-” represent “is” or “is not” a query target image, and those enclosed by circles represent the feedback examples marked by the user. A query target image X is

nearest from a positive example, and there are more negative examples marked by the user. If accumulating the X ' posterior probabilities for the positive Gaussian components membership and that for the negative Gaussian components membership, the effect of the negative feedback is strengthened, so X is classified a negative sample in error.

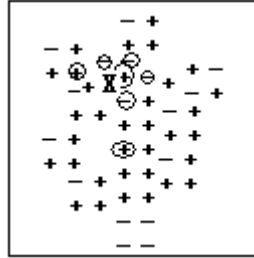


Fig. 1. The distribution of the positive and negative samples

Therefore, we take into account only the nearest positive example and the nearest negative example. After getting feedback information, the retrieval system calculates every image's posterior probability for the positive Gaussian models and the negative Gaussian models. Then the margin between the maximum posterior probability for the positive Gaussian models and the maximum one for the negative Gaussian models is used to adjust the distance of the image features.

The distance function between \mathbf{x} and \mathbf{q} modified by visual features feedback is given by

$$sim^*(\mathbf{q}, \mathbf{x}) = w_0 \times sim(\mathbf{q}, \mathbf{x}) + w_1 \times \frac{\max_{j \in +} g_j(\mathbf{x}) - \max_{j \in -} g_j(\mathbf{x})}{\left| \max_{j \in +} g_j(\mathbf{x}) + \max_{j \in -} g_j(\mathbf{x}) \right|}. \tag{4}$$

Wherein $\max_{j \in +} g_j(\mathbf{x})$ is the maximum $g_j(\mathbf{x})$ for the positive examples set,

$\max_{j \in -} g_j(\mathbf{x})$ is the maximum $g_j(\mathbf{x})$ for the negative examples set,

$\left| \max_{j \in +} g_j(\mathbf{x}) + \max_{j \in -} g_j(\mathbf{x}) \right|$ is the normalized factor, w_0, w_1 are weight factors.

4 Relevance Feedback in Semantics

The image query results that are retrieved according to the visual feature similarity are with the similar visual features but not with the same semantics. The user's query intention points to the semantic similar images. The user's query semantics maybe not correspond to a semantic class in the image database; however the semantic classes in the image database can be the important clue to optimize the query results. Although the relation of the semantic concepts is complicated, it is always that some semantic

concepts are consistent and some others are repulsive. It is difficult task to figure out the relation of the semantic concepts directly. It is feasible to learn the consistent or repulsive relation between the user query semantics and the semantic classes of the image database by using the positive and negative feedback examples.

Intuitively, the more the number of the images of a semantic class are marked as positive examples, the higher the probability of the semantic class being consistent with the query target, vice versa, the more the number of the images of a semantic class are marked as negative examples, the higher the probability of the semantic class being repulsive with the query target. So the proposed optimizing method is as follows:

1. Computing the posterior probability of the positive example semantic classes belonging to the query semantics.

$$p(S_q | S_i) = \frac{p(S_i | S_q) p(S_q)}{p(S_i | S_q) p(S_q) + p(S_i | S^-) p(S^-)}. \quad (5)$$

Wherein S_i stands for the semantic class i , S_q stands for the query semantics, i.e., the positive feedback semantic space, $p(S_q)$ stands for the probability of the positive feedback examples, S^- stands for the negative feedback semantic space, $p(S^-)$ stands for the probability of the negative feedback examples.

$$p(S_i | S_q) = \frac{p_i}{p}. \quad (6)$$

$$p(S_q) = \frac{p}{p+n}. \quad (7)$$

$$p(S^-) = 1 - p(S_q). \quad (8)$$

Wherein p_i stands for the number of the positive examples of the semantic class i , p stands for the total number of the positive examples, n stands for the total number of the negative examples.

So the probability for the query semantics membership of the sample x is given by

$$p(S_q | \mathbf{x}) = \sum_i p(S_q | S_i) p(S_i | \mathbf{x}). \quad (9)$$

2. Computing the probability for the negative feedback semantic space membership of the sample x :

$$p(S^- | \mathbf{x}) = \sum_i p(S^- | S_i) p(S_i | \mathbf{x}). \quad (10)$$

$$p(S^- | S_i) = \frac{p(S_i | S^-) p(S^-)}{p(S_i | S_q) p(S_q) + p(S_i | S^-) p(S^-)}. \quad (11)$$

$$p(S_i | S^-) = \frac{n_i}{n}. \quad (12)$$

Wherein n_i stands for the number of the negative examples of the semantic class i
 3. Updating the images similarity distance:

$$sim'(q, x) = w_2 \times sim^*(q, x) + w_3 \times p(S_q | x) - w_4 \times p(S^- | x). \quad (13)$$

Wherein w_2, w_3, w_4 are weight factors.

The probability for the semantic class i membership of x , $p(S_i | x)$, can be calculated by the diverse method. For example, the method from our another paper [5] is given by

$$p(s_i | x) = \sum_j p(s_i | c_j) p(c_j | x). \quad (14)$$

Wherein c_j stands for an index cluster.

Another method is that the sample x is a member of the semantic class S_i or not determinately:

$$p(S_i | x) = \begin{cases} 1 & \mathbf{x} \in S_i \\ 0 & \mathbf{x} \notin S_i \end{cases}. \quad (15)$$

For the simpleness, the latter is adopted in the paper experiments.

5 Evaluation

In this section, we present experiments on an image database to show the proposed method performance.

The image database includes 8342 images from the Internet and Corel Image Gallery, and they are classified into 23 semantic classes. The visual feature extracted for each image is the 36-dimension HSV histogram. The weight factors w_0, \dots, w_4 are assigned 0.6, 0.4, 0.4, 0.3, 0.3 ordinally. Fig.2 shows the progress of retrieving traditional Chinese paintings by a traditional Chinese painting example. The first retrieval results, which are retrieved by the visual feature, include many human photos because a man is presented on the query example image. Only after 4 cycles relevance feedbacks, the most of retrieval result images are traditional Chinese paintings.

In this paper, precision is used to measure the performance. In our experiments, an object of result set is correct if and only if the object belongs the same semantic class with the query example. The precision P is defined by $P = c/t$. Where c is the number of the correct result objects, and t is the number of the all result objects. We chose 4 images from the image database randomly as query example images to do k -NN retrieval experiments. Fig. 3 illustrates the precision of the proposed method, wherein $k=90$, each curve stands for a query session. The precisions of the 4 original



a) Original retrieval results



b) The results after 1st feedback



c) The results after 2nd feedback



d) The results after 3rd feedback



e) The results after 4th feedback

Fig. 2. The progress of retrieving traditional Chinese paintings

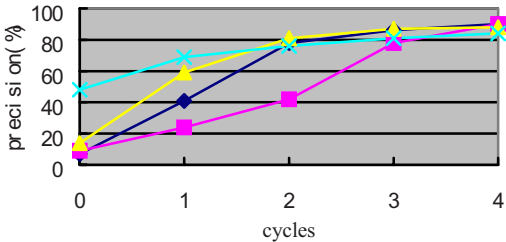


Fig. 3. The precision curves of k-NN image retrieval

results (0 cycles) are not good (7, 9, 14, 48 respectively), but the precisions rise to above 84~90 quickly after 4 cycles feedback. The speed of optimizing precision by the proposed method is higher than the other current approaches because our method integrates the visual features and semantics information.

6 Conclusions

The paper proposed a Bayes-based relevance feedback method integrating the visual features and the semantics for CBIR. We argue that both the positive examples and the negative ones follow the Gaussian mixture model with several Gaussian components. To offset the insufficient of the feedback examples marked by users, the whole index cluster is marked as a feedback example if one image of it is marked. The query semantic space can be estimated by the semantic classes of the positive and negative index clusters because each index cluster belongs to a certain semantic class. So, the relevance feedback rises to the semantics level. Thanks to integrating the visual features and the semantics, the proposed method can optimize the semantic precision with a few cycles feedbacks.

References

1. Rui Y., Huang T.S., Mehrotra S.: Content-Based image retrieval with relevance feedback in MARS. Proceedings of IEEE international conference on image processing. New York, (1997)815–818
2. Rui Y., Huang T.S.: Optimizing Learning in Image Retrieval. Proc. of IEEE Int. Conf. On Computer Vision and Pattern Recognition, Hilton Head, SC, (2000)236–243
3. Cox I.J., Miller M.L., Minda T.P., et al.: The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. IEEE Transactions on Image Proceeding, Special Issue On Image and Video Proceeding for Digital Libraries, Vol.9(1), (2000)20–37
4. Su Zhong, Zhang Hongjiang, Ma Shaoping.: An Image Retrieval Relevance Feedback Algorithm Based on the Bayesian Classifier. Journal of Software. Vol.13(10), (2002)2001–2006 (in Chinese)
5. Shi Zhiping, Li Qingyong, Shi Zhiwei, et al.: Semantics supervised cluster-based index for video databases. In: Sundaram H., Naphade M.R., Smith J.R., Rui Y. (Eds.): Image and Video Retrieval. Lecture Notes in Computer Science, Vol.4071, Springer, (2006)453–462

A Novel Hierarchical Block Image Retrieval Scheme Based Invariant Features*

Mingxin Zhang^{1,2}, Zhaogan Lu¹, and Junyi Shen¹

¹ Dept. Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, P.R. China

² College of Mathematics and Information Science, Northwest Normal University, Lanzhou 730070 China
zhangmx@mail.xjtu.edu.cn, luzhaogan@mailst.xjtu.edu.cn

Abstract. Image retrieval is generally implemented by image matching or based-regions retrieval, but it's difficult to balance retrieval performance and complexity. Query images may appear with different scales and rotations in different images, so a hierarchical image segmentation is proposed to partition the retrieved images into equal blocks with different sizes at different levels. Then, the similar metrics of these sub-blocks to query image, are evaluated to retrieve those sub-blocks with contents in query images. Meanwhile, information about scales and locations of query objects in retrieved images can also be returned. The hierarchical block image retrieval schemes with geometric invariants, normalized histograms and their combinations are tested by experiments via a database with 500 images, respectively. The retrieval accuracy with geometric invariants as invariant features can achieve 78% for the optimal similar metric threshold. Furthermore, the scheme can also work with different size images.

Keywords: image retrieval, geometric invariants, normalized histogram, hierarchical image segmentation.

1 Introduction

Content-based image retrieval (CBIR) is a set of techniques for retrieving relevant images from a database on the basis of automatically derived features, which accurately specify the information content of each image. Furthermore, the information content of one image, are generally presented by typical low-level image features, such as color, shape, and texture, and so on. However, it is very difficult to separate different objects in one image and characterize them by their color, shape and texture features. So, many CBIR schemes ^[1,2] are currently implemented by searching the most similar images in image databases to one special query images according to color or texture features of these images other than the query objects contained the query images. When the query object only occupies one part of the query image, these CBIR schemes will lead to bad retrieval performance.

* Work supported by Gansu Natural Science Foundation of China under Grant No. 3ZS051-A25-047.

Alternatively, region-based image retrievals are contributed by many recent literatures^[3,4], which can obtain the objects in images and its locations, but it will become impossible for those images with complex backgrounds by reason of the limits in image segmentation. Therefore, how to effectively retrieve images with specific objects is still a challenge for CBIR in applications.

In this paper, a hierarchical block image retrieval (HBIR) scheme is proposed to retrieve those images containing special objects, where geometric invariants^[5] and normalized histogram^[6] are used as features. According to the scheme, in order to find the objects in an image with different scales and rotations, the image will be equally segmented into sub-blocks with different size at pyramid hierarchical levels. Then, the similar metrics of these sub-blocks to query objects are evaluated to find those sub-blocks whose similar metrics superior to a specified threshold. Scales and locations of query objects in the retrieved images can also be figure out at pyramid hierarchical levels.

2 Scale and Rotation Invariant Features

Objects in an image may present different scales and rotations from its query counterparts, so the required features have to be invariant to translation, scaling, and orientation. Many methods have been proposed to extract image invariants^[7-9], such as zernike moments, rotation invariant Gabor and complex wavelet transform, rotation and scale invariant log-polar wavelet feature, discrete fourier transform moments, and so on. However, the order of zernike moments as image features, badly depends its image reconstruction accuracy, which will change with image contents. Other approaches involve log-polar transform, Gabor and wavelet transform, and invariant features can be exacted after complex image preprocessing. So, we adopt geometric invariants^[5] and normalized histogram^[6] as image features, which have simple implementation without image preprocessing.

2.1 Geometric Invariants

According to the results in literature^[8], two sets of moment invariants are derived on the base of geometric moments as showed above. The first set is invariant to orthogonal transformations, while the second is invariant to general affine transformation. There are seven functions invariant to orthogonal transformations, i.e., ϕ_1, \dots, ϕ_7 , and four invariant functions invariant to general affine transformation, such as $\varphi_1, \dots, \varphi_4$.

In this paper, the two sets of moment invariants will be used as image invariant features, and the feature vector can be described as

$$\mathbf{M} = [\phi_1, \dots, \phi_7, \varphi_1, \dots, \varphi_4]^T,$$

Where $(\cdot)^T$ represents matrix transpose.

Furthermore, if \mathbf{M}_0 denotes the feature vector of query image, the dissimilarity measure between query image and its queried image is given by Canberra distance metric between their feature vectors, i.e.,

$$D_m(\mathbf{M}_o, \mathbf{M}) = \sum_{i=1}^{11} \frac{|\mathbf{M}_o(i) - \mathbf{M}(i)|}{|\mathbf{M}_o(i)| + |\mathbf{M}(i)|} \tag{1}$$

where $\mathbf{M}_o(i)$ and $\mathbf{M}(i)$ denote the i -th element of \mathbf{M}_o and \mathbf{M} , respectively.

According to this formula, the Canberra distance metric is normalized between 0 and 1. For color images, its moment invariants are implemented for red, green and blue components, and the correspondent dissimilarity measures are denoted as $D_m(r)$, $D_m(g)$, and $D_m(b)$. Then, their average is used as the dissimilarity measures of color images, that is

$$D_m = \frac{D_m(r) + D_m(g) + D_m(b)}{3} \tag{2}$$

2.2 Normalized Histogram

Suppose that $H_o(i)$ describes the normalized histogram of query image with 256 grayscales, then the similar metric between histograms of query image and queried image, is given as the maximal absolute value of their cross-correlation function, i.e.,

$$D_h = \max_{m=-255, \dots, 0, \dots, 255} |R_{H_o, H}(m)| \tag{3}$$

where their cross-correlation functions $R_{H_o, H}(m)$ is defined as

$$R_{H_o, H}(m) = \begin{cases} \frac{1}{E(N-|m|)} \sum_{i=0}^{255} H_o(i+m)H(i) & m \geq 0 \\ R_{H_o, H}^*(-m) & m < 0 \end{cases}$$

while E is a factor to normalize the cross-correlation functions at zero lag to 1, i.e.,

$$E = \sqrt{\sum_{i=0}^{255} [H_o(i)]^2} \cdot \sqrt{\sum_{i=0}^{255} [H(i)]^2}$$

As the cross-correlation function can cancel the effect of intensity on histogram, it is a more practical similar metric than histogram intersection^[14]. Moreover, according to above formula, the cross-correlations functions at zero lag is normalized to the range of [0,1], so the histogram similar metric is also normalized to the range of [0,1].

Similarly, the histogram similar metric for color images, is obtained by averaging the histogram similar metrics of its red, green and blue components, which are presented as $D_h(r)$, $D_h(g)$ and $D_h(b)$, respectively. That is

$$D_h = \frac{D_h(r) + D_h(g) + D_h(b)}{3} \tag{4}$$

2.3 Similarity Metric

If geometric invariants and normalized histogram are integrated, the similar metric of color images is normalized to [0,1] by the linear combination of that for geometric invariants and normalized histogram, i.e.,

$$D = \frac{D_m + (1 - D_h)}{2} \quad (5)$$

It can also be referred as the similar distance for color images when their geometric invariants and normalized histogram are synthetically considered. According to the similar metric, the images containing specific contents can be found by checking similar metric D whether smaller than a given similar metric threshold.

3 Hierarchical Block Image Retrieval

Image retrieval is generally conducted by searching the most similar images from databases to the query image, or region-based retrieval according to image segmentation results. The former has small complexity but bad retrieval accuracy, while the latter has good retrieval accuracy but great complexity. In order to balance computational complexity and retrieval accuracy, we use a flexible strategy for progressive image retrieval.

According to HBIR, an image is firstly segmented into four equal size blocks, and then each block is further segmented into four equal size sub-blocks. The same operation is repeated N times, where N is called the levels of HBIR, thus the image is segmented into 4^N sub-blocks at N level HBIR. A pyramid hierarchical sub-block process is executed to segment images into different size blocks, and the extraction of scale and rotation invariant features for all these sub-blocks is followed. Once the similar metric of these sub-blocks with the query objects are calculated, those sub-blocks matching query objects will be retrieved with their scale and location information.

For given similarity metric threshold T_o , the N level HBIR scheme can be formulated as following algorithm:

- (1) Extract geometric invariants M_o and normalized histogram $H_o(i)$ of query image.
- (2) Calculate similar metrics of different sub-blocks
 - For n = 0:N
 - Segment image into 4^n sub-blocks
 - For m = 0: $4^n - 1$
 - (a) calculate the center coordinates $P(n,m)$ of sub-block $B(n,m)$.
 - (b) extract geometric invariants $M(n,m)$ and normalized histogram $H(n,m)$.
 - (c) calculate similarity metrics $s(n,m)$ between sub-blocks and query image
 - End
- (3) Find those sub-blocks whose similarity metrics $s(n,m) \leq T_o$.
 - If number of matching sub-blocks ≥ 1
 - Retrieve the image and mark the locations of these sub-blocks.
 - Retrieval succeeds
 - Else
 - Retrieval fails.
 - End
- (4) Image retrieval is over.

4 Experimental Results and Discussion

To check the retrieval efficiency of proposed HBIR scheme, a test image database with 500 different size color images is constructed, which includes various types of images like flowers, birds, natural scenes, cars, etc. Fig.1 shows some image samples from the test databases with different sizes and categories.



Fig. 1. Sample images from test image database

The retrieval performance is measured by two performance metrics, i.e., retrieval accuracy and retrieval rate. Retrieval accuracy is defined as the ratio of the number of retrieved relevant images to the total number of retrieved images, while the retrieval rate for the query image is measured by the ratio of the number of retrieved relevant images to the total number of query images in databases. They are mainly effected by similar metric thresholds and the levels of HBIR, i.e., unlike the retrieval accuracy, the retrieval rate will inversely change with similar metric threshold. Consequently, the optimal similar metric threshold can be determined by the balance of retrieval accuracy and retrieval rate through test experiments. When geometric invariants and normalized histogram are used as features to conduct the HBIR image retrieval scheme, their changes with similar metric threshold is displayed in Fig.2, and the threshold correspondent to their intersection is showed to be 0.175, which can balance retrieval accuracy and retrieval rate. The threshold is called the optimal similar metric threshold, whose correspondent retrieval accuracy and retrieval rate is evaluated to test the retrieval performance of HBIR image retrieval.

When we considered the first image in Fig.3 as a query image, the HBIR image retrieval scheme was conducted to find those images with the contents in the query image. Some most matching images were showed in Fig.3 as examples to test the retrieval results for the HBIR scheme, where the locations of the symbols, i.e., '+', 'o', '*' and 'x', are the center coordinates of matching sub-blocks in retrieved images for the 0, 1, 2, 3 level HBIR schemes, respectively. Thus, the information of the query about its sizes and locations in retrieved images is showed clearly, and this is also unique for the HBIR schemes when compared with other common image retrieval schemes.

Subsequently, we evaluated the retrieval performance of the HBIR scheme at different levels, where geometric invariants, normalized histograms and their

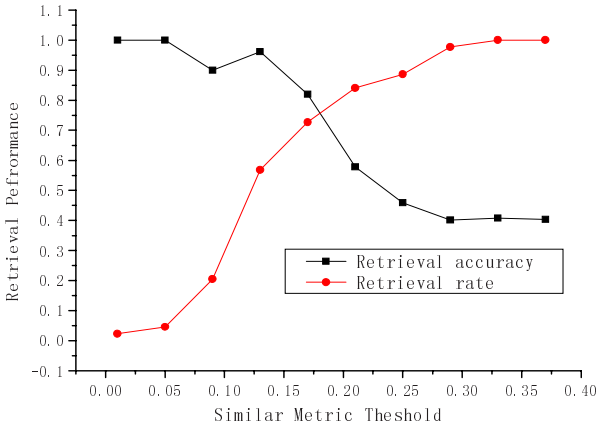


Fig. 2. The changes of retrieval accuracy and retrieval rate with similar metric threshold for 4 level HBIR image retrieval, when geometric invariants and normalized histograms are simultaneously used as features

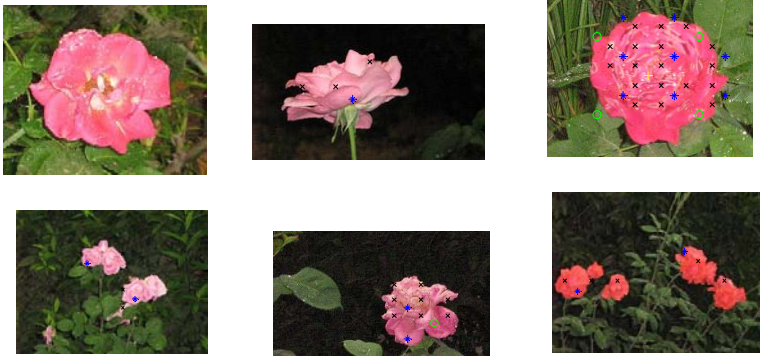


Fig. 3. Retrieval results of the first image via the HBIR image retrieval scheme with similar metric threshold 0.175, when geometric invariants and normalized histograms are used as features, while the sizes and locations of query images are also marked in retrieval results

combinations are used as features, respectively. In Fig. 4, the retrieval accuracy in three different cases and at different levels is showed, when the optimal similar metric threshold is given. The zero HBIR image retrieval scheme is actually the common image retrieval scheme based on whole image matching. From Fig. 4, it is clear that the retrieval accuracy with geometric invariants and normalized histograms as features is improved from 65% to 76%, while with geometric invariants from 40% to 78% and normalized histograms from 62% to 73%, respectively. Furthermore, at low levels, the retrieval performance of the scheme with geometric invariants is inferior to that of the schemes with normalized histograms and their combinations, while the latter has approximate retrieval accuracy. However, with the increase of HBIR level number, its retrieval accuracy is improved quickly. As the balance of the normalized

histogram scheme and the geometric invariant scheme, the HBIR image retrieval with their combinations at high levels can obtain better retrieval performance than that with normalized histograms but inferior to that with geometric invariants.

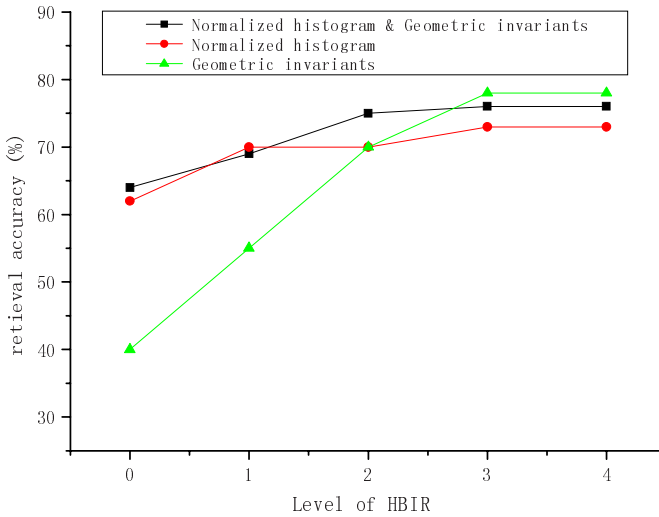


Fig. 4. Effect of the level number of HBIR image retrieval on retrieval performance of schemes with geometric invariants, normalized histograms and their combinations, respectively

5 Conclusion

In order to balance image retrieval accuracy and complexity, a feasible HBIR image retrieval scheme is proposed to conduct image retrieval with different size and rotation objects in images. It can be used a tradeoff between image retrieval based whole matching and that based interest-regions. The information about query objects in retrieved images with different sizes and locations can be returned via the scheme, which segments retrieved images into blocks with different sizes via a pyramid hierarchical block process. The scheme was tested with geometric invariants, normalized histograms, and their combinations as image invariant features, respectively. As showed by experiments, the retrieval accuracy for common image retrieval can be further improved by pyramid hierarchical block segmentation, the HBIR scheme is a feasible image retrieval method.

References

1. Tzagkarakis G., Beferull-Lozano B., Tsakalides P., "Rotation-invariant texture retrieval with gaussianized steerable pyramids", IEEE Trans. Image Processing, vol.15, no.9, pp:2702-2710, Sept. 2006.
2. Yap P.T., Paramesran R., "Content-based image retrieval using Legendre chromaticity distribution moments", IEE Proceedings of Vision, Image and Signal Processing, vol.153, no.1, pp:77-24, Feb. 2006.

3. Khanh V.H., Tavanapong K.A.W., "Image retrieval based on regions of interest", , IEEE Trans. Knowledge and Data Engineering, vol.15, no.4, pp:1045-1049, July-Aug. 2003.
4. Pratikakis I., Vanhamel I., Sahli H., Gatos B., Perantonis S.J., "Unsupervised watershed-driven region-based image retrieval", IEE Proceedings of Vision, Image and Signal Processing, vol. 153, no.3, pp:313-322, June 2006.
5. Masoud A., Ahmed H.T., "Geometric Invariance in Image Watermarking", IEEE Trans. Image Processing, vol.13, no.2, pp:145-153, Feb. 2004.
6. Jia Kebin, Fang Sheng, Zhu Qing, "Rotation and translation invariant color image retrieval", Proceedings of 6th International Conference on Signal Processing, vol.2, pp:1063-1066, Aug. 2002.
7. Dong-Gyu Sim, Hae-Kwang Kim, Dae-II Oh, "Translation, scale, and rotation invariant texture descriptor for texture-based image retrieval", Proceedings of 2000 International Conference on Image Processing, vol.3, pp:742-745, Sept. 2000.
8. Vasconcelos N., "On the complexity of probabilistic image retrieval", Proceedings of IEEE 8th International Conference on Computer Vision, vol.2, pp:400-407, July 2001.
9. Daekyu Shin, daewon Kim, Hyunsool Kim, et al, "An image retrieval technique using rotationally invariant Gabor features and a localization method", Proceedings of 2003 International Conference on Multimedia and Expo, vol.2, pp:701-704, July 2003.

A New Unsupervised Hybrid Classifier for Natural Textures in Images

Mará Guijarro ¹, Raquel Abreu², and Gonzalo Pajares¹

¹ Dept. Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense, 28040 Madrid, Spain
pajares@dacya.ucm.es

² Dept. Informática y Automática, E.T.S. Informática UNED, 28040 Madrid, Spain
raquel.abreuhernando@telefonica.es

Abstract. One objective for classifying textures in natural images is to achieve the best performance possible. Unsupervised techniques are suitable when no prior knowledge about the image content is available. The main drawback of unsupervised approaches is its worst performance as compared against supervised ones. We propose a new unsupervised hybrid approach based on two well-tested classifiers: Vector Quantization (VQ) and Fuzzy k -Means (FkM). The VQ unsupervised methods establishes an initial partition which is validated and improved through the supervised FkM. A comparative analysis is carried out against classical classifiers, verifying its performance.

Keywords: Vector Quantization, Fuzzy k -Means, unsupervised, image texture classification.

1 Introduction

Nowadays the increasing technology of aerial images is demanding solutions for different image-based applications. The natural texture classification is one of such applications due to the high spatial resolutions achieved in the images. The areas where textures are suitable include agricultural crop ordination, forest or urban identifications and damages evaluation in catastrophes or dynamic path planning during rescue missions or intervention services also in catastrophes (fires, floods).

Different classical techniques have been studied for image texture classification, namely: Bayesian, K-Nearest, Neural Networks, Vector Quantization [1], [2], [3], [4], [5]. These classifiers are supervised methods, which perform appropriately if a correct partition is used for training them. This implies that the patterns have been correctly assigned as belonging to each cluster. Unfortunately, the variety of textures in aerial images could become high or even if unpredictable. Hence, unsupervised automatic classification approaches should be suitable in order to establish the best cluster partition. The Vector Quantization approach (VQ) is one of the possible unsupervised approaches, the main drawback of VQ is its strong dependency from the threshold used for the partition and the order in which the patterns are processed.

We propose the use of the VQ classifier supervised by the Fuzzy k -Means (FkM) classifier. This hybrid scheme makes the main finding of this paper.

There are pixel-based and region-based approaches. A pixel-based approach tries to classify each pixel as belonging to one of the classes. The region-based identifies patterns of textures within the image and describes each pattern by applying filtering (laws masks, Gabor filters, Wavelets, etc.), it is assumed that each texture displays different levels of energy allowing its identification at different scales [6], [7], [8]. This is out of the scope of this paper.

The aerial images used in our experiments do not display texture patterns. This implies that textured regions cannot be identified by applying region-based. In this work we use a pixel-based approach under RGB color representation because it performs favorably against other color mappings, as reported in [7]. Hence, the three RGB spectral values are the features used in our method. The same texture could display different RGB levels; this makes the problem fuzzy in nature, justifying the choice of the FkM approach.

The paper is organized as follows. Section 2 describes the design of the automatic hybrid classifier, where the VQ and FkM methods are briefly described. Section 3 shows experimental and comparative results. Finally, in the section 4 some concluding remarks are presented.

2 Automatic Hybrid Classifier Design

Our system works in two phases: training and classification. During the training, the best partition is obtained from the available patterns. During the classification, the new incoming patterns are classified from the results obtained during the training.

Figure 1 displays the training system architecture with two main modules (VQ and FkM) working as follows:

- 1) The VQ module receives the training patterns \mathbf{x}_i and a threshold T and provides a partition (clusters) P_{VQ} . This means that the FkM receives the number of clusters (c), their centers (\mathbf{v}_j) and the training patterns \mathbf{x}_i^j belonging to each cluster.
- 2) The FkM module modifies the submitted partition by updating the cluster centers (\mathbf{v}_j) and computes a membership grade μ_{ij} of the pattern \mathbf{x}_i to the cluster j . In this new partition the patterns remain attached to the clusters.
- 3) The new partition P_{FM} is submitted for verification (see section 2.3).
- 4) If P_{FM} is a valid partition, it is stored in the knowledge base (KB) so that the cluster centers are available during the classification phase after the training.
- 5) If P_{FM} is not validated, the threshold T is modified and a new partition, with the initial set of training patterns available, is intended until validation. As before the new partition is generated by the VQ module.

During the classification phase the system recovers, from the KB, the cluster centers obtained during the training phase. Each new pattern is classified as belonging to the available clusters by applying the FkM approach. Hence, this pattern will belong to each cluster with a different membership degree. The final decision is made based on the maximum membership grade value.

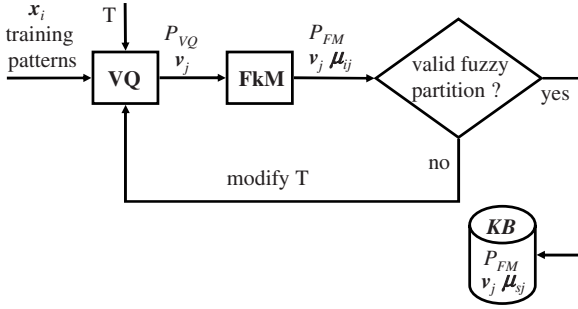


Fig. 1. Hybrid classifier architecture

2.1 Vector Quantization

The VQ approach [9] starts with the observation of a set X of n training patterns, i.e $X = \{x_1, x_2, \dots, x_n\} \in \mathcal{R}^3$. The patterns are three-dimensional because the features are the three spectral RGB values.

Define the threshold T value and select a metric distance (Euclidean).

- 1) For each training pattern x_i , compute its distance to each cluster center v_j , $d_{ij}(x_i, v_j)$. If x_i is the first pattern it is the first center, i.e. $v_1 = x_i$.
- 2) Determine $d_{ik}(x_i, v_k) < d_{ij}(x_i, v_j) \forall j \neq k$.
- 3) If $d_{ik}(x_i, v_k) < T$ then assign x_i to the cluster associated to the center v_k (cluster k) which gives the minimum distance and update this centre v_k by averaging the patterns belonging to this cluster; x_i belongs to the cluster k .
- 4) If $d_{ik}(x_i, v_k) \geq T$ then a new cluster is created and its new cluster centre is exactly the pattern, i.e. $v_k = x_i$.

The partition P_{VQ} is obtained with c clusters, each one with its cluster center v_j where $j = 1, 2, \dots, c$ and a subset $X^c \subset X$ of patterns belonging to each cluster.

2.2 Fuzzy k-Means

Given the number of clusters c and following [10], [11], [12] the FkM algorithm is based on the minimization of the objective function J ,

$$J(U; v) = \sum_{i=1}^n \sum_{j=1}^c \mu_{ij}^m d_{ij}^2 \tag{1}$$

subject to $\mu_{ij} \in [0,1]; \sum_{j=1}^c \mu_{ij} = 1; \sum_{i=1}^n \mu_{ij} < n; 1 \leq j \leq c, 1 \leq i \leq n$. (2)

where $v = \{v_1, v_2, \dots, v_c\}$. These cluster centers are to be updated. The $n \times c$ matrix $U = [\mu_{ij}]$ contains the membership grade of pattern i with cluster j ; $d_{ij}^2 \equiv d^2(x_i, v_j)$ is also the squared Euclidean distance. The number m is called the exponent weight [11]. In order to minimize the objective function (1), the cluster centers and

membership grades are chosen so that high memberships occur for patterns close to the corresponding cluster center. The higher the value of m , the less those patterns whose memberships are low contribute to the objective function. Such patterns tend to be ignored in determining the cluster centers and membership grades [12]. The parameter m is set to 4 in this paper by cross-validation [10] (see section 3.1).

The original FkM computes for each \mathbf{x}_i at the iteration k its membership grade and updates the cluster centers according to equations (3) and (4),

$$\mu_{ij}(k) = \left(\sum_{r=1}^c (d_{ij}(k)/d_{ir}(k))^{2/(m-1)} \right)^{-1}. \quad (3)$$

$$\mathbf{v}_j(k+1) = \sum_{i=1}^n \mu_{ij}^m(k) \mathbf{x}_i / \sum_{i=1}^n \mu_{ij}^m(k). \quad (4)$$

The stopping criterion of the iteration process is achieved when $\|\mu_{ij}(k+1) - \mu_{ij}(k)\| < \varepsilon \quad \forall ij$ or a number N of iterations is reached.

During the classification phase, given a pattern \mathbf{x}_i , it is classified as belonging to the cluster j if $\mu_{ij} > \mu_{ik} \quad \forall k \neq j$, where the membership grades are computed according to (3) and (4).

2.3 Partition Validation

The next step consists in the cluster validation. This is carried out by computing the *partition coefficient* for the number of clusters. This coefficient has been tested favourably in different applications and it is specified as follows [12], [13]:

$$PC(U; c) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^c (\mu_{ij})^2. \quad (5)$$

The maximum value of PC for different values of c determines the best partition, i.e. the best number of clusters for the set of training patterns available. Values of PC near the unity indicate that the partition is acceptable. PC ranges in $[0,1]$ and the unity is the best value. Following the scheme in the figure 1 and based on the PC value, if the partition is rejected then the threshold T must be modified in order to try a new better partition P_{FM} ; otherwise if the current partition is accepted, it is stored in the KB . Each training pattern \mathbf{x}_i is assigned to the class j according to the maximum membership grade; the cluster centers \mathbf{v}_j are those computed by the FkM. These are also stored in the KB to be recovered during the classification phase of new patterns.

We consider that a partition is accepted if PC is greater than the threshold value t set to 0.8 in this paper, which is a commonly accepted value [11].

Different partitions are intended by supplying the patterns in different orders of processing. The order is randomly established.

Moreover, we are using RGB values as features ranging in $[0,255]$. This implies that the clusters (patterns) are mapped in the 3-dimensional space where each axis varies between 0 and 255. Therefore, the maximum Euclidean distance d_M in this space is given by the two opposite points $(0,0,0)$ and $(255,255,255)$ resulting the following value $d_M \approx 442$. Hence, this will be the maximum threshold value T . Its minimum value is bounded by zero. We define the following quantum magnitude

$Q = d_M/40$. The modification process for T is expressed by (6), where we have specified the iteration as k ,

$$T(k) = T(k-1) + Q(1-PC). \quad (6)$$

Initially we set $T(0) = 0$. PC is computed according to the equation (5), initially PC is set to zero, i.e. $T(1) = Q$.

3 Comparative Analysis and Performance Evaluation

We have used a set of 26 digital aerial images acquired during May in 2006 from the Abadin region located at Lugo (Spain). They are multispectral images with 512×512 pixels in size. The images are taken at different days from an area with several natural textures. The initial training patterns are extracted from 10 images of the full set. The remainder 16 images are used for testing and four sets, S0, S1 S2 and S3 of 4 images each one, are processed during the test according to the strategy described below. The images assigned to each set are randomly selected from the 26 images available.

3.1 Design of a Test Strategy

In order to assess the validity and performance of the proposed hybrid VQ and FkM (VF) method we obtain three initial partitions, namely: P_{VQ} , P_{FM} and P_{MI} . P_{VQ} , P_{FM} are the automatic validated partitions obtained by the unsupervised procedure described in this paper, Fig. 1. P_{MI} is a manual partition obtained as described below. Each partition is used as the initial training set of patterns by the following two supervised classical clustering procedures [10]: a) the parametric Bayes classifier (BC) and b) the Self-Organizing feature maps (SO). Other supervised strategies perform similarly.

The test is carried out according to the following steps:

STEP 0 (initial partitions): for each image (from the 10 available) we perform a downsampling by 16, obtaining $10 \times 32 \times 32$ training patterns, i.e. $n = 10240$.

We apply our VF approach starting with $T = Q$ until the partition is validated. At this time we have available the final P_{VQ} and P_{FM} , i.e. the number of clusters c and their corresponding cluster centers are known. Now, we manually select n training patterns from the same set of the 10 images and build c clusters driven by the previous cluster centers. Each manually selected training pattern is compared with each cluster center and it is assigned to the cluster which gives the minimum feature (spectral) distance between the pattern and the corresponding cluster center. So, we obtain the manual initial partition P_{MI} . This partition is used for setting the parameter m (section 2.2) through cross-validation [10]. We select the 90% of patterns for training the FkM, with the other 10% we search for the value of m (testing different values), which gives the minimum classification error.

The P_{VQ} , P_{FM} and P_{MI} partitions are used for classifying the new patterns during the next steps.

STEP 1: given the images in the sets S0 and S1, classify each pixel as belonging to a cluster according to the VF, BC and SO methods. Compute the percentage of successes according to the ground truth defined for each class and for each image. The patterns classified from S1 are added to the previous training patterns and a new

training process is carried out according to each method. The set S_0 is used as a pattern set to verify the performance of the training process as the learning increases.

Perform the same process for STEPs 2 and 3 but using the sets S_2 and S_3 respectively instead of S_1 . Note that S_0 is also processed as before.

The number of training patterns added at each STEP is $4 \times 512 \times 512$ because this is the number of pixels classified during the STEPs 1 to 3 belonging to the sets S_1 , S_2 and S_3 . In the KB we have always available a partition, which is different depending on the STEP because at each STEP there is a different number of patterns.

3.2 Analysis of Results

Figure 2 displays different threshold values against PC , attached are displayed the number of clusters obtained.

The best PC score value is obtained for $T = 70.83$, obtaining four clusters. Larger threshold values result in two clusters and with values greater than 130 a unique cluster is obtained (they are not displayed).

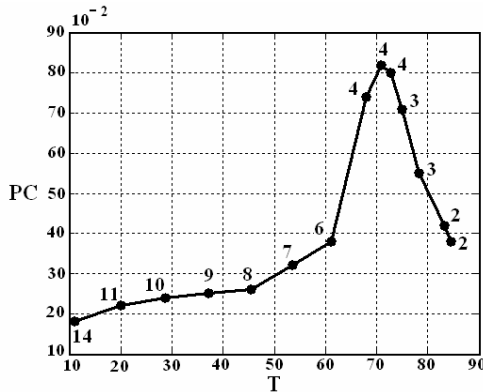


Fig. 2. Threshold values against PC scores (attached are the number of clusters)

Figure 3 displays: (a) a representative original image to be classified belonging to the set S_0 ; (b) the inter-clusters correspondence between the original colors and the labels assigned to each cluster; (c) the labeled image after the classification with our VF approach; (d) the ground truth for the cluster number two.

Each ground truth is built by applying the FkM and then modifying the results manually according to the human expert criterion.

The color for each cluster, Figure 2(b), matches with the natural color assigned to the corresponding cluster center. The labels are artificial colors derived from a color map identifying each cluster.

The correspondence between labels and the different textures is as follows:

- 1) yellow with forest vegetation
- 2) blue with bare soil;
- 3) green with agricultural crop vegetation
- 4) red with buildings and man made structures.

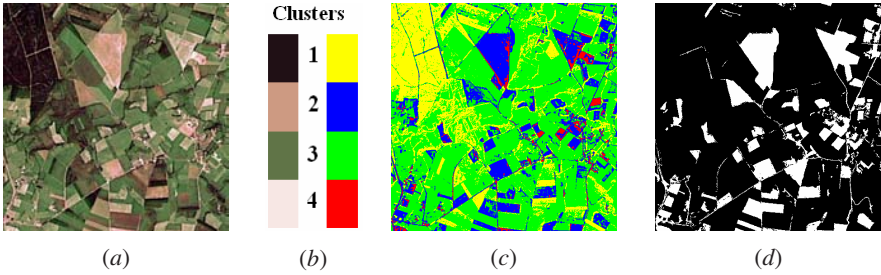


Fig. 3. (a) Original image; (b) colors and labels; (c) labeled textures (b) ground truth for the cluster number two

Table 1 shows the percentage of successes in terms of correct classifications obtained for the different methods and for each initial partition. For each STEP we show the two sets of testing images processed SP0 and SP1/SP2/SP3. These percentages are computed taking into account the correct classifications for the four clusters according to the corresponding ground truth.

Table 1. Percentage of successes obtained for the methods analysed given the initial partitions at the three STEPs

%	Partition	Methods	STEP 1		STEP 2		STEP 3	
			SP0	SP1	SP0	SP2	SP0	SP3
P_{VQ}		VF	71.4	70.0	73.1	74.3	72.8	79.2
		BC	69.9	69.3	70.2	74.2	71.2	78.3
		SO	66.2	67.7	67.1	70.1	69.6	70.2
P_{FM}		VF	75.4	76.2	83.7	81.3	91.3	92.3
		BC	74.9	76.1	82.9	80.1	91.1	91.9
		SO	74.2	73.8	79.2	78.2	86.4	87.1
P_{MI}		VF	75.6	77.2	83.2	81.4	91.7	92.2
		BC	75.2	76.9	82.6	80.5	91.6	91.3
		SO	74.8	73.1	79.0	78.7	85.4	86.6

From results in table 1, one can see that the performance obtained with the initial partition P_{FM} is comparable to that obtained with P_{MI} for all methods and the three STEPs. This means that the proposed unsupervised approach performs in a similar fashion than the supervised ones.

The worst results are obtained with the initial P_{VQ} partition. Therefore, the use of only the unsupervised VQ is not suitable.

We can infer that the performance of the different methods improves as the number of training patterns increases, i.e. high learning rates obtain better performances than low ones.

4 Conclusions

We propose a new unsupervised hybrid and automatic making decision process for classifying natural textures. The proposed method combines the VQ and FkM

approaches achieving results which are comparable to those obtained by the supervised ones. The performance of our approach is analysed with different classical approaches, verifying that it performs favourably in the set of aerial images tested. This approach could be applicable to other textured images even with different attributes.

Acknowledgments. The authors would like to thank to SITGA (Servicio Territorial de Galicia) in collaboration with the Dimap company (<http://www.dimap.es/>) for the original aerial images supplied and used in this paper. This research was funded by the Community of Madrid, project “COSICOLOGI” S-0505/DPI-0391.

References

1. Frate, F. del, Pacifici, F., Schiavon, G. and Solimini, C.: Use of Neural Networks for Automatic Classification from High-Resolution Images. *IEEE Trans. Geoscience and Remote Sensing*, 45(4) (2007) 800-809
2. Giacinto, G., Roli, F. and Bruzzone, L.: Combination of Neural and Statistical Algorithms for Supervised Classification of Remote-Sensing. *Pattern Recognition Letters*, 21(5) (2000) 385-397
3. Chan, J.W.C., Laporte, N. and Defries, R.S.: Texture Classification of Logged Forest in Tropical Africa Using Machine-learning Algorithms. *Journal Remote Sensing*, 24(6) (2003) 1401-1407
4. Hanmandlu, M., Madasu, V.K. and Vasikarla, S.: A Fuzzy Approach to Texture Segmentation. *Proc. IEEE Int. Conf. Information Technology: Coding and Computing (ITCC04)*, The Orleans, Las Vegas, Nevada, (2004) 636-642.
5. Lam, E. P.: Wavelet-based Texture Image Classification Using Vector Quantization. In: Astola, J.T., Egiiazarian, K.O and Dougherty, E.R. (eds.): *Proc. SPIE, Image Processing: Algorithms and Systems V*, vol. 6497 (2007) 687-696
6. Puig, D. and Garcá, M.A.: Automatic Texture Feature Selection for Image Pixel Classification. *Pattern Recognition*, 39(11) (2006) 1996-2009
7. Maillard, P.: Comparing Texture Methods through Classification. *Photogrammetric Engineering and Remote Sensing*, 69(4) (2003) 357-367
8. Drimbarean, P.F. and Whelan, P.F.: Experiments in Colour Texture Analysis. *Pattern Recognition Letters*, 22 (2001) 1161-1167
9. Cherkassky, V., Mulier, F. *Learning from Data: Concepts, Theory and Methods*, Wiley, New York, (1998)
10. Duda, R.O., Hart, P.E. and Stork, D.G.: *Pattern Classification*, Jhon Willey and Sons, New York (2001)
11. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer, Plenum Press, New York (1981)
12. Zimmermann, H.J.: *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers, Norwell (1991)
13. Kim, D.W., Lee, K.H. and Lee, D.: Fuzzy Cluster Validation Index based on Inter-cluster Proximity. *Pattern Recognition Letters*, 24 (2003) 2561-2574.

Visual Texture Characterization of Recycled Paper Quality

José Orlando Maldonado, David Vicente Herrera, and Manuel Graña Romay

Computing Intelligence Group
Dept. CCIA, UPV/EHU
manuel.grana@ehu.es

Abstract. When performing quality inspection of recycled paper one phenomenon of concern is the appearance of macroscopic undulations on the paper sheet surface that may emerge shortly or some time after its production. In this paper we explore the detection and measurement of this defect by means of computer vision and statistical pattern recognition techniques that may allow early detection at the production site. We propose features computed from Gabor Filter Banks (GFB) and Discrete Wavelet Transforms (DWT) for the characterization of paper sheet surface bumpiness in recycled paper images. The lack of a precise definition of the defect and the great variability of the sheet deformation shapes and scales, both within each image and between images, introduce additional difficulties to the problem. We obtain, with both proposed modeling approaches (GFB and DWT), classification accuracies are comparable to the agreement between human observers. The best performance is obtained using DWT features.

Keywords: Quality control, Texture analysis, Industry inspection.

1 Introduction

In the manufacture of recycled paper, the quality control processes may be critical due to the great variability of the raw materials and its effect on the final product quality. Due to uncontrolled raw materials some inhomogeneous distribution of fibers at the microscopic level may produce the presence of undulations of diverse sizes and shapes on the paper sheet surface at the macroscopic level. The goal is to establish a predictive model establishing the relationship between the macroscopic paper sheet measurements and the information about the input raw materials available at production time. To obtain such model, we need the quantitative specification of the paper sheet quality. This article addresses this problem from a computer vision and pattern recognition perspective.

All along the article we will be calling “bumpiness” the appearance of undulations and protuberances on the paper sheet. Our original goal was to define a bumpiness scale from the human labeling of the images obtained from quality control paper sheet samples. This task has proven to be beyond the abilities of the human operators, because they cannot sustain such a fine bumpiness mental model across the experiment due to the variability of the shapes and scales of the protuberances in the paper images. Besides, the task is quite tedious and the human observer loses

concentration easily. Therefore we have restated our initial goal as a classification problem of the images into classes of increasing bumpiness. Figure 1 shows several images of recycled paper sheets. Image acquisition was performed with a conventional flatbed office scanner at an optical resolution of 600 dpi, which generated high-resolution images. Images of recycled paper were pre-processed applying a contrast enhancement that consists on the selection of the upper grey levels that encompass 90% of the image accumulative histogram and renormalization of the image intensity range, to emphasize the visual features of the bumpiness defect, because scanning paper sheets produces white images with very little contrast. The enhanced images show the presence of certain textures that do not keep any regular pattern. Original size of each paper sheet is standard A4: 24 cm x 29.7 cm, the size of the scan images was 850 x 1170 pixels.

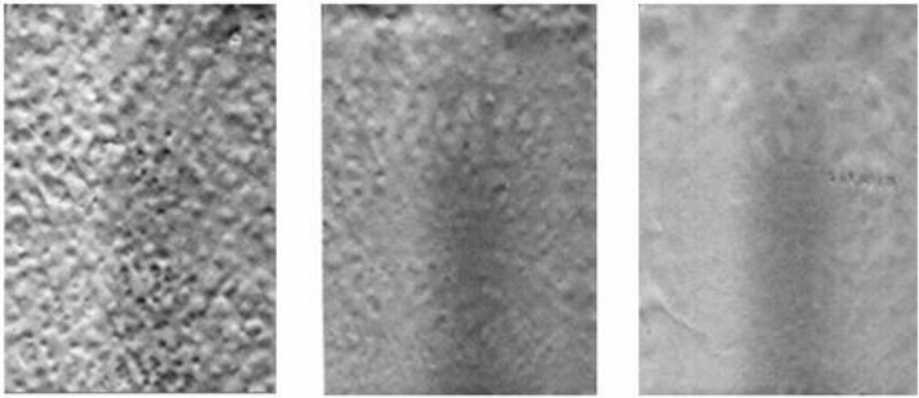


Fig. 1. Images of recycled paper with several levels of bumpiness. Original images are white, with little contrast.

Examples of quality inspection based on texture features are [1], [2], [3], [4], [5], [6], [7] for textile defect detection, [8] for the classification of marble slabs, and [9] for classification of wood surfaces.

When we approach the problem of characterizing the bumpiness of recycled paper sheets through image texture classification we have found that: it is difficult to define a quantitative measure of the paper sheet bumpiness, the partition of the sample images into classes is not trivial and there is a lack of consensus among the human observers, here is a high variability in visual texture features for images of paper sheets that could be classified in the same class, and finally, data is scarce and the defect is not controllable or replicable. As a consequence, we have found that the construction of the desired classification values for each paper sheet image is a laborious problem that deserves proper attention.

The paper is organized as follows: Section 2 is devoted to GFB design for texture characterization. Section 3 is devoted to the definition of DWT texture features. Section 4 gives the results on the classification of the recycled paper images based on GFB and DWT texture features. Section 5 gives some conclusions and directions for further work.

2 Texture Features Computed on Gabor Filters

Gabor elementary functions [10] have the property of being highly selective both in the space domain and in the frequency domain. Although Gabor original works focused in 1D representation, their principles were extended later by [11] to 2D domain.

A two dimensional Gabor function can be written as:

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y} \right) \exp \left[-\frac{1}{2} \left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2} \right) \right] \exp[2\pi i(Ux + Vy)] \quad (1)$$

Where (x', y') , are Euclidean coordinates (x,y) rotated in the space domain:

$$\begin{aligned} x' &= x \cos(\theta) + y \sin(\theta) \\ y' &= -x \sin(\theta) + y \cos(\theta) \end{aligned} \quad (2)$$

Thus, a Gabor function is a Gaussian function modulated by a complex sinusoid. Parameters σ_x, σ_y , characterize the spatial support and bandwidth of the filter. There are six parameters that must be adjusted when implementing the Gabor filter: $[F, \theta, \sigma_x, \sigma_y, B_F, B_\theta]$. The frequency and angular bandwidths (B_F, B_θ) can be set to constant values according to the results found in the psycho visual studies. The frequency (F) and orientation (θ) define the filter centre location in Fourier space. In order to determine the unknown parameters, the following equations can be used:

$$\sigma_x = \frac{\sqrt{\ln 2} (2^{B_F} + 1)}{\sqrt{2\pi f} (2^{B_F} - 1)}, \quad (3)$$

$$\sigma_y = \frac{\sqrt{\ln 2} (2^{B_F} + 1)}{\sqrt{2\pi f} (2^{B_F} - 1)}, \quad (4)$$

In order to extract the texture features from images using filters banks, the even and odd versions of each filter have been used. Let $I(x,y)$ denote an image to be analyzed, $G_e(x,y)$ and $G_o(x,y)$ are convolution masks of the even and odd filter versions previously deduced from $g(x,y)$ constructed with specific values of the parameters $[F, \theta, \sigma_x, \sigma_y, B_F, B_\theta]$, then we calculate the Gabor filter response energy as:

$$E(x, y) = \sqrt{[\mathfrak{S}^{-1}(\mathfrak{S}(G_e(x, y)) \bullet \mathfrak{S}(I(x, y)))]^2 + [\mathfrak{S}^{-1}(\mathfrak{S}(G_o(x, y)) \bullet \mathfrak{S}(I(x, y)))]^2}. \quad (5)$$

Where \mathfrak{S} and \mathfrak{S}^{-1} denote the direct and inverse discrete Fourier transform. Given a filter bank with m orientations and n spatial frequencies, we construct the following texture feature vector:

$$V = \left[(S_{f,\theta}, A_{f,\theta}) : f = 1, \dots, m; \theta = 1, \dots, n \right] \quad (6)$$

Where

$$S_{f,\theta} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N E_{f,\theta}(x, y) \quad (7)$$

$$A_{f,\theta} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [S_{f,\theta} - E_{f,\theta}(x, y)] \quad (8)$$

In this expression, M and N denote the size of the image and $E_{f,\theta}(x, y)$ is the Gabor energy of the response of a filter tuned in frequency f and orientation θ .

3 Texture Features Based on DWT Coefficients

The wavelet transform is a multiresolution signal description introduced in [12], [13] by Mallat, the synthesis of the proposed pyramidal algorithm is shown in figure 2, where the input image is decomposed through of application of successive lowpass and highpass filters by rows and columns respectively, generating four subimages: an approach to lowest resolution as well as the horizontal, vertical and diagonal details.

Texture has been analyzed from the point of view of the DWT in several instances [14]. Texture is defined as the spatial ordering of the image intensity variations. In order to characterize it we focus on the DWT coefficients that contain the image gradient information. Thus, the feature vector is given by the average and standard deviation of each one of the horizontal, diagonal and vertical detail DWT coefficients of the image, at some of the n decomposition levels (i.e. $f_{LH}^i(x, y)$, $f_{HH}^i(x, y)$, $f_{HL}^i(x, y)$). The coarse representation of the image is of no interest. Thus, we construct the vector:

$$V = \left[(S_{i,d}, A_{i,d}) : i = n', \dots, n''; d \in \{LH, HH, HL\} \right] \quad (9)$$

Where n' and n'' are the lower and the upper resolution levels considered, respectively, and

$$A_{i,d} = \frac{1}{M_{i,d} N_{i,d}} \sum_{x=1}^{M_{i,d}} \sum_{y=1}^{N_{i,d}} f_d^i(x, y), \quad (10)$$

$$S_{i,d} = \frac{1}{M_{i,d} N_{i,d}} \sum_{x=1}^{M_{i,d}} \sum_{y=1}^{N_{i,d}} (A_{i,d} - f_d^i(x, y))^2. \quad (11)$$

Where $M_{i,d}, N_{i,d}$ are the sizes of the coefficients of orientation d at decomposition level i .

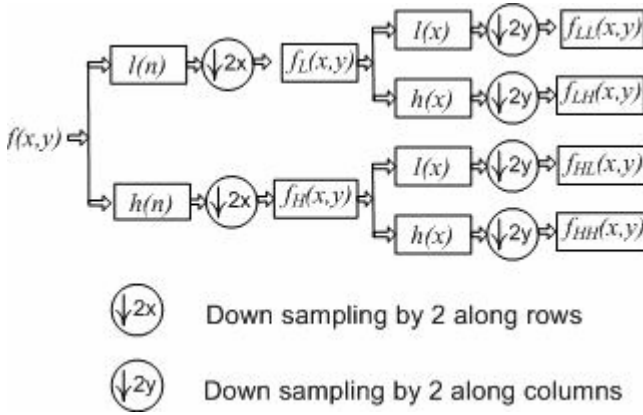


Fig. 2. Figure depicted the fast algorithm for the computation of the discrete wavelet transform proposed by Mallat

4 Automatic Recycled Paper Sheet Image Classification

To obtain the ground truth image labeling, we have used five expert human observers, each performing three times the manual classification of the paper images. After this manual classification process we have 15 labels for each recycled paper image so we can assign it a class label by majority voting. We have made experiments with two types of texture feature vectors, one computed from the results of the GFB designed for the problem and the other from DWT coefficients. Moreover, two methods for the automatic classification of images, using the texture feature vectors as inputs have been examined: the k-nearest neighbor (k-NN) and a feedforward neural network trained with the backpropagation algorithm, usually referred as Multi Layer Perceptron (MLP) in the literature.

For the approach based on GFB, we designed two filter banks that gave us two different texture feature vectors for each image, with the aim to test the sensitivity of the approach to filter bank design. In the first bank, we used a set of Gabor filters with radial bandwidth of 1.4 and angular bandwidth of 35°. The individual filters were tuned to frequencies of 1/8, 1/16, 1/32 and 1/64 cycles/pixel, and with orientation of 0°, 30, 60, 90°, 120° and 150°. With this parameter configuration there is an overlapping of the 56% of the receptive fields of the filters in frequency domain. The receptive field is defined as the ellipsoid that corresponds to the half-magnitude contour of the band filter in frequency space. In the second filter bank design the parameter setting is defined so that the receptive fields of band filters do not overlap.

The k-NN algorithm was executed with diverse values of the number of voting neighbors, we have found that setting k=7 gives the best results for this problem. The MLP was trained with the on-line gradient descent Backpropagation algorithm. The MLP architecture consist of three layers: the input layer that matches the texture feature vector, the hidden layer, with ten neurons, and the output layer, with two neurons, one for each bumpiness class. The activation function chosen was the hyperbolic tangent for all layers. Table 1 present the classification results obtained on

the test set of recycled paper images with the feature vector built from the GFB. The best results are obtained with the MLP on both sets of GB features. The success rate is comparable to the agreement between human observers referred in the previous section. Relative to texture features based on DWT coefficients we have constructed the feature vectors from the average as well as standard deviation of the coefficients of decomposition levels 4 to 6, which it show the best performance. In addition we have tested with the family of Daubechies functions wavelets from order 1 (known as the Haar wavelet transform) to order 8.

Table 1. Results for GFB features without receptive field overlap and GFB features with receptive field overlap respectively

	Without receptive field overlap		With receptive field overlap	
	7-NN	MLP	7-NN	MLP
C1	80%	78,5%	87,7%	80%
C2	67,7%	76,9%	64,6%	75,4%
Average	73,8%	77,7%	76,2%	77,7%

Table 2. Success in the classification by 7-NN with different levels of decomposition and different mother wavelets

	db1	Db2	Db3	Db4	Db5	Db6	Db7	Db8
123	66,15	60,77	56,15	69,23	62,31	64,62	50,77	50,00
234	69,23	72,31	67,69	77,69	70,77	74,62	72,31	61,54
345	77,69	76,92	73,08	73,85	76,92	71,54	73,85	72,31
456	83,85	70,00	62,31	70,00	63,85	70,77	73,85	73,85
567	80,00	56,15	50,00	56,15	57,69	65,38	54,62	56,15
789	70,00	55,38	54,62	51,54	52,31	50,00	50,77	60,00

As with GFB texture features, we have used the k-NN algorithm and the MLP neural network. In table 2 it can be seen the success achieved by means of the k-NN method for each wavelet mother function and each feature vector. The best results were obtained with the vector constructed from the coefficients of levels 4 to 6, with the Haar mother wavelet (db1) and neighborhood parameter k=7. As could be expected, the lower decomposition levels do not reflect the spatial characteristics of the problem, being more representative of noise at short spatial ranges. The higher decomposition levels reflect spatial features of greater span than the ones produced by the bumpiness in the paper sheet. In table 3, the results obtained by means of MLP are shown. In this case the best results also correspond to features computed from the decompositions at levels 4 to 6, but with Daubechies mother wavelet of order 4 (db4).

This result improves the ones obtained with the k-NN algorithm, as well as the ones obtained from features computed over GFB applied to the images.

Table 3. Success in the classification by MLP with different levels of decomposition and different wavelets mother

Deep	db1	db2	db3	db4	db5	db6	db7	db8
1-3	78,46	69,23	53,85	42,31	66,15	63,85	60,77	62,31
2-4	76,15	76,15	73,85	71,54	70,77	70,77	64,62	65,38
3-5	79,23	78,46	78,46	71,54	82,31	76,15	73,85	74,62
4-6	80,00	46,92	80,77	85,38	82,31	78,46	76,15	83,08
5-7	77,69	52,31	51,54	44,62	43,85	76,15	56,92	76,15
6-8	50,00	50,00	50,00	47,69	50,00	46,92	44,62	50,00

In all cases we observe that the error of the classification is smaller for Hard bumpiness images, which agrees with the results found in the analysis of manual image labeling, where it is easy to notice that, in general, an image with Hard bumpiness is easier to distinguish and to classify by the user. The boundary cases between Hard and Low bumpiness are the bigger source of disagreement between the human observers, and the main source of error for the automatic classification systems.

5 Conclusions and Further Work

Detection and measurement of macroscopic undulations and protuberances on recycled paper sheet images (bumpiness) can be made by means of image processing and statistical pattern recognition techniques. The problem is stated as a classification problem: each class corresponds to a degree of bumpiness. Contrary to other texture classification problems for quality inspection, the definition of the texture classes and their relationship to the measurement of the bumpiness is far from trivial. Due to the difficulties encountered by the human observers in the image labeling process, the number of classes has been reduced to three (“no bumpiness”, “Low bumpiness” and “Hard bumpiness”). The “no bumpiness” class is trivial, because the image lacks any feature whatsoever. In this paper we focus on the search for the appropriate definition of the texture features. The classification algorithms are standard: the k-NN algorithm and the MLP neural network. Specifically, the use of GFB to compute texture features of the images allows for the construction of automated classifiers that reach a level of classification success comparable with that obtained by human labelers. The use of texture features based on DWT coefficients improves the results obtained with GFB, and the human labelers agreement, therefore DWT features may be considered optimal so far for the problem at hand.

Our future research will focus in testing the use of the optimal texture features for the computation of an index that may serve as a continuous measure of the bumpiness level of a recycled paper sheet image. This continuous index may later be correlated with the production parameters with the aim of incorporating it in the quality control process. Again, the validation of this continuous index will involve new human labeling tests. We are working on the design of the human validation interface to lower the burden of maintaining a precise discrimination between past viewed images

and the new ones being presented. This can be accomplished asking the human observer to verify some properties on collections of images, such as being of equal bumpiness or ascending/descending order of bumpiness.

References

1. Sari-Sarraf, H. and Goddard, J.S., Jr.: Vision system for on-loom fabric inspection. *IEEE Trans. Ind. Appl.* 35, 1252-1259 (1999)
2. Chi-Ho, C. and Pang, G.K.H.: Fabric defect detection by Fourier analysis. *IEEE Trans. Ind. Appl.* 36, 1267-1276 (2000)
3. Anagnostopoulos, C., Anagnostopoulos, I., Vergados, D., Kouzas, G., Kayafas, E., Loumos, V., and Stassinopoulos, G.: High performance computing algorithms for textile quality control. *Math. Comput. Simulat.* 60, 389-400 (2002)
4. Abouelela, A., Abbas, H.M., Eldeeb, H., Wahdan, A.A., and Nassar, S.M.: Automated vision system for localizing structural defects in textile fabrics. *Pattern Recogn. Lett.* 26, 1435-1443 (2005)
5. Ngan, H.Y.T., Pang, G.K.H., Yung, S.P., and Ng, M.K.: Wavelet based methods on patterned fabric defect detection. *Pattern Recogn.* 38, 559-576 (2005)
6. Kumar, A. and Pang, G.K.H.: Defect detection in textured materials using Gabor filters. *IEEE Trans. Ind. Appl.* 38, 425-440 (2002)
7. Scharcanski, J.: Stochastic Texture Analysis for Measuring Sheet Formation Variability. *IEEE Trans. Instrum. Meas.* 55, 1778-1785 (2006)
8. Martinez-Alajarin, J., Luis-Delgado, J.D., and Tomas-Balibrea, L.M.: Automatic system for quality-based classification of marble textures. *IEEE Trans. Syst. Man. Cy. C.* 35, 488-497 (2005)
9. Funck, J.W., Zhong, Y., Butler, D.A., Brunner, C.C., and Forrer, J.B.: Image segmentation algorithms applied to wood defect detection. *Comput. Electron. Agr.* 41, 157-179 (2003)
10. Gabor, D.: Theory of communication. *J. IEE.* 93, 429-457 (1946)
11. Daugman, J.G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by twodimensional visual cortical filters. *J. Opt. Soc. Am.* 2, 1160-1169 (1985)
12. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal.* 11, 674-693 (1989)
13. Mallat, S.G.: Multifrequency channel decompositions of images and wavelet models. *Acoustics, IEEE Trans. Signal Proces.* 37, 2091-2110 (1989)
14. Fahmy, G., Black, J.J., and Panchanathan, S.: Texture characterization for joint compression and classification based on human perception in the wavelet domain. *IEEE Trans. Image Process.* 15, 1389 – 1396 (2006)

Combining Improved FYDPS Neural Networks and Case-Based Planning – A Case Study

Yanira de Paz, Quintín Martín, Javier Bajo, and Dante I. Tapia

Departamento Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
{yanira, qmm, jbajope, dantetapia}@usal.es

Abstract. This paper presents a hybrid deliberative architecture based on the concept of CBP-BDI agent. A CBP-BDI agent is a BDI agent that incorporates a CBP reasoning engine. The work here presented focuses in the development of the CBP internal structure. The planning mechanism has been implemented by means of a novel FYDPS neural network. The system has been tested and this paper presents the results obtained.

1 Introduction

In this article we present a novel hybrid planning system based on the combination of neuronal networks with CBP (Case-based planning) systems [8]. CBP allows us to retrieve past experiences when a new plan is created which lends the system a large capacity for learning and adaptation [8]. The neuronal networks proposed within this research framework are self-organised, based on Kohonen [11] networks, but which present certain improvements (FYDPS neural Neural Network) [13]. These improvements allow the network to reach a solution much more rapidly. Furthermore, once a solution has been reached, it makes it possible to make new modifications taking restrictions into account (specifically time restrictions).

Case-based planning is based on the way through which a new plan is generated through experiences acquired in the past (after the creation and execution of plans to resolve similar problems to the current one). Case-based planning is carried out through a CBP cycle [2], [3], [8]. The CBP cycle is formed by four sequential stages: retrieve, reuse, revise and retain. In the retrieve stage past experiences are recuperated with a description of the problem similar to that of the current problem. In the revise stage the results attained after executing a new plan are evaluated. Lastly, in the retain stage, lessons are learnt from the new experience. Each one of the stages of the CBP cycle may be implemented in various ways, using different algorithms. In this article we present a novel model that allows the integration of the planning based on cases from FYDPS networks. This model offers greater speed for obtaining the solutions that Kohonen networks, and incorporated restrictions in the network.

The hybrid planning system developed has been applied to an existent Multiagent System, developed for guiding and advising users in Shopping Centres (also known as shopping malls) [2], [3]. A shopping centre is a dynamic environment, in which shops change, promotions appear and disappear continuously, etc. The proposed system

helps users to identify a shopping or leisure plan as well as to identify other users within a given shopping mall. Multiagent systems (MAS) are specifically recommended for solving dynamic distributed systems. A CBP-BDI agent is a deliberative agent that works at a high level with the concepts of Believe, Desire, Intention (BDI) [2], [9]. A CBP-BDI agent uses case-based planning as a reasoning mechanism, which allows it to learn from initial knowledge, to interact autonomously with the environment as well as with users and other agents within the system, and to have a large capacity for adaptation to the needs of its surroundings. The multiagent system used a system of planning based on geodesic calculus [2], [3]. The results obtained with the planning system proposed in this study are compared to those obtained with the previous planning system and with a classic planning system.

Section two presents the shopping mall wireless multiagent system, then section three introduces the planning strategy and section four presents the novel FYDPS neural network model finally, the system is evaluated and the conclusions discussed.

2 Shopping Mall Multiagent System

This paper presents a distributed architecture whose main characteristics are the use of a CBP-BDI guiding agent, wireless agents and RFID technology [2], [3]. The CBP-BDI agent incorporates a reasoning Case Based Planning (CBP) engine which allows the agent to learn from initial knowledge, to interact autonomously with the environment and users, and allows it to adapt itself to environmental changes by discovering knowledge “know how”. The aim of this work is to obtain a model for recommending plans in dynamic environments. The proposal presented has been used to develop a guiding system for the users of a shopping mall that helps them to identify bargains, offers, leisure activities, etc. An open wireless system has been developed, which is capable of incorporating agents that can provide useful guidance and advice services to the users not only in a shopping centre, but also in any other similar environment such as the labour market, educational system, medical care, etc. Users (clients in the mall) are able to gain access to information on shops and sales and on leisure time activities (entertainment, events, attractions, etc) by using their mobile phone or PDA. Mechanisms for route planning when a user wants to spend time in the mall are also available. Moreover, it provides a tool for advertising personalized offers (a shop owner will be able to publicise his offers to the shopping

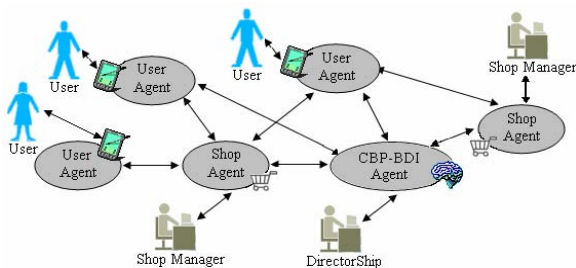


Fig. 1. Shopping Mall multiagent system: CBP-BDI agent, Shop agents and User agents

mall users), and a communication system between management, the commercial sector or shoppers. Figure 1 shows the shopping mall multiagent system structure that has been developed and that is explained in detail in previous works [2], [3].

3 New Neural Network-Based Hybrid Plannig System

The CBP is a variation of the CBR [1] which is based on the generation of plans from cases. The deliberative agents, proposed in the framework of this investigation, use this concept to gain autonomy and improve their guiding capabilities. The relationship between CBP systems and BDI agents can be established by implementing cases as beliefs, intentions and desires which lead to the resolution of the problem. In a CBP-BDI agent, each state is considered as a belief; the objective to be reached may also be a belief. The intentions are plans of actions that the agent has to carry out in order to achieve its objectives [9], so an intention is an ordered set of actions; each change from state to state is made after carrying out an action (the agent remembers the action carried out in the past, when it was in a specified state, and the subsequent result). A desire is any of the final states reached in the past (if the agent has to deal with a situation, which is similar to one in the past, it will try to achieve a similar result to the one previously obtained). Below, the CBP hybrid guiding mechanism, used by the CBP-BDI guiding agent, is presented: Let $E = \{e_0, \dots, e_n\}$ the set of the possible interesting places to visit and shop at.

$$a_j : E \xrightarrow{e_i} E \xrightarrow{a_j(e_i)=e_j} E \tag{1}$$

An Agent plan is the name given to a sequence of actions (1) that, from a current state e_0 , defines the path of states through which the agent passes in order to offer to the user the better path according to each user's characteristics. Below, in (2), the dynamic relationship between the behaviour of the agent and the changes in the environment is modelled. The behaviour of agent A can be represented by its action function $a_A(t) \forall t$, defined as a correspondence between one moment in time t and the action selected by the agent,

$$Agent A = \{a_A(t)\}_{t \in T \subseteq N} \tag{2}$$

From the definition of the action function $a_A(t)$ a new relationship that collects the idea of an agent's action plan (3) can be defined,

$$p_A : Tx_A \xrightarrow{(t, a_A(t))} A \xrightarrow{p_A(t)} A \tag{3}$$

in the following way,

$$p_A(t_n) = \sum_{i=1}^n a_{iA}(t_i - t_{i-1}) \tag{4}$$

Given the dynamic character that we want to print onto our agent, the continuous extension of the previous expression (4) is proposed as a definition of the agent plan, in other words (5) -

$$p_A(t_n) = \int_0^{t_n} a_A(t) dt \quad (5)$$

The variation of the agent plan $p_A(t)$ will be provoked essentially by: the changes that occur in the environment and that force the initial plan to be modified, and the knowledge from the success and failure of the plans that were used in the past, and which are favoured or punished via learning. The hybrid planning is carried out through a neural network based on the Kohonen network [10]. The neurons are organized in a two-layer unidirectional architecture. The learning method is presented as follows: 1) To present the input vector $X^p = (x_1^p, \dots, x_i^p, \dots, x_N^p)^T$ in the input layer. 2) The weightings initially take random weightings in (0,1). 3) To calculate the intensity of the neurons of the output layer. The Euclidean distance:

$$y_k = \sqrt{\sum_{i=1}^N (x_i - w_{ki})^2} \quad (6)$$

4) To determine the winning neuron that will be that of smaller Euclidean distance, and 5) To upgrade the weights of the neurons that connect the input layer with the output neuron:

$$w_{ki}(t+1) = w_{ki}(t) + \eta(t)g(k, h, t)(x_i(t) - w_{ki}(t)) \quad (7)$$

$$g(k, h, t) = e^{\frac{-|k-h|^2}{2R(t)^2}} \quad (8)$$

Where w_{ki} is the weight of the connection between the input neuron i and the output neuron k ; t is the iterations; η is the learning rate; h is the position of the winning neuron; k is the neuron of the output layer; and i is the neuron of the input layer. In $k-h$ a Euclidean distance is calculated between the neurons.

4 Defining Self-organising Neural Network in a Novel Way: FYDPS Neural Network

The basic Kohonen network [11] cannot be used to resolve our problem since it attempts to minimise distances without taking into account any other type of restriction, such as time limits. In the present study a planner is described that is based on Kohonen networks but with a number of improvements (FYDPS Neural Network) [13] that allow us to reach a solution far more rapidly. Furthermore, once a solution has been reached, it is re-modified in order to take restrictions into account.

4.1 Objective 1: Reach a Solution More Rapidly

As such, for this modification of the basic algorithm (FYDPS), we are aiming to make the solution search more agile and in order to achieve this, the basic vicinity function used in the Kohonen network is modified and the number of neurones in the output

layer corresponds to the places that the subject wishes to visit. The topology of the neural network being considered is described below. The input layer is formed by two neurons, each one of those receives one of the coordinates of the shop presented as input. A vector of neurons is used of size the same as the number of places to visit by clients of the problem in the output layer, as in [10], [12]. The number of neurons in the output layer isn't modified. Let $x_i \equiv (x_{i1}, x_{i2}) \quad i = 1, \dots, N$ the coordinates from the shop i and $n_i \equiv (n_{i1}, n_{i2}) \quad i = 1, \dots, N$ the coordinates the neurons i in \mathfrak{R}^2 . N shops will be visited by the fixed client. Then there will be: Two neurons in the input layer and N neurons in the output layer. It will be considered a vicinity function decreasing with the number of iterations.

$$g(k, h, t) = \text{Exp} \left[\left(-\frac{|k-h|}{N/2} \right)^{\frac{\text{Máx}_{i,j \in \{1, \dots, N\}, i \neq j} \{f_{ij}\}}{\text{Máx}_{i,j \in \{1, \dots, N\}, i \neq j} \{f_{ij}\}}} - \sqrt{(n_{k1} - n_{h1})^2 + (n_{k2} - n_{h2})^2} \right) - \lambda \frac{|k-h|t}{\beta N} \right] \quad (9)$$

λ and β are determined empirically, their respective values are: 5 and 50. t is the current iteration. $\text{Exp}[x] = e^x$. N is the number of the shops that are visited by a fixed client and f_{ij} is the distance given by the Floyd Algorithm [14].

The radius of final vicinity should be similar to 0 so that the winner is only upgraded. Iteratively the group of shops will be presented, so that the weights of the neurons approach the coordinates of the shops. When concluding the process, there will be a neuron associated to each shop. To determine the route to follow, we will leave the shop associated to the neuron i to the associated to $i+1$, for $i=1, 2, \dots, N$, passing the whole vector of neurons. To close the road, the last tract will be given by the route that joins the shop in the commercial center associated to the neuron N with the associated to the neuron 1 . The distance of the road will be given by the sum of the distances between the successive couples of shops of the road. The learning rate is a decreasing function:

$$\eta(t) = \text{Exp} \left[-\sqrt[4]{\frac{t}{\beta N}} \right] \quad (10)$$

The function of activation of the neurons is the identity. When the system stops, the route to continue will be given by the weights of the neurons that will be very closer to the coordinates of the shops. To know which the following shop in the journey, is we pass to the following element of the vector of neurons. The neurons are stored in a vector that contains the weights of each one in the current instant. So that the vector defines a ring, the neuron n_1 is the following to the n_N considered. With a big radius of vicinity, in the first iterations of the algorithm the victory of a neuron affects great part of the map, so that a global self-organization takes place. If the radius decreases, the effect of a victory affects every time a part smaller in the map, so that the the criterion to stop the learning of the network is that the distance among shops cannot be optimized more. The initial number of total iterations is of $T_1 = \beta N$

(first phase). When $t = \beta N$, all the couples of possible neurons are exchanged (exchanging their weights) in the obtained ring of neurons, if the distance is optimized then the learning isn't finished. In general, in the phase Z , the total number of interactions to carry is:

$$T_Z = T_{Z-1} - \frac{T_{Z-1}}{Z} \quad (11)$$

The aim of these phases is to eliminate the crossings. Concluded the iterations of each phase is proven if the distance is already optimized, in such a way that in the phase that stops the learning, the distance is minimum. The primary objective - to achieve artificial neural networks that are faster than basic Kohonen networks, applied to the problems that the basic networks resolve - has been achieved. In the apparatus below the necessary modifications are introduced into the algorithm so that the network can take restrictions into account, and therefore be able to resolve other problems that cannot be resolved by basic Kohonen networks.

4.2 Objective 2: Taking Time Restrictions into Account

Instead of using the Euclidean distance, a different distance is used that we call "temporal distance". Without losing any generality it can be supposed that a distance unit is equivalent to a time unit.

$$dt_{ij} \equiv dt(x_i, x_j) \triangleq \text{Máx}\{f_{ij} + t_i, b_j\} \quad (12)$$

Where t_i is the time hended to get to shop "i" from the previous shop plus the time taken on tasks to be carried out within shop "i" (in other words, the service time in "i") and b_j is the time limit for carrying out the tasks in shop "j". In this way, the vicinity function of the network modified from the FYDPS network is:

$$g(k, h, t) = \text{Exp} \left[\left(-\frac{|k-h|}{N/2} \right) \frac{\text{Máx}_{\substack{i,j \in \{1, \dots, N\} \\ i \neq j}} \{d_{ij}^*\} - \sqrt{(n_{k1} - n_{h1})^2 + (n_{k2} - n_{h2})^2}}{\text{Máx}_{\substack{i,j \\ i \neq j}} \{d_{ij}^*\}} - \lambda \frac{|k-h|t}{\beta N} \right] \quad (13)$$

$$d_{ij}^* \equiv d^*(x_i, x_j) \triangleq \frac{f_{ij} + t_i}{dt_{ij}} \quad (14)$$

When the network obtains as a result the optimum plan p^* . If the plan p^* is not interrupted, the agent will reach a desired state $e_j \equiv e^*$. In the learning phase, a weighting $w_j(p)$ is stored. With the updating of weighting $w_j(p^*)$, the planning cycle of the CBP motor is completed. Let's suppose that the agent has initiated a plan p^* but at a moment $t > t_0$, the plan is interrupted due to a change in the environment. The solutions given by the neural network meet the conditions of the Bellman Principle of Optimality [2], in other words, each one of the plan's parts is partially optimum between the selected points. This guarantees that if g_0 is optimum route for interrupted

e_0 in t_1 , because e_0 changes to e_1 , and g_1 optimum route to e_1 that is begun in the state where g_0 has been interrupted, it follows that: $g = g_0 + g_1$ is optimum route to $e = e_0 (t_1 - t_0) + e_1 (t_2 - t_1)$.

5 Results and Conclusions

The hybrid planning model proposed has been integrated within a previously Developed multiagent system [2], [3] and has been tested at the Tormes shopping mall in Salamanca. The users at the commercial centre access the planning service from their mobile devices (telephones or PDA's). An example of its use is illustrated in Table 1. In the example shown in Table 1, the user defines his desire to buy music and clothes and to go to a gift shop. The user also indicates his desire to see an action film at the cinema, eat a pizza, and have a drink. The system proposes an additional activity to go to a mobile phone shop. The available time introduced was seven hours and no limit was placed on the money available. Table 1 shows the sequence of shops visited, the distance covered, and the time used at each stage of the plan. The user leaves initial point 0 in the Shopping Mall. First, he visits the gift shop. Then he visits a mobile phone shop. The third stage consists in a visit to a music shop and during stages 4 and 5 he goes shopping in clothes shops. In stage 6 the user goes to the cinema. At this stage the user has a time restriction to be at the cinema 10 minutes before the film begins (18.50 - 19.00) and to use the service during the subsequent hour and forty minutes. When the user leaves the cinema, he dines at restaurant 7. Lastly, the user visits three video arcades.

Table 1. Planning example

Shop Coordinates	Distance	Arrival Time	End Time
(254, 444)	0	17:00	
(376, 456)	138	17:02	17:08
(436, 456)	84	17:10	17:20
(472, 532)	60	17:21	17:35
(530, 456)	82	17:37	18:00
(602, 390)	148	18:03	18:40
(460, 94)	388	18:47	20:30
(332, 270)	320	20:36	21:00
(604, 270)	172	22:13	22:37
(504, 300)	46	22:38	23:00
(504, 332)	46	23:01	23:27

Table 2. Comparison of planners

Planner	Execution Time (Secs)
Classic	134
Geodesic based	39
Proposed planner	27

In terms of the efficacy obtained with the new planning model, the results of the new model have been compared with those of a classic planner and with the prior system. A set of synthetic tests has been developed, proposing 50 cases for generating a plan in each planner. The average times taken by each planner to generate a plan is illustrated in Table 2. Table 2 shows how the planner proposed in this study significantly improves the time taken over classical planner, and also slightly improves on the time taken by a geodesic based planner.

References

1. Aamodt A. and Plaza E.: Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7., pp 39-59 (1994)
2. Bajo J., de Paz Y., de Paz J.F., Martín Q. and Corchado J.M.: SMAS: A Shopping Mall Multiagent Systems. Proceedings of IDEAL'06, LNAI, vol 4224 pp. 1166-1173, Springer Verlag. (2006)
3. Bajo J., Corchado J.M. and Castillo L.F.: Running Agents in Mobile Devices. Proceedings of IBERAMIA'06, LNAI, vol 4140 pp. 58-67, Springer Verlag. (2006)
4. R.E. Bellman, Dynamic Programming. (Princeton University Press, Princeton, New Jersey, 1957).
5. Bratman, M.E.: Intentions, Plans and Practical Reason (Harvard University Press, Cambridge, M.A., 1987).
6. Corchado J.M., and Laza R.: Constructing Deliberative Agents with Case-based Reasoning Technology, International Journal of Intelligent Systems, 18 (2003) 1227-1241.
7. Corchado J.M., Pavó J., Corchado E. and Castillo L.F.: Development of CBR-BDI Agents: A Tourist Guide Application, in: Proc. ECCBR'04, Lecture Notes in Artificial Intelligence, Vol. 3155 (Springer, Berlin, 2004) 547-559.
8. Cox M.T., Muñoz-Avila, H. and Bergmann R. Case-based planning. The Knowledge Engineering Review, Vol. 00:0, 1-4. c 2005, Cambridge University Press
9. Glez-Bedia M., Corchado J.M., Corchado E. and Fyfe C.: Analytical Model for Constructing Deliberative Agents, Engineering Intelligent Systems, Vol 3 (2002) 173-185.
10. Jin H.D., Leung K.S., Wong M.L., Xu Z.B.: An Efficient Self-Organizing Map Designed by Genetic Algorithms for the Traveling Salesman Problem. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, vol. 33, no 6. (2003). pp 877-888.
11. Kohonen T.: Self-organization and associative memory, Springer Verlag (1984).
12. Leung K.S., Jin H.D., Xu Z.B.: An expanding Self-organizing Neural Network for the Traveling Salesman Problem. Neurocomputing, vol. 62. (2004). pp 267-292.
13. Martín Q., De Paz J.F., De Paz Y., Pérez E., Solving TSP with a modified kohonen network, European Journal of Operational Research, (2007).
14. Martín Q., Santos M.T., De Paz Y., Operations research: Resolute problems and exercises, Pearson, (2005) 189-190.

CBR Contributions to Argumentation in MAS

Stella Heras, Vicente Julián, and Vicente Botti

Department of Information Systems and Computation
Technical University of Valencia
Camino de Vera s/n. 46022 Valencia, Spain
Tel.: (+34) 96 387 73 50; Fax.: (+34) 96 387 73 59
{sheras, vinglada, vbotti}@dsic.upv.es

Abstract. On discussing the necessary features for a group of agents to argue about their positions and intentions over a specific issue some questions arise: Why do agents interact? Is saving the knowledge generated in the interaction useful? How do agents manage arguments? How do agents dialogue? CBR is an adequate way to tackle such argumentation issues in MAS. This paper clarifies the advances achieved by applying CBR to argumentation in MAS, identifies open issues and proposes new ideas to face future challenges.

1 Introduction

Recently, the application of Argumentation Theory to Artificial Intelligence has gained an increasing interest [6]. In Multi-Agent Systems (MAS) technology a key reason for this advance is the importance of using arguments to make agents truly intelligent. This implies that agents must be able to act as individual entities and autonomously interact with others. In such interactions, the goals of the agents may conflict with those of their partners. Moreover, in dynamic and open environments the knowledge that the agents have about the environment and other agents and even their own goals are susceptible to change. To deal with this uncertain information and changing intentions, the agents must reach agreements that harmonize not only conflicts with partners, but also their mental states and preferences. Argumentation is a natural way to reach agreements. Its techniques can be used to specify autonomous agent reasoning and to facilitate multi-agent interaction [12].

Case-Based Reasoning (CBR) [1] is other important field where argumentation techniques have also been successfully applied. The main assumption in CBR is that similar problems have similar solutions. Therefore, when a CBR system solves a new problem, it retrieves precedents from its case-base and adapts their solutions to fit the current situation. This reasoning methodology has a high resemblance with the way people argues over its experiences. Thus, CBR has been applied in precedent-based legal systems, where similar cases should have similar verdicts. A preliminary example of case-based legal reasoning system is HYPO [5], which generates legal arguments to decide the winner of a dispute by citing precedents. Its development resulted in the systems CABARET [13], CATO [3] and BankXX [14]. Other successful application of CBR to argumentative frameworks is the PERSUADER system [17], which supports conflict resolution through persuasive negotiation.

The fine results of argumentation-based CBR systems demonstrate that CBR is a suitable methodology for argumentation processes. Moreover, storing past experiences as cases is a powerful way to manage incomplete, uncertain and inconsistent information. Therefore, CBR is expected to be an adequate way to tackle the underlying matters of argumentation in MAS. CBR can also increase the intelligence of agents by providing them with learning capabilities. Recently, some research has been done to adapt argumentative CBR to argumentation-based interaction in MAS. This paper analyses the main trends, focusing on how CBR has been used to improve the autonomous reasoning of agents and to direct the interaction between them. Moreover, the paper attempts to clarify the advances achieved in the area, to identify open issues and to propose new ideas to face future challenges.

2 CBR-Aided Argumentation in MAS

Walton and Krabbe [20] established in the context of the Argumentation Theory a dialogue typology that depends on the initial situation, the main goal of the dialogue and the participants aims. The dialogue types include: *persuasion*, *negotiation*, *information inquiry*, *deliberation* and *eristics*. This typology can be used to classify different types of communication in MAS. The main research done to apply CBR to argumentation in MAS falls into *argumentation-based negotiation* and *collaborative deliberation*. Next sections identify the three main contribution areas. First, the case-base has been used as knowledge resource for improving the autonomous reasoning of agents or the system overall reasoning. Second, reasoning about cases has been used for managing arguments. Third, the interaction protocol has been directed by the CBR decisions.

2.1 Case-Base as Knowledge Resource

Since the knowledge about past experiences is the basis over which the CBR systems reason, the specific contents and structure of the case-base is an important design decision that differentiates and defines their operation. In MAS the case-base has been used as a knowledge resource that represents the ‘memory’ of the agents or the system, like a track record storing the experience that the MAS components gain during their life. This knowledge increases the agents autonomy and may lead to a more successful interaction between them. The contents of the cases depend on the reasoning objective of the MAS.

In argumentation-based negotiation dialogues, CBR has been applied to negotiate more efficiently by using the experience obtained from past negotiations. This is the approach of Soh and Tsatsoulis [15]. In their framework a group of situated agents which control certain sensors try to reach mobile physical objectives following the best path. The sensors have limited coverage and power capabilities and hence, the agents must negotiate to track their objectives. When a new negotiation starts, the agents determine how to argue with their partners (the best *negotiation strategy*) by using the knowledge about past negotiations that they store in their case-bases. Therefore, the cases include *case descriptors*, which are parameters used to

determine if a current negotiation is similar to that represented in a case and a set of *negotiation parameters*, which define the main characteristics of the past negotiation.

In the case of argumentation-based collaborative deliberation frameworks, the role of CBR depends on the agent capabilities and hence, on the reason behind the collaboration. On one hand, if the agents are completely autonomous and are able to obtain a solution for the issue under discussion without any help, CBR has been used to solve the problem independently. This is the approach adopted in the *Argumentation-based Multi-Agent Learning (AMAL)* framework of Ontañ and Plaza [10]. Here, the agents act as independent classifiers (between a pre-defined set of solution classes) that have private case-bases and learn from their experiences. The argumentation-based interaction in this case is intended to improve the solution quality by adding the knowledge of a set of expert agents. If the knowledge of two agents gives rise to contradictory solutions, CBR is used to choose the preferred one. Therefore, each case $\langle P, S \rangle$ is a tuple that contains a *case description* (set of attributes that describe the problem P) and the *solution class* S to which the case belongs.

On the other hand, CBR has been applied to moderate the dialogue of a group of agents that have not enough autonomy to independently solve a problem. These agents must collaborate to achieve a coherent joint decision via argumentation. In this kind of frameworks the agents can either interact directly or centralise the interaction into a *Mediator Agent (MA)*. In this fashion, the architecture of the decision support system *CARREL+* [19] was extended with a new argument-based selection model called *ProClaim* [18]. In *CARREL+* an agent that represents an organ donor (DA) and a set of agents that represent potential recipients (RA's) argue about the viability of the organ transplantation. *ProClaim* features a MA that controls the deliberation. The MA uses CBR to evaluate the arguments about organ viability that the agents generate in basis of the knowledge acquired from past transplants. Therefore, each case of the case-base has associated a *case description* with the medical information describing the transplant and an *argument graph* that shows the arguments that were submitted by the agents engaged in that past transplant process. Moreover, an argument graph has an *evidential support*, which is a tuple $\langle F, K \rangle$ where F represents the certainty in the decision correctness and K the number of cases that share the argument graph.

Following a similar approach, Karacapilidis et al. developed a collaborative deliberation framework to integrate CBR and argumentative reasoning into the *Argument Builder Tool* of the decision maker system *HERMES* [8]. The system helps a group of human agents to argue in supporting or attacking alternative positions in a decision process. It maps the process to a *discussion graph* with a tree structure. The framework differs from *ProClaim* since in this case the system is who controls the interaction between agents. CBR is a reasoning component included in the system design that provides warrants about arguments by referring to previous disputes. There is a common case-base where the cases are flexible entities with store a set of argumentation elements that can be interpreted depending on the state of the discourse (previous instances of the discussion graph) and the agent interests.

Challenges. How to reason about arguments, their interpretation and the relations between them are key issues in argumentation domains. Therefore, the contents of the case-base must consider such issues and ease the reasoning over them. If the arguments are simply stored as data in the cases, the semantic knowledge about them that has been acquired during the interaction can be wasted. Moreover, to provide the cases with general domain knowledge would also help to their interpretation. For instance, objective conventions about the domain can be incorporated in the cases. A suitable way to do that is to apply Knowledge Intensive CBR (KI-CBR) [2]. KI-CBR would allow agents to reason about semantic criteria instead of reasoning only about the syntactic properties of cases. How argumentation-based interactions can profit from the properties of this reasoning methodology is still an open issue that researchers must further investigate.

During the interaction, agents may need to contract commitments to reach mutually beneficial agreements. The representation and manipulation of commitments in different dialogue types is also an open issue that needs from more research. CBR can be useful to store and manage such commitments. For instance, the CBR cycle can be used to decide if an agent should contract a commitment that could come into conflict with previous commitments. Reasoning about past commitments can be also useful to determine, in view of the final result obtained from the fulfilment of past commitments, if contracting a similar commitment is really worth. Other interesting use of CBR can be to provide warrants about the agent capability to make a commitment by showing similar commitments that it kept in the past.

Finally, important issues to research in case-based argumentation in MAS are case-base consistency matters. Under what conditions a change in the environment or the input/output of agents in the dialogue can affect to the case-base contents? If the case-base stores arguments, how do these changes could affect to their validity?

2.2 Case-Based Argument Management

In multi-agent settings, arguments are information that an agent send to other agent to convince it about some specific issue. Essential elements that differentiate argumentation-based multi-agent frameworks from conventional frameworks are their ability to generate arguments, to select the best outgoing argument from the set of possible candidates and to evaluate incoming arguments.

A possible way to generate arguments is in fact do not create them on purpose, but using pieces of knowledge about the agent, the world, or the target of the discussion. This information may be obtained from the knowledge stored in the agent case-base. In [15] arguments are pieces of information that the initiator agent sends to the responder in order to surpass its resource negotiation threshold and convince it to share the resource. The CBR manager obtains them by comparing the case descriptors and the current situation and retrieving information from the most similar case of the case-base. Following the same approach, in [10] both specific cases and justified predictions are used as three argument types: justified predictions, counterarguments and counterexamples. A *justified prediction* is a tuple $\langle A, P, S, D \rangle$ which is generated by the agent A to specify that it believes that the correct solution for the problem P is S and the evidence that supports this decision is D . The meaning of D is that all or most of the cases that are similar to P in the case-base of A belong to the predicted

solution S . A justified prediction can be generated using CBR in combination with any learning method, like decision trees and *LID* [4]. A *counterargument* is a justified prediction $\langle A, P, S', D' \rangle$ that generates an agent to rebut an argument generated by other agent by showing its evidence D' for believing that the correct solution for the problem P is S' and not S . A *counterexample* is a case that an agent sends to support a counterargument (it classifies P as belonging to a different solution S').

A different way to generate arguments is to explicitly create them by using some logic language and Argumentation Theory concepts, as *argument schemes* and *critical questions* [0]. An argument scheme is a pattern that captures the reasoning in an argumentation dialogue. Arguments that instantiate its associated critical questions are counterarguments that refute it. *CARREL+*, for instance, uses this approach and characterises completely the space of possible arguments in its *Argument Scheme Repository (ASR)* [18]. Arguments in *CARREL+* are instantiations of the argument schemes and critical questions of the ASR. The agents build them by using a first order logic programming language [9]. The case-base of the MA is structured by means of these argument schemes and critical questions. Therefore, CBR is also used to provide additional arguments that were relevant in similar previous dialogues.

Once the candidate outgoing arguments have been generated, the next step is to select the best argument to achieve the agent intentions. A typical approach for selecting arguments is to rank them with respect to some preference order. In [15], for instance, the CBR manager of each agent uses domain specific rules to order its arguments from more to less usefulness to convince the responder agent in each step of the negotiation. This process can also be done concurrently with the argument generation, implicitly with the reasoning cycle of the CBR system, as happens in [10].

Finally, the arguments submitted must be evaluated in order to decide the result of the interaction (if the sender has convinced the receiver or not). To perform the evaluation, objective considerations (conventions about the world) and subjective considerations (intentions and preferences) can be borne in mind [11]. Combining both considerations the responder agent in [15] retrieves the best negotiation strategy from its case-base and adapts it using a *relaxed constraint satisfaction* approach. Then, it uses the information gathered from the past negotiation to evaluate the evidence support of the arguments (pieces of information) that it receives.

In [10] agents use subjective criteria, based on a *confidence measure* for each justified prediction, to evaluate two contradicting arguments in accordance with a *preference relation*. To compute the confidence on a justified prediction, an agent calculates the number of counterexamples and cases that support it in its case-base. The more the supporting cases, the higher the preference for an argument.

In *ProClaim* [18] the MA uses its objective considerations, which are based on previous similar argumentations, to evaluate the current *argument graph*. When there are arguments that attack each other, the MA decides the winner. To perform this task, it compares the argument graph with those stored in its case-base and retrieves the similar argument graphs. Then, it rejects those that contain non-applicable arguments by using the case descriptions and those whose evidential support falls below some threshold. The resulting set of graphs is merged in a solution graph that constitutes the CBR proposed decision about the viability of an organ transplantation. This solution graph provides evidential support to determine the winning arguments and thus, the decision validity of the current argument graph. Similarly, in [7] CBR is

used to warrant arguments by retrieving previous argumentation processes whose arguments apply to the current dialogue.

Challenges. There is a need to provide standardised ways to manage arguments in MAS by using CBR. Over the last years, the power of explanations to make the reasoning process more understandable has gained an increasing interest in the CBR research community [16]. The CBR explanation techniques can be applied to generate and evaluate arguments in MAS. For instance, explanations could be used as arguments to support a specific solution derived from the agent case-base. Some preliminary work was done in [10], but it only applies to classification domains. Regarding argument evaluation, an agent could check if the reasoning process that generated an argument was correct. Moreover, if arguments are stored in cases, KI-CBR [2] would facilitate their evaluation by showing their relations with other arguments.

Other open issue is how to adapt the information of the case-base to fit the current situation and, thus, infer valid arguments from it. This adaptation has been performed by using domain specific rules in [15], but due to the dynamicity of argumentation domains these rules may become obsolete quickly. Further research must investigate new adaptation techniques that dissociate this process from the specific domain.

Moreover, CBR provides agents with the capability of learning from the behaviour of their partners. Future work must specify how to profit this information and perform a strategic argumentation by generating and selecting arguments that would easily persuade specific agents. Other open issues are those related with trust and reputation. How can agents assure the truthfulness of the knowledge that are considering to store in the case-base? Should an agent believe the information of its cases?

2.3 Case-Based Guide of the Interaction Protocol

The *interaction protocol* makes use of the communication language to define the interaction between the agents and specify at each time which agent can communicate with other and what is allowed to say. CBR has been applied to conduct the interaction protocol across its possible phases.

In [15] the interaction protocol is explicitly defined by means of a *finite-state machine*. However, the specific stages of the interaction are influenced by the information of the case-base. Before the start of the negotiation, the CBR manager retrieves the negotiation strategy by comparing the descriptors of the cases with the agent current state and the world state. The negotiation strategy determines the maximum time that the agent has before the objective moves to other area. Similarly, the stages of the interaction protocol in [10] depend on the arguments inferred from the case-bases. The agents use the *AMAL* protocol to interact. At each round an agent can either to *assert* an argument or to *rebut* it with a counterargument (or a counterexample). If the agents arrive to a consensus in the joint prediction, the interaction protocol ends, otherwise a weighted vote chooses the final solution.

In [18] the MA directs the RA's possible moves at each stage of the deliberation. The ASR encodes the full argumentation space and guides agents to explore possible dialectical moves. Given a submitted argument *A* instantiating an argument scheme, the MA can reference the ASR in order to identify the schemes that constitute an attack on *A*. As it was pointed out before, the MA uses CBR to evaluate the

conflicting arguments. As result of this evaluation, the organ viability is determined or the RA's are asked to provide more arguments.

Challenges. The interaction protocol also needs additional rules to govern, for instance, the admissions or withdrawals of new agents into the dialogue, the proposals validity, the fulfilment of the commitments contracted and the dialogue termination. Future work must investigate how CBR can make use of the knowledge acquired from the experience of past dialogues to take these decisions. For example, when deciding under which conditions the interaction must finish, the maximum time to interact could be delimited using the information of past dialogues that ended in disagreement because they took too much time. Moreover, the necessary time to reach an agreement could be inferred from the time that took to reach past similar agreements.

Some rules over the CBR reasoning cycle can also avoid infinite dialogues. Agents are not allowed to send the same argument or counterargument to the same agent in [10], for instance. Finally, CBR can also be used to warrant the argumentation success by stopping or readdressing current interactions when they are very similar to past experiences that ended in an unprofitable agreement.

3 Concluding Remarks

Recently, the application of Argumentation Theory techniques to assist interaction in MAS has gained an increasing interest. In this area, there are still many issues to cope with. CBR can be used to tackle some of such issues, since it has proved its suitability to manage argumentation in many precedent-based legal systems. The aim of this paper is to clarify the current contributions of CBR to the development of argumentation in MAS, to identify open issues and to provide new ideas to face future challenges. The three main application areas of CBR to argumentation in MAS are: as knowledge resource, as argument manager and as guide of the interaction protocol.

However, the adaptation of argumentative CBR to argumentation-based interaction in MAS is still in its early stages. Currently, there are common assumptions about the agents of argumentative frameworks that could no longer apply in real scenarios. Usually, the agents are thought to be always cooperative, to be reliable, to share the same ontology and to act rationally. These features could not reflect the behaviour of agents in open environments. More research must be done to adapt argumentation in MAS to interested agents or agents coming from other platforms. Therefore, CBR must be able to deal with information from different domains and uncertain veracity. A major advantage of using CBR for argumentation in MAS is that agents can learn from past dialogues by storing in their case-bases the knowledge acquired during the interaction. There is many work to do in order to determine which type of information is useful to store, how to avoid redundancy or how to deal with huge case-bases. Other important issues to manage are privacy and security in case-bases.

Acknowledgements. This work was partially supported by the Valencian Government through project GV06/315 and the Spanish Government and FEDER funds through projects TIN2005-03395 and TIN2006-14630-C03-01.

References

1. A. Aamodt and E. Plaza, Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, vol. (7)1: 39-59, 1994.
2. A. Aamodt, Knowledge-intensive case-based reasoning in Creek. In *Proceedings of the 7th European Conference (ECCBR-04)*, pp. 1-15, 2004.
3. V. Aleven and K.D. Ashley, Teaching Case-Based Argumentation Through a Model and Examples, Empirical Evaluation of an Intelligent Learning Environment. In *Proc. of the 8th World Conf. of the Artificial Intelligence in Education Society*, pp. 87-94, 1997.
4. E. Armengol and E. Plaza, Lazy induction of descriptions for relational case-based learning. In *Proc. of the Euro. Conf. on Machine Learning (ECML-01)*, pp. 13-24, 2001.
5. K.D. Ashley, Reasoning with Cases and Hypotheticals in Hypo. *International Journal of Man-Machine Studies*, vol. (34): 753-796, 1991.
6. T.J.M. Bench-Capon, P.E. Dunne, *Argumentation in Artificial Intelligence*, *Artificial Intelligence*, doi: 10.1016/j.artint.2007.05.001, 2007.
7. N. Karacapilidis, et al., Using Case-Based Reasoning for Argumentation with Multiple Viewpoints, In *Proc. of the 2nd Int. Conf. on CBR (ICCB-97)*, pp. 541-552, July 1997.
8. N. Karacapilidis and D. Papadias, Computer supported argumentation and collaborative decision-making: the HERMES system, *Information Systems*, vol. 26(4): 259-77, 2001.
9. S. Modgil, et al., Towards formalising agent argumentation over the viability of human organs for transplantation. *Proc. Mexican Int. Conf. on AI (MICAI 05)*, pp 928-938, 2005.
10. S. Ontañón and E. Plaza, Learning and Joint Deliberation through Argumentation in Multi-Agent Systems, In *Proceedings of AAMAS-07*, 2007.
11. I. Rahwan, et al., Argumentation-based negotiation. *Knowledge Engineering Review*, 18(4):343-375, 2003.
12. I. Rahwan, Guest Editorial: Argumentation in Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, 11(2): 115-125, 2005.
13. E.L. Rissland and D.B. Skalak, CABARET: Rule Interpretation in a Hybrid Architecture. *International Journal of Man-Machine Studies*, vol. (34): 839-887, 1991.
14. E.L. Rissland et al., Bankxx: a program to generate argument through case-based search. In *Proc. of the Int. Conf. on AI and Law (ICAIL-93)*, pp 117-124, 1993.
15. L-K. Soh and C. Tsatsoulis, Agent-Based Argumentative Negotiations with CBR, *AAAI Fall Symposium on Negotiation Methods for Autonomous Cooperative Sys.*, 16-25, 2001.
16. F. Smo, et al., Explanation in Case-Based Reasoning: Perspectives and Goals. *Artificial Intelligence Review*, vol. (24) 2: 109-143, 2005.
17. K. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, vol. 28, 1990.
18. P. Tolchinsky et al., CBR and Argument Schemes for Collaborative Decision Making. In *Proceedings of COMMA-06*, vol. (144): 71-82. IOS Press, 2006.
19. J. Vázquez-Salceda et al. The Organ Allocation Process: A Natural Extension of the Carrel Agent-Mediated Electronic Institution, *AI Communications*, vol. 16(3): 153-165, 2003.
20. D.N. Walton, E.C.W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, Albany, NY, USA, 1995.
21. D.N. Walton, *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1996.

Case-Base Maintenance in an Associative Memory Organized by a Self-Organization Map

A. Fornells and E. Golobardes

Grup de Recerca en Sistemes Intel·ligents
Enginyeria i Arquitectura La Salle, Universitat Ramon Llull
Quatre Camins 2, 08022 Barcelona, Spain
{afornells, elisabet}@salle.url.edu
<http://www.salle.url.edu/GRSI>

Abstract. Case-Based Reasoning (CBR) systems solve new problems using others which have been previously resolved in a case memory, where each case represents a solved situation. Therefore, the case memory size and its organization influences on the computational time needed to solve new situations. For this reason, we organize the memory using a Self-Organization Map for defining patterns to allow system to do a selective retrieval using only the cases of the most suitable pattern. This works presents a case-based maintenance to incrementally introduce knowledge in SOM without retraining it because this process is very expensive in terms of computational time. The strategy is semi-supervised because we use the feedback provided by the expert and, at the same time, the self-organization of cases when clusters are readjusted. Results show a successful case-based maintenance.

Keywords: Semi-Supervised, Case-Base Maintenance, Case-based reasoning, Self-Organization Map, Soft-Computing.

1 Introduction

Case-Based Reasoning (CBR) [1] is an analogical approach based on exploring a case memory of previously solved problems (cases) to propose a new solution. The greater the case memory size, the greater the time needed to find the most suitable cases. Therefore, the role of case memory organization is a crucial issue to improve the retrieval time by leading the system only to the potentially useful cases according to the new problem.

We focus on a case memory organization by means of a Self-Organization Map (SOM) [2] in the SOMCBR (Self-Organization Map in a Case-Based Reasoning system) [3] approach. SOM is one of the most well-known and used techniques for visualization and clustering of high-dimensional data [4, 5]. It converts complex and nonlinear statistical relationships between high-dimensional data into simple geometric relationships on a low-dimensional space. In other words, it is able to group similar cases in clusters represented by a pattern. These patterns allow SOMCBR to do a selective retrieval based on using only the cases from the clusters where their pattern is similar to the new case. Thus, the computational time is reduced, obtaining a meaningful property for real time environments [6]. Moreover, the Soft-Computing

capabilities of SOM allow systems to manage better complex and uncertain domains than other approaches based on Hard-Computing techniques [7, 8].

The goal of this paper is to present an incremental and semi-supervised strategy for the case-based maintenance in SOMCBR. CBR systems learn from their experience through new solved cases validated by an expert. Nevertheless SOM was not conceived to allow the supervised introduction and deletion of data, it was defined to group data from scratch using an unsupervised process. Consequently, the knowledge maintenance in SOMCBR implies the retraining of SOM, which is an important drawback in real time environments due to its high computational cost.

The contribution does not focus on implementing one of the many CBR case-based maintenance policies. It focuses on (1) adding, (2) deleting and (3) reorganizing the cases without retraining SOM and still maintaining the benefits of SOMCBR: computational time improvement without accuracy rate reduction respect to a flat CBR system¹. This strategy is successfully tested and compared to a flat CBR system.

The paper is organized as follows. Section 2 briefly reviews SOMCBR. Section 3 proposes the case-based maintenance strategy. Section 4 presents experimentation and results. Section 5 summarizes some related work about SOM. Finally, section 6 ends with conclusions and further work.

2 Previous Work: SOMCBR

SOMCBR [3] is a hybrid system where the case memory of a CBR system is organized by means of a SOM to provide an efficient access to cases. As Fig. 1 shows, SOM projects the original space from an input layer of N neurons (as many neurons as features) to an output layer of a new space with less dimensions (it is usually a grid of $M \times M$ neurons) with the aim of identifying groups of similar cases. Each group or cluster is represented by a director vector of N dimensions v_m .

When a new input case c_i is introduced in the input layer, each neuron m from the output layer computes a degree of similarity between its v_m and c_i applying a metric such as for example the complement of the normalized Euclidean distance (see Eq. 1). A value closer to 1 means the cases from m should be similar to c_i . This allows SOMCBR to do a selective retrieval based on looking for the most suitable clusters, and next, using only the cases from them. The reduction of cases improves the retrieval time, but it may imply a degradation of the accuracy rate if clusters are not enough representatives. This issue is related to the data complexity² [7, 8].

$$\text{similarity}(c_i, m) = \left| 1 - d(\overline{c_i}, \overline{m}) \right| = \left| 1 - \sqrt{\frac{\sum_{n:1..N} (c_i(n) - v_i(n))^2}{N}} \right| \quad (1)$$

¹ We refer to a flat CBR system as the one which linearly explores the memory.

² The data complexity refers to the class separability and the discriminant power of features, and not about its representation in the case memory.

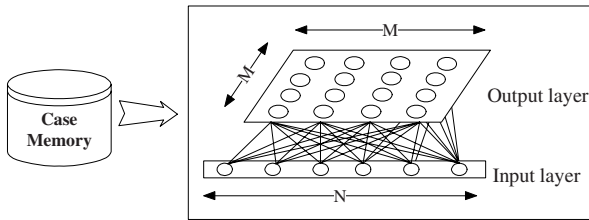


Fig. 1. SOM organizes the case memory by grouping similar cases in a cluster

3 An Incremental and Semi-supervised Case-Based Maintenance

This section presents an incremental and semi-supervised case-based maintenance with the aim of adding, deleting and reorganizing cases without retraining SOM. A new input case can be right, wrong, or no classified. This result is related to the capability of finding the most suitable cases. If SOMCBR is not able to do a right classification (we suppose that the similarity function is well defined), it could be at least because of one of the next reasons: there are not any similar cases in the case memory, the clusters with similar cases have not been selected, or there are noisy or uncertain cases. According to these reasons, the system has to decide how to update the memory using the feedback from the expert to allow a successful future classification. It is a semi-supervised case-based maintenance because cases are introduced or deleted according to the feedback provided by the expert, and self-organized when the clusters are updated. The next points describe how to tackle each one of the last situations, and finally, the global strategy is summarized in Fig. 2.

Addition of new knowledge. There are situations in which the case memory does not contain knowledge of some part of the domain. The solution is to introduce the bad solved or no classified cases in the case memory for allowing a future successful prediction. Next, the director vectors have to be adjusted with the new knowledge.

Detection and deletion of noisy or uncertain cases. Sometimes the system retrieves potentially similar cases to the new case, but in fact, they are not similar. If we assume that the similarity function is well defined, the problem is a consequence of the noisy or the uncertainty of the retrieved cases. A way of tackling this issue is by controlling through a counter how many times each case has produced a bad classification. Thus, if the counter has a greater value than a certain threshold (φ), the case can be considered as a candidate to be deleted. In contrast, the counter is set again to zero if the case produces a right classification.

Promoting the right way. There are scenarios in which the selected cluster does not contain the most similar cases, even though they exist in the case memory. This can be produced due to the data complexity, which can hinder the detection of the most suitable cluster. In this situation there is no need of adding this new case because this knowledge already exists in the case memory. The solution is to adjust the organization to fix this lack of precision, it means, to update the director vectors by looking which one is the real most suitable cluster.

Let c_i be the new input case. Let c_1, c_2, \dots, c_K be the set of the K -Nearest Neighbor (K -NN) cases of c_i . Let m be one of the $M \times M$ clusters. Let θ_j be a vector where each

position contains the degree of belonging between the case j and each one of the clusters, which can be computed by Eq. 1.

If c_i is similar to the K -NN retrieved cases, its expected θ_i should be similar to θ_k of them. Thus, its value can be estimated as the weighting mean of θ_k using the K -NN cases as Eq 1 shows. The weight introduced by the similarity between c_i and each one of the cases is important because the relations between c_i and the K -NN cases are different. Moreover, θ_i is normalized by dividing its value by the sum of the similarities. The greatest $\theta_i(p)$ value indicates the expected clusters where c_i would be mapped. Next, the p director vector is updated to improve the cluster selection.

$$\theta_i = \frac{\sum_{c_k \in K-NN} [\text{similarity}(c_i, c_k) \cdot \theta_k(m)]}{\sum_{c_k \in K-NN} \text{similarity}(c_i, c_k)}, \forall m \in M \times M \quad (2)$$

The adjustment of the director vectors and their consequences. All the clusters are related between them due to the fact that SOM represents the original data topology. Thus, if a cluster is updated, their neighbors (usually 6 or 8 according to the architecture) need also to be updated to maintain their relationships. Nevertheless, not all the relations between neurons are equal. For this reason, the updating process has to be influenced by the degree of proximity between clusters, which can be computed by Eq. 1 but using two models instead of a model and a case. In contrast, cases are associated to clusters according to the proximity respect to the director vectors. Therefore, its update can imply that some case may be automatically reorganized.

- 1 Let c_i be the new input case
- 2 Let s be the most similar cluster of c_i
- 3 Let $\alpha(t)$ be the learning factor of the SOM training
- 4 Let \bar{v}_j be the director vector of a model j
- 5 **forall** $c_k \in$ the set of the K -NN retrieved cases **do**
- 6 **if** c_k has the same class as c_i **then** Set to zero the c_k counter
- 7 **else** Increment one unit the c_k counter
- 8 Remove c_k from the case memory if its counter is equal to γ
- 9 **if** c_i was bad or non classified **then**
- 10 Look for the K -NN cases of c_i that belong to the same class
- 11 **if** there are not K -NN cases **then** Add c_i in the cluster s
- 12 **else** The expected cluster s is the highest θ_i
- 13 **forall** $\bar{v}_j \in$ the eight direct neighbors of s **do**
- 14 $\bar{v}_j(t+1) = \bar{v}_j(t) - \alpha(t) \cdot \text{similarity}(\bar{v}_s, \bar{v}_j) \cdot (\bar{c}_i - \bar{v}_j(t))$
- 15 $\bar{v}_s(t+1) = \bar{v}_s(t) + \alpha(t) \cdot (\bar{c}_i - \bar{v}_s(t))$
- 16 **forall** c_r cases from s and its eight direct neighbors **do**
- 17 Evaluates if c_r fits better in another cluster

Fig. 2. The incremental and semi-supervised case-based maintenance strategy

4 Experiments, Results, and Discussion

This section evaluates if the incremental and semi-supervised case-based maintenance allows the system to successfully (1) introduce new cases, (2) detect and deleting noisy and uncertain cases, and (3) reorganize cases of clusters without retraining SOM and, at the same time, maintaining the benefits of SOMCBR with regard to a flat CBR system. Because performance is closely related to data typology, the experimentation is done using several datasets of different domains and characteristics from the UCI Repository [9] as Table 1 shows.

Figure 3(a) shows the evolution of the SOMCBR error rate (●) when the percentage of test cases is increased from 10% to 90% and using an aggressive configuration for erasing the potentially noisy cases ($\phi=2$). These values are the mean of 10 executions using different random seeds and applying a 10-fold stratified cross validation. The normalized Euclidean distance (see Eq. 1) is used as similarity function, K -NN is set to 5 which is a confidence value, and the map size is fixed and maintained to 3×3 because at the beginning there are few cases. Next, we compare its performance to a flat CBR system (+) using the same criteria for adding and removing cases from the case memory in several scenarios. In each case, a different proportion of percentage of train and test cases are used to analyze the impact of the incremental approach over the case-base maintenance.

We can observe that error rate in both approaches is almost the same while the computational time (□) is drastically improved in SOMCBR. It means that knowledge is successfully managed in SOM because strategy is able to readjust the director vectors to better fit to the new data. Another aspect to highlight from Fig. 3(a) is that SOMCBR is able to classify the same percentage of cases but having a more reduced case memory as the line with \triangle shows, which is the mean percentage of case memory reduction between the SOMCBR and CBR. This is produced thanks to the SOM capabilities of defining patterns, which allows the system to detect redundant cases and not to add them in the case memory.

Table 1. Description of the datasets sorted by the number of instances

Name	Cases	Attributes	Classes	Name	Cases	Attributes	Classes
iris	150	4	3	wdbc	569	30	2
hepatitis	155	19	2	bal	625	4	3
ionosphere	155	19	2	wbcd	699	9	2
wine	178	13	3	wisconsin	699	9	2
wpbc	198	33	2	pim	768	8	2
sonar	208	60	2	vehicle	846	18	4
glass	214	9	6	tao	1888	2	2
heartstatlog	270	13	2	segment	625	4	3
iris	150	4	3	wdbc	2310	19	7
Bpa	345	6	2	Waveform	5000	40	3

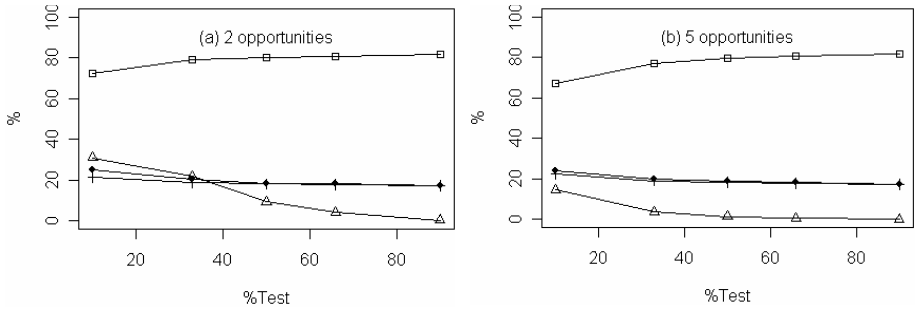


Fig. 3. Charts show the performance of the retain strategy using two criteria for removing noise cases (2 and 5 times). They represent the evolution of the mean percentage of (1) the error for a flat CBR system (+) and for the SOMCBR system (●), (2) the mean case memory size reduction between both (△) and (3) the mean percentage of reductions in the cases retrieved when SOMCBR is applied (□). Each one of these values is represented for different percentage of test cases in the x axis.

On the other hand, Fig. 3(b) provides the same results but using a more conservative deletion policy ($\varphi=5$). The main difference with respect to Fig. 3(a) is the case memory size. In this scenario, more errors are needed to erase a case. Although the aggressive configuration provides better results than the conservative, the selection of the φ value depends on the domain typology, its level of imprecision, and the system memory available.

Therefore, the application of the case-based maintenance is successful because the benefits of SOMCBR are maintained with respect to a flat CBR system.

5 Related Work: Incremental and Unsupervised Learning in SOM

There are different variants of SOM which allow the introduction of new knowledge but using an unsupervised process. They are based on dynamically update the network topology by adding or removing neurons using a defined function of error. The main difference between them is the structure of the neurons, and the criteria to update them when the new knowledge appears.

GCS (Growing Cell Structure) [10] and GNG (Growing Neural Gas) [11] define a structure composed by hypertetrahedrons of a dimensionality previously selected. A hypertetrahedron is the most simple polyhedron in a D -dimension. For example, the hypertetrahedron for d equal to 1, 2 and 3 are lines, triangles and tetrahedrons. The model is initialized with one hypertetrahedron, where the edges join the neurons. The adaptation process consists in introducing new neurons after λ steps. First, the neuron with the maximum accumulated error is found (q). Next, the longest edge which joins q with another neuron (p) is split in two segments, which are joined by the new neuron. The main difference between them is the connection of neurons. GCS forces that the new neuron is connected to all the neighbors of p and q to guarantee the global structure composed by hypertetrahedrons. In contrast, GC uses the Hebbian

learning [12] to join by an edge the first and the second winning neuron. Consequently, GCS has a regular structure, while GNG not. Although this last issue allows GNG to reflect best the data topology, it is difficult for the data visualization due to the non-uniformity of the dimensionality.

Alternatively, GG (Growing Grid) [13] and GSOM (Incremental Self-Organizing Network) [14] are based on hyper-rectangular structures. CG is like GCS, but it forces the addition of hyper-rows or hyper-columns to enforce a hyper-rectangular structure on the graph. In contrast, GSOM inserts the new neurons in the center of the current topology instead of the model with the maximum accumulated error, and it also automatically chooses a dimensionality during the growth process.

In contrast, ILM [15] defines a grid of two dimensions as in the SOM algorithm. The topological update is based on expanding the neurons situated on the perimeter of the grid when the relation between the numbers of elements is greater than the mean number of elements by neuron. Next, it applies an ascendant hierarchical clustering for optimizing the final number of neurons.

Finally, there is other work focused on the use of topology preserving mapping models and CBR [16].

6 Conclusions and Further Work

This paper has presented an incremental and semi-supervised strategy for the case-based maintenance in SOMCBR. The goal is to (1) introduce new cases, (2) detect and delete noisy cases, and (3) reorganize them without retraining the SOM because, it is very expensive in terms of computational time. The main difference of this approach with respect to the others is that this approach uses the feedback provided by the expert to decide when it is necessary to add, delete or readjust the patterns. The impact of the case-based maintenance for managing the new experience has been studied by comparing the solving capabilities of SOMCBR with respect to a flat CBR system using the proposed learning strategy.

The application of the strategy has been successful so SOMCBR has been able to learn from the new experience while its performance with respect to a flat CBR system has been maintained: improvement of the computational time while the accuracy rate is maintained. Moreover, the results show how SOMCBR is able to detect redundant cases thanks to the SOM capability for defining patterns. Thus, the redundant cases are not added and the case memory is smaller.

Further work is focused on using the feedback provided by the expert to update the initial fixed architecture to an extended version, and fit it better to the data topology and the distribution of cases in the clusters.

Acknowledgments. We would like to thank the Spanish Government for the support under grant TIN2006-15140-C03-03 and the *Generalitat de Catalunya* for the support under grants 2005SGR-302 and 2007FIC-0976. Also, we would like to thank *Enginyeria i Arquitectura La Salle* of Ramon Llull University for the support to our research group.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundations issues, methodological variations and system approaches. *AI Communications*, Vol. 7 (1994) 39–59.
2. Kohonen, T.: *Self-Organization and Associative Memory*. Springer Series in Information Sciences. Vol. 8. Springer Berlin Heidelberg (1989).
3. Fornells, A., Golobardes, E., Vernet, D., Corral, G.: Unsupervised case memory organization: Analysing computational time and soft computing capabilities. In 8th European Conference on Case-Based Reasoning. *LNAI*. Vol. 4106. Springer-Verlag (2006) 241–255.
4. Kaski, S., Kangas, J., Kohonen T.: Bibliography of Self-Organizing Map (SOM) Papers: 1981-1997. <http://www.cis.hut.fi/research/refs/> (1998).
5. Oja, M., Kaski, S., Kohonen, T.: Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001. <http://www.cis.hut.fi/research/refs/> (2003).
6. Fornells, A., Golobardes, E., Vilasf, X., J. Martí Integration of strategies based on relevance feedback into a tool for retrieval of mammographic images. In 7th International Conference on Intelligent Data Engineering and Automated Learning. *LNCS*. Vol. 4224. Springer-Verlag (2006) 116–124.
7. Fornells, A., Golobardes, E., Martorell, J.M., Garrell, J.M., Bernadó E., Macià N: Measuring the applicability of self-organization maps in a case-based reasoning system. In 3rd Iberian Conference on Pattern Recognition and Image Analysis. *LNCS*. Vol. 4478. Springer-Verlag (2007) 532–539.
8. Fornells, A., Golobardes, E., Martorell, J.M., Garrell, J.M., Bernadó E., Macià, N.: A methodology for analyzing the case retrieval from a clustered case memory. In 7th International Conference on Case-Based Reasoning, *LNAI*. Springer-Verlag (2007) *In press*.
9. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998).
10. Fritzke, B.: Growing cell structures - a self organizing network for unsupervised learning. *Neural Networks*. Vol 7, Number 9 (1994)1441–1460.
11. Fritzke, B.: Growing self-organizing networks, why? In *ESANN'96: European Symposium on Artificial Neural Networks* (1996) 61–72.
12. White, R. H.: Competitive hebbian learning: Algorithms and demonstrations. *Neural Networks*. Vol 5, Number 2 (1992) 261–275.
13. Fritzke, B.: Growing grid - a self organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*. Vol 5, Number 2 (1995) 9–13.
14. Bauer, H., Villmann, T.: Growing a hypercubical output space in a self-organizing feature map. *IEEE Trans. on Neural Networks*. Vol 8, Number 2 (1997) 218–226.
15. Benabdeslem, K.: Hybrid neural system for time series prediction. In 28th International conference on information technology interface. *IMAC/IEEE* (2006) 349–354.
16. Corchado, E., Corchado, J.M., Aiken, J.: IBR retrieval method based on topology preserving mappings. *Journal of Experimental & Theoretical Artificial Intelligence*. Vol. 16, Number 3 (2004) 145-160.

Hybrid Multi Agent-Neural Network Intrusion Detection with Mobile Visualization

Álvaro Herrero¹, Emilio Corchado¹, Mará A. Pellicer¹, and Ajith Abraham²

¹ Department of Civil Engineering, University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
{ahcosio, escorchado}@ubu.es

² Norwegian University of Science and Technology, Norway
ajith.abraham@ieee.org

Abstract. A multiagent system that incorporates an Artificial Neural Networks based Intrusion Detection System (IDS) has been defined to guaranty an efficient computer network security architecture. The proposed system facilitates the intrusion detection in dynamic networks. This paper presents the structure of the Mobile Visualization Connectionist Agent-Based IDS, more flexible and adaptable. The proposed improvement of the system in this paper includes deliberative agents that use the artificial neural network to identify intrusions in computer networks. The agent based system has been probed through anomalous situations related to the Simple Network Management Protocol.

Keywords: Multiagent Systems, Artificial Neural Networks, Unsupervised Learning, Projection Methods, Computer Network Security, Intrusion Detection.

1 Introduction and Previous Work

As it is known, the rapid growing of computer networks and the interconnection among them has entailed some security problems. New security failures are discovered everyday and there are a growing number of bad-intentioned people trying to take advantage of such failures. It is unquestionable that organizations need to protect their systems from these intruders and consequently, new network security tools are being developed. The most widely used tool of this kind is firewalls but Intrusion Detection Systems (IDSs) are becoming more and more popular. They monitor the activity of the network with the purpose of identifying intrusive events and can take actions to abort these risky events.

A wide range of techniques have been used to build IDSs. On the one hand, there have been some previous attempts to take advantage of agents and Multiagent Systems (MAS) in the field of Intrusion Detection (ID), as for example [1], [2], [3]. It is worth mentioning the mobile-agents approach [4], [5]. On the other hand, some different machine learning models – including Data Mining techniques and Artificial Neural Networks (ANN) – have been successfully applied for ID, as for example [6], [7], [8], [9], [10].

Additionally, some other AI techniques have been combined (such as genetic algorithms and fuzzy logic [11], genetic algorithms and K-Nearest Neighbor (K-NN)

[12] or K-NN and ANN [13] among others) in order to face ID from a hybrid point of view. In some cases they provide intelligence to MAS. This paper proposes the use of a dynamic multiagent architecture employing deliberative agents capable of learning and evolving with the environment. Some of the agents contained in this architecture are known as CBR-BDI agents [14] because they integrate the BDI (Believes, Desires and Intentions) model and the Case-Based Reasoning (CBR) paradigm. These agents may incorporate different identification or projection algorithms depending on their goals. In this case, an ANN will be embedded in such agents to perform ID in computer networks.

The use of embedded ANN in the deliberative agents of a dynamic MAS let us take advantage of some of the properties of ANN (such as generalization) and agents (reactivity, proactivity and sociability) making the ID task possible. The overall architecture of the system as well as its different components are described as follows: Section 2 outlines the concept of CBR-BDI agent, section 3 describes the overall architecture of the system, and section 4 presents some conclusions and future work.

2 CBR-BDI Architecture

The Mobile Visualization Connectionist Agent-Based IDS (MOVICAB-IDS) [10], [15], [16] has been designed to detect anomalous situations taking place in a computer network. In this paper, some of the agents of the MOVICAB-IDS MAS have been implemented as CBR-BDI agents including an ANN for ID. The multiagent architecture of MOVICAB-IDS is depicted in Fig 1. CBR-BDI agents [17], [18] use CBR systems [19] as a reasoning mechanism, which allows them to learn from initial knowledge, to interact autonomously with the environment, users and other agents within the system, and to have a large capacity for adaptation to the needs of its surroundings.

MOVICAB-IDS includes as a novelty, in this paper, deliberative agents using a CBR architecture, that allows them to respond to events, to take the initiative according to their goals, to communicate with other agents, to interact with users, and to make use of past experiences to find the best information to achieve goals. The proposed CBR-BDI agents work at a high level with the concepts of Believes, Desires and Intentions (BDI) [20]. CBR-BDI agents have learning and adaptation capabilities, what facilitates their work in dynamic environments. Different models can be embedded in the four steps of the CBR reasoning process. In this work, these agents use an ANN to identify intrusions in computer networks. The following section outlines the architecture and reasoning process of the CBR-BDI agents used for ID.

3 MOVICAB-IDS Architecture

MOVICAB-IDS has been constructed as an IDS for distributed computer networks. The proposed MAS incorporates different types of agents. Some of them are reactive agents while others are CBR-BDI agents. The extended version of the Gaia methodology [21] has been applied, and some roles and protocols were identified after the Architectural Design Stage. Examples of protocols are the NegotiateAnalysis

protocol (when new data is ready for analysis, an analyzer is chosen) and the ChangeAnalysisConfig protocol (when the configuration of the analysis has been changed, the new configuration is sent to the CONFIGURATIONMANAGER). The Detailed Design Stage concluded that there is a one-to-one correspondence between roles and agent classes in this system, so the agent classes finally identified are: Sniffer, Preprocessor, Analyzer, ConfigurationManager, Coordinator and Visualizer. The outcomes of Gaia methodology [21], [22] are modelled by AUML [23]. Six agents have been developed in this study to improve previous versions of MOVICAB-IDS. They all are listed, and special attention will be placed to the ANALYZER CBR-BDI agent.

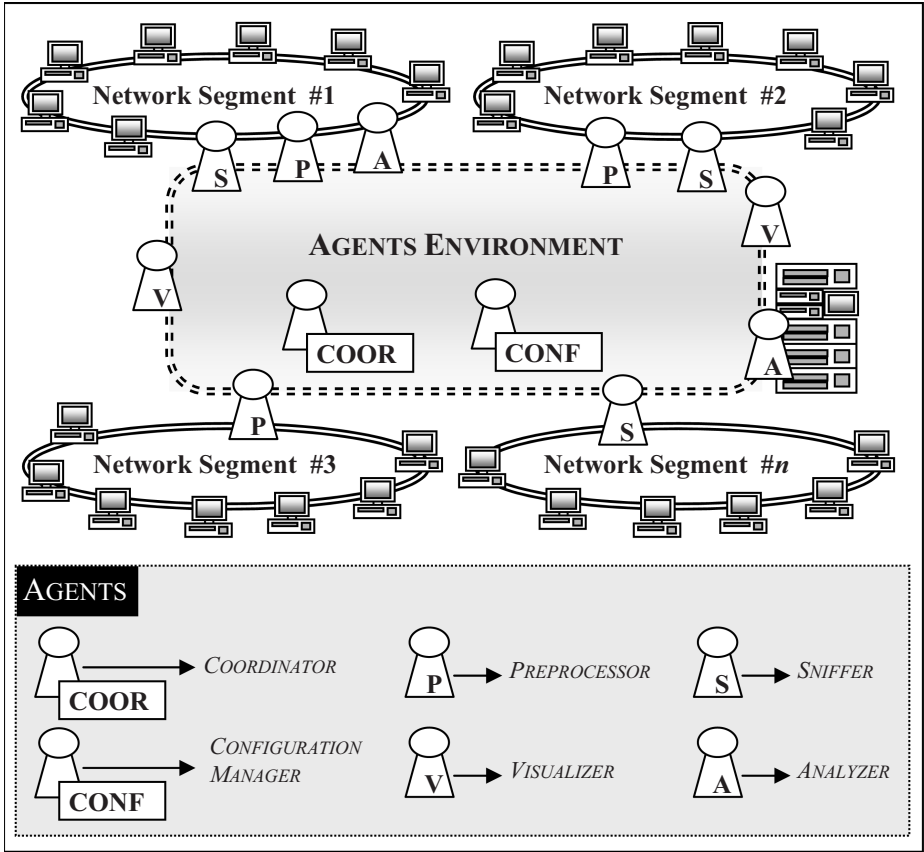


Fig. 1. MOVICAB-IDS Structure

Sniffer

This reactive agent is in charge of capturing traffic data. The continuous traffic flow is captured and split into segments in order to send it through the network for further process. Finally, the readiness of the data is communicated. One agent of this class is located in each of the network segments that the IDS has to cover (from 1 to n).

Additionally, there are cloned agents (one per network segment) ready to substitute the active ones if they fail because these agents are the most critical ones. Nothing could be done if traffic data is not captured.

Preprocessor

After splitting traffic data, the generated segments must be preprocessed to apply subsequent analysis. For the shake of network traffic, it would be advisable to locate one of these reactive agents in the same host where a SNIFFER is located. By doing so, the high-volume raw data will not travel along the network. Once the data has been preprocessed, an analysis for this new piece of data is requested. The sending of this data will not overload the network as its volume is much smaller than the one of split data.

Analyzer

This is a CBR-BDI agent. It has got embedded a connectionist model within the adaptation stage of its CBR system that helps to analyze preprocessed traffic data. This connectionist model is called Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [24]. It extends the Maximum Likelihood Hebbian Learning (MLHL) model [25] that is a neural implementation of Exploratory Projection Pursuit (EPP). This agent generates a solution (or achieve its goals) by retrieving a case and analyzing the new one using a CMLHL network. Each case incorporates several features, as can be seen in Table 1.

Table 1. Representation of case features. Classes: P (Problem description attribute) and S (Solution description attribute).

Class	Feature	Type	Description
P	Segment length	Integer	Total segment length (in ms).
P	Network segment	Integer	Network segment where the traffic comes from.
P	Date	Date	Date of capturing.
P	#source ports	Integer	Total number of source ports.
P	#destination ports	Integer	Total number of destination ports.
P	#protocols	Integer	Total number of protocols.
P	#packets	Integer	Total number of packets.
P	Protocol/packets	Array	An array (of variable length depending on each dataset) containing information about how many packets of each protocol there are in the dataset.
S	#Iterations	Integer	Number of iterations.
S	Learning rate	Float	Learning rate.
S	p	Float	CMLHL parameter.
S	Lateral strength	Float	CMLHL parameter.

As it is known, the CBR life cycle consists of four steps: retrieval, reuse, revision and retention [19]. The techniques and tools used by the Analyzer agent to implement these steps are described in the following paragraphs.

Retrieval stage: when a new analysis is requested, the ANALYZER agent tries to find the most similar case to the new one. Euclidean distance is used to find the most

similar case in the multidimensional space defined by the features characterizing each dataset (see problem description features in Table 1).

Reuse Stage: once the most similar case has been found, its solution is reused. This solution consists of the values of the parameters used to train a connectionist model (see solution description features in Table 1). This model is CMLHL [24]. It extends the MLHL model [25] that is a neural implementation of EPP. The classical statistical method of EPP [26] provides a linear projection of a data set onto a set of basis vectors which best reveal the interesting structure in data. MLHL identifies interestingness by maximising the probability of the residuals under specific probability density functions which are non-Gaussian.

CMLHL extends the MLHL paradigm by adding lateral connections [24], which have been derived from the Rectified Gaussian Distribution [27]. The resultant net can find the independent factors of a data set but does so in a way that captures some type of global ordering in the data set. Considering a D -dimensional input vector (\mathbf{x}), and an Q -dimensional output vector (\mathbf{y}), with W_{ij} being the weight (linking input j to output i), then CMLHL can be expressed as:

1. Feed-forward step:

$$y_i = \sum_{j=1}^D W_{ij} x_j ; i = 1, \dots, Q. \quad (1)$$

2. Lateral activation passing:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ ; i = 1, \dots, Q. \quad (2)$$

3. Feedback step:

$$e_j = x_j - \sum_{i=1}^Q W_{ij} y_i ; j = 1, \dots, D. \quad (3)$$

4. Weight change:

$$\Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j) |e_j|^{p-1} ; i = 1, \dots, Q ; j = 1, \dots, D. \quad (4)$$

Where: η is the learning rate, τ is the "strength" of the lateral connections, b is a bias parameter, p a parameter related to the energy function and A is a symmetric matrix used to modify the response to the data. The effect of this matrix is based on the relation between the distances separating the output neurons.

A set of trainings (for the same CMLHL model with a combination of parameter values varying in a specified range) is proposed by tacking into account the distance between the new case and the most similar one. That is, if they are quite similar, a reduced set of trainings are going to be performed. On the contrary, if the most similar case is far away from the new one, a great number of trainings are going to be generated.

Revision Stage: the CMLHL model is trained with the new dataset and the combination of parameters values generated in the reuse stage. When the new projections (the outputs of the CMLHL model for each combination) of the dataset

are ready, they are shown to the human user (the network administrator typically) through the VISUALIZER agent. The user has to choose one of these projections as the best one; the one that provides the clearest snapshot of the traffic evolution.

Retention Stage: when a projection is chosen by the user, the ANALYZER agent stores a new case containing the dataset-descriptor and the solution (parameter values used to generate this projection) in the case base for future reuse (See Table 1). ANALYZER agents share their case bases.

The ANALYZER is clearly the most resources-consuming class of agents. The amount of computational resources needed to analyze the continuous data coming from different network segments is extremely high. To overcome this demand, ANALYZER agents can be located in high-performance computing cluster or in most common machines (as can be seen in Fig. 1).

ConfigurationManager

It is worth mentioning the importance of the configuration information. The processes of data capture, split, preprocess and analysis depends on the values of several parameters, as for example: packets to capture, segment length, features to extract... All this information is managed by the CONFIGURATIONMANAGER reactive agent, that is in charge of providing this information to the SNIFFER, PREPROCESSOR and ANALYZER agents.

Coordinator

There can be several ANALYZER agents (from 1 to m) but only one COORDINATOR. The latter is in charge of sharing the analysis work out among the former. In order to improve the efficiency and perform a real-time processing, the preprocessed data must be dynamically and optimally assigned. This assignment is performed taking into account both the capabilities of the machines where ANALYZER agents are located and the analysis demands (amount and volume of data to be analysed).

Visualizer

This is an interface agent. At the very end of the process, the analyzed data is presented to the network administrator (or the person in charge of the network) by means of a functional and mobile visualization interface. To improve the accessibility of the system, the administrator may visualize the results on a mobile device (as can be seen in Fig. 2), enabling informed decisions to be taken anywhere and at any time. Depending on the platform where the information will be shown, the offered visualization facilities will be different.

4 Results, Conclusions and Future Work

MOVICAB-IDS identifies anomalous situations due to the fact that these situations do not tend to resemble parallel and smooth directions (normal situations) or because their high temporal concentration of packets. It can be seen in Fig. 2, where a port sweep situation has been identified (Group 1) and visualized in a mobile platform. On the other hand, a more advanced visualization is offered in Fig. 3 for a different data

set. In this case, it is easy to notice some different directions (Groups A and B) to the normal data ones. Also, the density of the packets is higher for these anomalous groups.

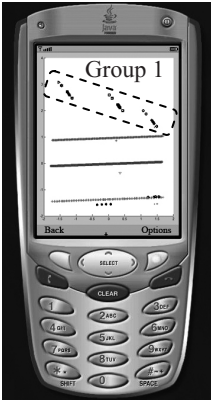


Fig. 2. Mobile Visualization

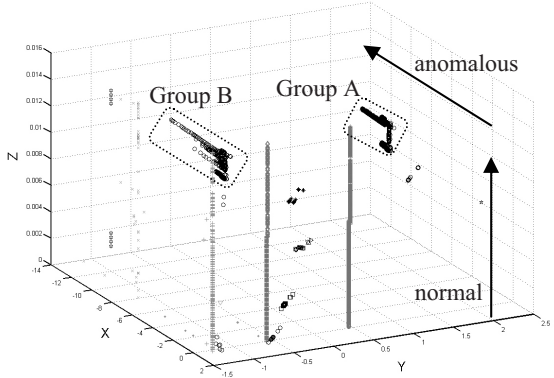


Fig. 3. Advanced Visualization

The effectiveness of MOVICAB-IDS in facing some anomalous situations has been widely demonstrated in previous works [10], [15], [16]. The projections obtained [by CMLHL overcome that obtained by Principal Component Analysis (PCA) [10], Maximum Likelihood Hebbian Learning [28] and Auto Associative Back Propagation Networks [28]. Figures 4 and 5 show a comparison of projections obtained by CMLHL (Fig. 4) and PCA (Fig. 5) for the same data. As can be easily seen, CMLHL is able to identify the port sweeps (Groups 1 and 2) contained in the dataset while PCA is not.

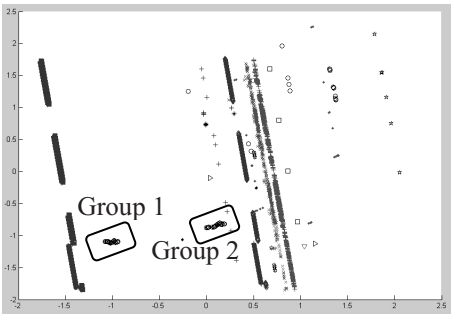


Fig. 4. CMLHL Projection

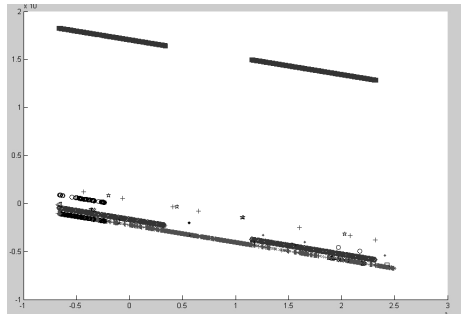


Fig. 5. PCA Projection

As a conclusion, we could say that this paper presents an improved MAS that incorporates CBR-BDI agents with embedded ANN. This improved version of MOVICAB-IDS gives the following advantages:

- Scalability: new agents (both SNIFFER and ANALYZER) can be dynamically added at any time.
- Failure tolerance: backup instances of some agents can be ready to run as soon as the working instances fail, showing a proactive behaviour.
- Real-time processing: by splitting the data and allowing the system to process it in different processing units (agents located in different machines).
- Mobile visualization: the visualization task can be performed in a wide variety of devices (as it is shown in Fig. 2).

Future work will focus on improving the dynamical assignment of analysis in order to take advantage of the computational resources in a more efficient way and on studying different distributions and learning rules for the data analysis. New neural models or adaptations of the existent ones would be tried.

Acknowledgments. This research has been partially supported by the MCyT project TIN2004-07033 and the project BU008B05 of the JCyL.

References

1. Spafford, E.H., Zamboni, D.: Intrusion Detection Using Autonomous Agents. *Computer Networks: The Int. Journal of Computer and Telecommunications Networking* 34(4), 547-570 (2000)
2. Hegazy, I.M., Al-Arif, T., Fayed, Z.T., Faheem, H.M.: A Multi-agent Based System for Intrusion Detection. *IEEE Potentials* 22(4), 28-31 (2003)
3. Dasgupta, D., Gonzalez, F., Yallapu, K., Gomez, J., Yarramsetti, R.: CIDS: An agent-based intrusion detection system. *Computers & Security* 24(5), 387-398 (2005)
4. Wang, H.Q., Wang, Z.Q., Zhao, Q., Wang, G.F., Zheng, R.J., Liu, D.X.: Mobile Agents for Network Intrusion Resistance. In: *APWeb 2006*. LNCS, vol. 3842, pp. 965-970. Springer, Heidelberg (2006)
5. Deeter, K., Singh, K., Wilson, S., Filipozzi, L., Vuong, S.: APHIDS: A Mobile Agent-Based Programmable Hybrid Intrusion Detection System. In: *Mobility Aware Technologies and Applications*. LNCS, vol. 3284, pp. 244-253. Springer, Heidelberg (2004)
6. Laskov, P., Dussel, P., Schafer, C., Rieck, K.: Learning Intrusion Detection: Supervised or Unsupervised? In: Roli, F., Vitulano, S. (eds.) *ICIAP 2005*. LNCS, vol. 3617, pp. 50-57. Springer, Heidelberg (2005)
7. Liao, Y.H., Vemuri, V.R.: Use of K-Nearest Neighbor Classifier for Intrusion Detection. *Computers & Security* 21(5), 439-448 (2002)
8. Sarasamma, S.T., Zhu, Q.M.A., Huff, J.: Hierarchical Kohonen Net for Anomaly Detection in Network Security. *IEEE Transactions on Systems Man and Cybernetics* 35(2), 302-312 (2005)
9. Zanero, S., Savaresi, S.: Unsupervised Learning Techniques for an Intrusion Detection System. In: *Proc. of the ACM Symposium on Applied Computing*. pp. 412-419 (2004)
10. Corchado, E., Herrero, A., Sáiz, J.M.: Detecting Compounded Anomalous SNMP Situations Using Cooperative Unsupervised Pattern Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) *ICANN 2005*. LNCS, vol. 3697, pp. 905-910. Springer, Heidelberg (2005)

11. Sindhu, S.S.S., Ramasubramanian, P., Kannan, A.: Intelligent Multi-agent Based Genetic Fuzzy Ensemble Network Intrusion Detection. In: Neural Information Processing. LNCS, pp. 983-988. Springer, Heidelberg (2004)
12. Middlemiss, M., Dick, G.: Feature Selection of Intrusion Detection Data Using a Hybrid Genetic Algorithm/KNN Approach. In: Design and application of hybrid intelligent systems. IOS Press. 519-527 (2003)
13. Kholfi, S., Habib, M., Aljahdali, S.: Best Hybrid Classifiers for Intrusion Detection. Journal of Computational Methods in Science and Engineering 6(2), 299 - 307 (2006)
14. Carrascosa, C., Bajo, J., Julián, V., Corchado, J.M., Botti, V.: Hybrid Multi-agent Architecture as a Real-Time Problem-Solving Model. Expert Systems with Applications: An International Journal 34(1), 2-17 (2008)
15. Corchado, E., Herrero, A., Saiz, J.M.: Testing CAB-IDS through Mutations: on the Identification of Network Scans. In: Proc. of the Int. Conf. on Knowledge-Based and Intelligent Information & Engineering Systems. LNAI, vol. 4252, pp. 433-441 (2006)
16. Herrero, A., Corchado, E., Sáiz, J.M.: MOVICAB-IDS: Visual Analysis of Network Traffic Data Streams for Intrusion Detection. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 1424-1433. Springer, Heidelberg (2006)
17. Corchado, J.M., Laza, R.: Constructing Deliberative Agents with Case-Based Reasoning Technology. International Journal of Intelligent Systems 18(12), 1227-1241 (2003)
18. Pellicer, M.A., Corchado, J.M.: Development of CBR-BDI Agents. International Journal of Computer Science and Applications 2(1), 25 - 32 (2005)
19. Aamodt, A., Plaza, E.: Case-Based Reasoning - Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1), 39-59 (1994)
20. Bratman, M.E.: Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A. (1987)
21. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing Multiagent Systems: the Gaia Methodology. ACM Transactions on Software Engineering and Methodology 12(3), 317-370 (2003)
22. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems 3(3), 285-312 (2000)
23. Bauer, B., Miller, J.P., Odell, J.: Agent UML: A Formalism for Specifying Multiagent Software Systems. International Journal of Software Engineering and Knowledge Engineering 11(3), 1-24 (2001)
24. Corchado, E., Fyfe, C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. Int. Journal of Pattern Recognition and Artificial Intelligence 17(8), 1447-1466 (2003)
25. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Mining and Knowledge Discovery 8(3), 203-225 (2004)
26. Friedman, J.H., Tukey, J.W.: A Projection Pursuit Algorithm for Exploratory Data-Analysis. IEEE Transactions on Computers 23(9), 881-890 (1974)
27. Seung, H.S., Socci, N.D., Lee, D.: The Rectified Gaussian Distribution. Advances in Neural Information Processing Systems 10, 350-356 (1998)
28. Herrero, A., Corchado, E., Gastaldo, P., Zunino, R.: A Comparison of Neural Projection Techniques Applied to Intrusion Detection Systems. In: Sandoval, F., Prieto, A., Cabestany, J., Grañ, M. (eds.) IWANN'2007. LNCS, vol. 4507, pp. 1138-1146. Springer, Heidelberg (2007)

Knowledge Extraction from Environmental Data Through a Cognitive Architecture

Salvatore Gaglio, Luca Gatani, Giuseppe Lo Re, and Marco Ortolani

Dept. of Computer Engineering, University of Palermo
Viale delle Scienze, I-90128, Palermo, Italy
{gaglio, gatani, lore, ortolani}@unipa.it

Abstract. Wireless Sensor Networks represent a novel technology which is expected to experience a dramatic diffusion thanks to the promise to be a pervasive sensory means; however, one of the issues limiting their potential growth relies in the difficulty of managing and interpreting huge amounts of collected data. This paper proposes a cognitive architecture for the extraction of high-level knowledge from raw data through the representation of processed data in opportune conceptual spaces. The presented framework interposes a conceptual layer between the subsymbolic one, devoted to sensory data processing, and the symbolic one, aimed at describing the environment by means of a high level language. The features of the proposed approach are illustrated through the description of a sample application for wildfire detection.

1 Introduction

Wireless Sensor Networks (WSNs) are an emerging technology that allows pervasive environmental monitoring through measurement of characteristic quantities [1]. Despite the difficulty of collecting and managing huge amounts of measurements, meaningful information can be extracted by means of intelligent in-network processing and correlation of sensed data. Although several interesting works present innovative applications where WSN pervasiveness has a dramatic impact in accomplishing monitoring and control tasks on well-defined scenarios, all these proposals are definitely application-specific [2, 3]. Our approach, on the other hand, aims at providing a comprehensive framework capable of general-purpose management of data streaming from the sensed environment to the highest level of knowledge representation.

Cognitive science aims at understanding how information is represented and processed in different kinds of agents, biological as well as artificial. Finding the most appropriate way of modeling the information is a challenging issue and currently two approaches are dominating: the *symbolic* approach starts from the assumption that cognitive systems can be modeled as Turing machines, whereas the *connectionist* approach models such systems through artificial neuron networks. According to Gärdenfors [4], neither approach can model complex aspects of cognitive phenomena in all situations; therefore, the author introduces a third form of representing information (*conceptual* representation), based on the exploitation of geometric structures.

This paper proposes the adoption of a three-layer cognitive architecture, similar to the one presented in [5] and [6], where the authors exploit the idea of *conceptual spaces*. In order to generalize to more challenging scenarios, we devise a hybrid architecture that employs WSNs as the sensing layer of an intelligent system. We describe the design of a flexible framework capable of collecting raw data through networked sensors, and of operating reductions and aggregations on them; processed data are then represented as vectors in ad-hoc geometric spaces, where the notion of similarity can be modeled in a natural way using opportune metrics. Meaningful concepts will finally be extracted and used for further higher-level inferences.

The rest of the paper is organized as follows. In Section 2, we introduce the cognitive architecture, discuss the functionalities of each layer, and detail our implementation choices. Section 3 illustrates how the architecture may be exploited in a sample scenario for wildfire monitoring. Finally, we draw our conclusions in Section 4.

2 The Proposed Architecture

Our architecture is composed of three layers, as schematically represented in Figure 1: the *subsymbolic layer*, where information is still deeply embedded into raw data; the *conceptual layer*, where basic quantities are represented in geometric form as points in a vector space, and finally the *symbolic layer*, where information is eventually coded as high-level symbols on which logical inferences may be carried on.

2.1 Subsymbolic Layer

The interface of the proposed architecture with the real world is represented by the Wireless Sensor Network infrastructure, which accounts for the lowest layer of the whole complex cognitive system. The subsymbolic layer exploits the computational capabilities of sensor nodes in order to implement data correlation techniques, that can lessen the burden of superfluous communication and processing. Sensors capture raw, unprocessed data, so it is crucial to extract hidden correlations in order to perceive the occurrence of unusual phenomena. Moreover, for the sake of our architectural scheme, the subsymbolic layer also needs to deal with adapting the dimensionality of the original data space to the requirements of the upper layer. In a conceptual architecture, identifying meaningful dimensions (named *quality dimensions*) is both the primary goal, and the challenging issue, when relating raw data to concepts. They represent various “qualities” of real objects and affect the way in which an opportune distance metric may be interpreted as a natural way of expressing similarity.

Our architecture assumes that sensor nodes collaborate with each other in order to exploit temporal correlation in sensed data and to build synthetical models. The approach relies on the observation that environmental phenomena are quite predictable, so they will likely present some temporal correlation that may be exploited if nodes try to establish collaborative relationships with each other. Following the original ideas of [7], this leads to the formation of clusters of sensor nodes that delegate to a representative node the task of building a predictive model

out of previous measurements. Neighboring sensor nodes help reduce data redundancy by entering into collaborative relationships and taking turns in transmitting. At any point in time, the node that is actively transmitting shares a model with each of the nodes in its group, based on their past behavior, and exchanges information with them in order to make sure that its response on their behalf meets some accuracy constraints; in the ideal case very little data needs to be exchanged without any information loss.

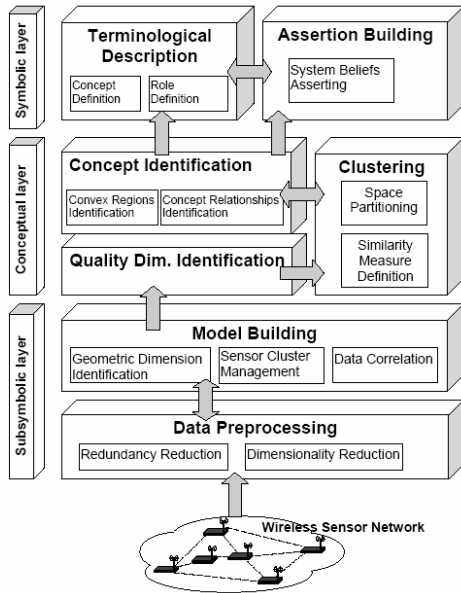


Fig. 1. The proposed architecture, showing the three levels of representation

2.2 Conceptual Layer

Environmental data gathered by the WSN are collected at a sink node, where they may be further processed and then mapped at the conceptual layer, in order to build a description of the sensed environment, in terms of a combination of geometric dimensions. Conceptual spaces can serve as tools for categorization and concept formation, providing a natural way of representing similarities, which is typically very difficult to handle at the symbolic layer. A conceptual space consists of a number of quality dimensions that can be generated by perceptual mechanisms, or can present a more abstract, non-sensory character. It is assumed that each of the quality dimensions is endowed with a certain topological or metric structure, so it is conceivable to define the similarity of two objects through the distance between their representing points in the space.

In this model, a categorization generates a partitioning of the conceptual space; Gärdenfors defines *natural concepts* as *convex regions*. However the fundamental issue concerns how to identify those regions in the space and, as those regions are supposed to group similar objects in the conceptual space together, a possible solution

requires using a *clustering* technique. The arising of meaningful clusters is obviously closely related to the choice of an opportune metric for the space; moreover, once this crucial step is performed, the clusters that have been identified need to be analyzed in order to assess their validity with respect to their subsequent use in the knowledge representation system; to this aim, it may be useful to identify representative objects for each cluster.

Amongst the different available clustering techniques suitable for separating convex regions in the given space, we have chosen to use Support Vector Machines (SVMs) [8] that, thanks to their general-purpose approach and their customizability with respect to the concepts of geometric space and metric, are promising for our specific purposes. They rely on the structural risk minimization principle in order to separate m classes through the best hyperplane. Although one of their drawbacks is their slowness during the training phase, this is not a serious limitation in our case, since we may as well rely on a single training phase to be performed at the beginning. However, if we wanted to add further knowledge at a later stage (as could happen, for instance, if we wanted to provide some feedback from the symbolic layer), we could resort to using a modified *incremental* version of SVMs [9] that allows for dynamic modifications of the model.

2.3 Symbolic Layer

At the symbolic layer our system must provide a concise description of the perceived environment in terms of a high-level logical language, capable of symbolic knowledge-based reasoning. Knowledge representation is a crucial aspect of AI and its main goal regards the expression of knowledge in computer-tractable form, such that it can be used to help intelligent agents in their activities. This layer aims at describing the relationships needed to infer complex events starting from “phenomena” extracted from the conceptual layer and, to this end, a description of the application domain ontology is needed. As in [5], the symbolic knowledge base description can be carried out by means of a hybrid representation formalism, in the sense of Nebel [10]. Such a hybrid formalism is constituted by two different components: a terminological module and an assertional module; the former contains the descriptions of the concepts relevant for the represented domain (e.g., types of objects and of relationships to be perceived), whereas the latter stores the assertions describing the particular perceived environment.

This distinction into two components is mirrored here by the use of two different formalisms. In particular, we adopt a KL-ONE-like notation [11] for representing the structured knowledge in the form of concept definitions, subsumption relationships, and roles. Furthermore, in order to enable the system to reason about instances of the defined terms, we represent them as one- and two-place predicate symbols, relying on a standard first-order logic.

3 Wildfire Detection: A Case Study

This Section describes a sample application of the proposed framework, aiming at monitoring a forest through a Wireless Sensor Network, in order to timely detect the

presence of fire. One of the most important aspects in wildfire detection and prevention is fire behavior analysis; according to [12], this may be defined as “the manner in which fuel ignites, flame develops, and fire spreads and exhibits other related phenomena as determined by the interaction of fuel, weather, and topography.”

In the scenario considered here, the area to be monitored is covered with low-cost, resource-constrained wireless sensor nodes, deployed at known locations and at varying heights, and sufficiently close to each other so as to form a connected network. Our system employs commonly available nodes equipped with sensors for monitoring environmental conditions such as wind speed and direction, air temperature, relative humidity, and barometric pressure, as well as more specific sensors such as detectors for carbon monoxide, carbon dioxide, water vapor, smoke, and flames. For the purpose of the required monitoring actions, large amounts of data will need to be collected and processed, in order to extract meaningful information about the state of the environment; the information deduced by means of this higher-level analysis will then be used to promptly alert in case of likely dangerous conditions. We adopt a model-based technique, in order to exploit correlation in the data through a distributed node grouping protocol [7]. This is motivated by the fact that in a typical sensor network monitoring task, the readings of sensor nodes show high temporal and spatial correlation; for instance, in the scenario considered here, environmental conditions will likely not be subject to significant changes over time, unless some disruptive event occurs. We are not interested in constant monitoring of environmental conditions, but rather in capturing unusual phenomena that might be interpreted as fire alarms or, in other words, outliers with respect to the given prediction model. Nodes may thus organize themselves into clusters ruled by a collaboration relationship. A cluster head acts as a representative for the whole group, by building a model for predicting its collaborators’ behavior; it will then transmit only those readings that differ from the predicted ones by more than a certain, pre-specified error threshold; moreover, when no transmissions are received from the collaborating nodes, the monitoring entity uses the sensors’ prediction models to infer their readings. Periodically, the cluster head will receive updated prediction models from its collaborators. Moreover, in order to provide resilience to natural variations in data, for instance as a consequence of different weather conditions depending on seasonal changes, we have modified the original collaboration protocol so that models are periodically updated to incorporate useful statistics, for instance about seasonal and daily trends of temperature, barometric pressure, and so on.

Sensor readings are used to characterize the conceptual spaces used in our system. When applying our architecture to the present scenario, we identify two conceptual spaces related to topology and environmental measurements, respectively. The dimensions of the former space correspond to the three spatial components of the known location of the sensor nodes, whereas for defining the latter space we interpret the different types of sensor readings as quality measures, thus individuating the following quality dimensions: variation of temperature with respect to the expected value, variation of toxic gas concentrations with respect to the expected values, water vapor concentration, barometric pressure, UV spectrum, and IR spectrum. The pre-processed sensor readings are mapped along these quality dimensions, in order to obtain an ad-hoc geometric representation of knowledge about the monitored

environment. Points in the conceptual spaces provide a compact representation of sensed quantities; a perception cluster may thus be defined as a finite set of points in one of the conceptual spaces, and such clusters may be identified through automated techniques, that adopt opportune metric functions as a measure of similarity. In particular, by applying SVMs to the above mentioned conceptual spaces, we are able to isolate specific regions, that represent basic concepts in our formulation. Such concepts may individuate, for instance, a “smoke cloud,” an “incomplete combustion area,” an “overheated area,” a “flaming area,” and so on. Furthermore, a projection technique, combined with the use of a SVM in the resulting projection space, is used to extract relationships among those concepts. For instance, assuming that a smoldering fire is in fact present in the monitored area, we can compute the projection of the conceptual points in the “overheated area,” “smoke cloud,” and “incomplete combustion area” regions onto the spatial dimensions, and cluster the resulting space. Provided that the original points regarded phenomena occurring in the same topological area, the process would identify a region representing a set of relationships involving the above mentioned concepts. The same process would identify other analogous relationships that would then be mapped onto symbolic structures as will be detailed in the following.

A knowledge representation system is adopted at the upper layer, with the final goal to infer higher-level events and to correctly classify them. As already discussed in the previous Section, we adopt a logically oriented formalism that is constituted by two different components: a terminological component, building domain descriptions, and an assertional component, storing facts concerning a specific context. In particular, the descriptive task aims at forming an extensible repertoire of terms: a domain-specific vocabulary defining concepts on the basis of their taxonomical structure and their potential relationships (i.e., properties, parts, etc.). In our case study, we consider a simplified taxonomy of wildfires, and specify their necessary conditions in terms of the basic concepts that can be identified as regions on the conceptual spaces described above. The different kinds of wildfire can be informally defined as follows:

- **smoldering**: a single-spot fire burning without flame, generating moderate heat, and emitting toxic gases (e.g., carbon monoxide) at a higher yield than flaming fires;
- **creeping**: a flaming fire with low flame;
- **running**: a flaming fire with high flame and with a well-defined front;
- **spotting**: a flaming fire producing firebrands that fall beyond the main fire perimeter and result in spot fires;
- **torching**: a flaming fire with high flame involving the foliage of a single tree (or a small clump of trees);
- **crowning**: a flaming fire ascending into the crowns of trees and spreading from crown to crown.

Figure 2 depicts the subsumption relationships between these concepts (we adopt a network notation similar to that of Brachman and Schmolze [11]). The complete formalization captures the relationships that characterize the wildfire taxonomy concepts in terms of the basic concepts singled out from measured data mapped into conceptual spaces. For example, Figure 3 represents a fragment of our terminological

knowledge base, showing that a WILDFIRE is a THING that emits any number of SMOKE CLOUDs, contains at least one COMBUSTION AREA, is delimited by at least one OVERHEATED AREA, and covers any number of FLAMING AREAs. It also shows that SMOLDERING COMBUSTION represents a single-spot WILDFIRE, without any FLAMING AREA, and containing one INCOMPLETE COMBUSTION AREA.

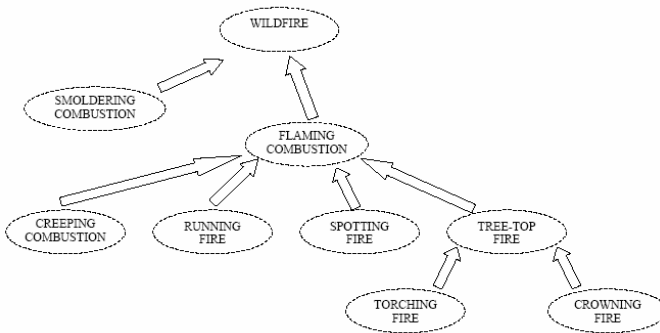


Fig. 2. Fragment of the terminological knowledge base describing the wildfire concepts taxonomy

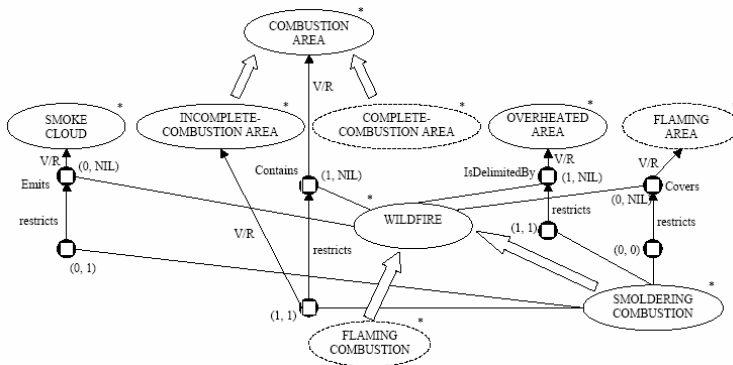


Fig. 3. Fragment of the terminological knowledge base describing the SMOLDERING COMBUSTION concept

The assertional component, on the other hand, maintains the linguistic information concerning specific perceived contexts. In our sample case study, we represent the concepts of the terminological components as one-argument predicates, and the roles (e.g., *Emits*, *Contains*, and so on) as two-arguments relationships. For example, in order to assert the existence of an instance SC#1 of the SMOLDERING COMBUSTION concept, the formula: *SmolderingCombustion*(SC#1) is added to the assertional knowledge base. To express that the filler of the role *Contains* for SC#1 is an instance ICA#1 of the INCOMPLETE COMBUSTION AREA concept, the formula *Contains*(SC#1, ICA#1) is asserted.

4 Conclusion

We have presented a new cognitive architecture for extracting meaningful, high-level information from the environment, starting from the raw data collected by a Wireless Sensor Network. Our proposal exploits the idea of representing knowledge through geometric structures which integrate the connectionist, data-driven approach with the symbolic one. The architecture has been illustrated with reference to an experimental setup for wildfire detection.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. *IEEE Communication Magazine* 40(8) (Aug. 2002), pp. 102-114
2. He, T., Krishnamurthy, S., Luo, L., Yan, T., Gu, L., Stoleru, R., Zhou, G., Cao, Q., Vicaire, P., Stankovic, J.A., Abdelzaher, T.F., Hui, J., Krogh, B.: Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sen. Netw.* 2(1) (2006), pp. 1-38
3. Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., Hong, W.: A macroscope in the redwoods. In: *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, New York, NY, USA, ACM Press (2005), pp. 51-63
4. Gärdenfors, P.: *Conceptual Spaces. The Geometry of Thought*. MIT Press, Cambridge, MA, USA (2000)
5. Chella, A., Frixione, M., Gaglio, S.: A cognitive architecture for artificial vision. *Artificial Intelligence* 89(1-2) (1997), pp. 73-111
6. Chella, A., Frixione, M., Gaglio, S.: Understanding dynamic scenes. *Artificial Intelligence* 123(1-2) (2000), pp. 89-132
7. Goel, S., Imielinski, T., Passarella, A.: Using buddies to live longer in a boring world. In: *Proc. IEEE PerCom Workshop. Volume 422*, Pisa, Italy (2006), pp. 342-346
8. Cristianini, N., Shawe-Taylor, J.: *An Introduction to Support Vector Machines*. Cambridge University Press (2000)
9. Cauwenberghs, C., Poggio, T.: Incremental and decremental Support Vector Machine learning. In: *Proc. of the 14th Conf. on Advances in Neural Information Processing Systems (NIPS)* (2000), pp. 409-415
10. Nebel, B.: Reasoning and revision in hybrid representation systems. In: *Lecture Notes in Artificial Intelligence. Volume 422*, Berlin, Springer-Verlag (1990)
11. Brachman, R., Schmoltze, J.: An overview of the KL-ONE knowledge representation system. *Cognitive Science* 9(2) (1985), pp. 171-216
12. Merrill, D.F., Alexander, M.E., eds.: *Glossary of Forest Fire Management Terms*. Canadian Committee on Forest Fire Management. National Research Council Canada, Ottawa, Ontario, Canada (1987)

A Model of Affective Entities for Effective Learning Environments

Jose A. Mocholí Javier Jaen, and Alejandro Catalá

Department of Information Systems and Computation, Polytechnic University of Valencia
Camino de Vera s/n, 46022 Valencia, Spain
{jmocholi, fjaen, acatala}@dsic.upv.es

Abstract. Learning is a never ending activity for humans; it takes place everywhere and even when we do not realize. However, current learning environments make students deal with lectures, mostly associated with low control of the situation and implicit motivation. In contrast, previous researches have shown that sports, games or hobbies are activities that make people reach optimal experiences where self-motivation, control of the situation, high level of concentration and enjoyment are present. Some current efforts to design next generation of learning environments make use of ubiquitous systems to encourage students to perform learning activities everywhere and at anytime. However, those approaches lack the affective factor related to optimal experiences. To address this problem we present eCoology, an edutainment application that creates a ubiquitous learning environment with emotional features, and discuss some experimental results.

Keywords: Affective Computing, Ant Colony Optimization, Evolutionary Computing, Ubiquitous Computing.

1 Introduction

Current educational institutions force students to deal with classmates and teachers following adult interactions and social rules of behavior which are associated with low control of the situation and implicit motivation [1]. However, the most positive experiences are reported in activities such as sports, gaming, arts and hobbies which merge fun with concentration and goal setting [2]. Particularly relevant in these activities has been the identification of *optimal experience* or *flow* [3] which is characterized by the perception of high environmental challenges, adequate personal skills, high levels of concentration, enjoyment and engagement, loss of self-consciousness, control of the situation, focused attention, positive feedback, clear ideas about the goals of the activity and intrinsic motivation [4]. Among these factors, we are especially interested in those related with emotions because, as concluded by many researches (e.g., [5]), emotions play a significant role in producing rational behavior and rational decision making, as well as it contributes to irrational behavior. Moreover, other researches (e.g., [6], [7]) have shown that even small changes in users' emotional state can significantly impact creativity and problem solving, and that emotion can be seen as a motivating and guiding force in perception and attention

[8]. Thus, one may conclude that emotions must be a key factor when designing computing systems for supporting education to better serve user's needs by recognizing that humans are powerfully influenced by emotions [9]. In fact, this is the main concern of a new research area known as *affective computing*: machines that might actually *have* feelings so they can adapt interactions with users to their emotional state [9]. However, although we are still far from that goal, we know that emotions also play an essential role in communication, even when they merely indicate that our message has been understood [10], and that the appropriate use of even limited affective abilities (e.g., visual expression of an emotional state) may lead to an improvement in the quality of users' experiences [9]. Our claim is that these two findings are critical when designing effective learning environments and, to verify their real impact on learning, we present in this work a prototype learning environment known as *eCoology* [11] that incorporates emotional features.

To this end we will firstly introduce the field of affective computing and propose an evolutionary model of affective entities for ubiquitous learning environments in which educational software agents may exhibit different emotional behaviors to enhance the learning process. Secondly, we will present eCoology, a ubiquitous AR learning framework that incorporates the affective models presented in this work and, finally, we will discuss some experimental results that prove the significant effects that emotions have on ubiquitous learning together with mobility and transparency.

2 eCoology: An Evolutionary Affective Learning Environment

eCoology¹ is an AR game application consisting of an ecosystem with plants, animals, energy sources and people. eCoology is an acronym for "An Electronic Cool Way of Learning about Ecology", and the main goal is to teach children (aged 8 to 12) about healthy everyday values such as having a balance diet, taking care of the environment, using renewable sources of energy, and behaving respectfully with others by means of an optimal flow experience based on ubiquitous games. Children are proposed gaming activities such as feeding animals with different types of food or evaluating over time the health status of the animals under their responsibility. Animals that are mainly given unhealthy food eventually look sick, behave as if they were sad and children have to find out the ultimate reasons of their situation. The rationale behind this activity is to make children think about their own feeding behavior, and its consequences by analyzing a similar situation with virtual animals in an AR ecosystem.

Having entities in eCoology with emotional behaviors is a key feature for children to perceive the augmented ecosystem as an attractive interactive environment for learning. Particularly interesting when dealing with immersive AR experiences is the representation of emotional behavior. This interest arises from the need of enhancing user engagement: an AR environment with virtual animals moving around and reacting according to some emotions can improve users' immersion and engagement.

To achieve this emotional behavior, we have extended the emotional model for the entities of eCoology presented in [12] with a decision making subsystem (Fig. 1):

¹ A short video of eCoology can be found at <http://momo02.dsic.upv.es/descargas/BBC.wmv>

- Perception: This is a basic component of any behavioral model for entities (be they emotional or not) that have to interact with their surrounding environment. It is responsible for extracting data from the environment (stimuli). This information may be obtained from an entity's field of vision, i.e., obstacles and objects, presence of other nearby entities (and their emotional state), and also from what an entity can hear, such as the call from its owner.
- Emotional Profile: This subsystem establishes the *character* of an entity in terms of what an entity will feel for a certain type of stimulus. Besides, it also contains an *emotional memory* to remember past interactions with other entities so that an entity may express emotional feelings with respect to them. This subsystem generates a probability distribution for the current emotional state based on an entity's character, its emotional memory, and the perceived stimuli from the environment.

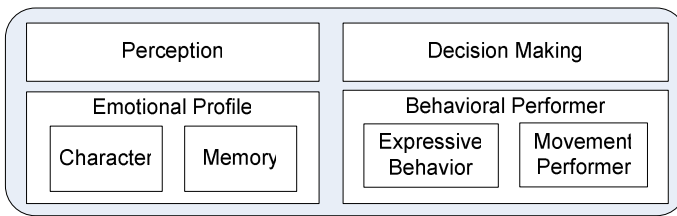


Fig. 1. Model of an emotional entity of eCoology

- Decision Making: With the perceived stimuli and the probability distribution for the current emotional state, this subsystem chooses the decision to make in order to keep the entity in a set of desirable emotional and cognitive states. A thorough explanation of this subsystem can be found in the following sections.
- Behavioral performer: This subsystem is responsible for expressing reactions and behaviors that relate to emotions and to the decisions an entity has made. These reactions include body expressions, such as those depicted in Fig. 2, and movement behaviors.



Fig. 2. Depiction of a dog that “feels” aggressive and a happy chicken

2.1 Decision Making and Emotions

In living beings, previous emotional experience creates covert biases that assist the reasoning process and facilitate the efficient processing of knowledge and logic

necessary for conscious decisions [13]. This is an important reason to take into account emotions when designing decision making algorithms such as the one introduced in [14], where authors proposed a hierarchical network of decisions for entities that must behave emotionally where each node/decision (they use the term *affective-cognitive decision, ACD*) of the network has an internal process that can select the more (probabilistically) appropriate lower-level decision given the current cognitive and emotional states. This process maintains several probabilistic models to simulate lower-level ACDs together with their emotional and cognitive values. In other words, to measure how attractive in emotional and cognitive terms are certain decisions. Once a lower-level ACD with a positive internal value is located, the entity actually performs it and gets the external reward. Then, all the probabilistic models are updated to reinforce the learning of those ACDs that are better in each situation.

Although this algorithm uses reinforcement learning to make entities learn more appropriate decisions for each pair of cognitive and emotional states and it goes through a simulation to select those decisions, we think that it can be improved since the simulation always stops at the first ACD that produces a positive value (not even the maximum, for example), and it does not take into account past sequential occurrences of cognitive and emotional states, i.e., this approach does not take advantage of the entities knowledge to simulate sequences of decisions that may improve the overall obtained reward.

Our proposal is based on [14] but with a completely different representation: it can be seen as a completely connected graph where nodes are the possible cognitive states and edges are decisions that connect two nodes. We also make use of a probability distribution for the current emotional state that is updated after an entity makes a decision. However, in our approach we want entities to be able to *foresee* the most probable sequence of situations and the associated most probable best sequence of decisions, i.e., the sequence of decisions that is more likely to maintain the entity in wanted states while obtaining the highest rewards.

However, since the number of possible cognitive states and the number of decisions to be made can be large, the search becomes a problem without solution in polynomial time. Therefore, to avoid this problem, we have made use of a well known metaheuristic: *ant colony optimization (ACO)*. This metaheuristic is based on a natural behaviour of actual ants in which they use a chemical substance, called pheromone, to mark paths to food. Several experiments [15] have shown that this communication mechanism is very effective in finding the shortest paths and, as a result, it has inspired several stochastic models [15] that describe the dynamics of these colonies. This led Dorigo and his colleagues [16], [17], [18] to introduce ACO as a metaheuristic that targets combinatorial optimization problems, such as routing, assignment, scheduling, subset, machine learning and routing problems [18].

Using Ant Colonies for Decision Making. Living beings are making decisions at every time: follow a path, take an object, communicate with other beings, etc., and usually make a certain decision because we *foresee* it will benefit us in some way, i.e., it will make us reach or keep a certain emotional or cognitive state. In our approach we use ACO to foresee which decision an entity should make in order to maximize the probabilities of reaching/keeping itself in a set of desired cognitive and emotional states. We have based our approach on the ideas presented in [14] and adapted them

to design an ACO algorithm to forecast which decision maximizes the rewards an entity will obtain in the near future.

At any given moment of the world simulation in eCoology, the environment may present some stimuli an entity has to react to and that are gathered by the perception component of an entity's emotional model (see Fig. 1). Those stimuli are processed by the emotional profile to produce a probability distribution (\hat{e}) for the current emotional state (e). Afterwards, the decision making component feeds our ACO algorithm with the current cognitive state (c), the current emotional probability distribution (\hat{e}), the model of the cognitive reward $R_{CR}(c)$ (cognitive states with positive values are cognitively wanted), the model of the emotional reward $R_{ER}(\hat{e})$ (emotional states with positive values are emotionally wanted), the model of the external reward $R_{ext}(c,d)$ (decisions more suitable for a given cognitive state have greater values), and the decision-making model $Q_{DM}(\hat{e},c,d)$ (the greater values the better a decision d is for the given cognitive and emotional states). Ant colonies use these models to forecast the most profitable decision making based on the next f most probable states transitions (f is the number of future states transitions our algorithm will forecast). The result of this forecasting is an ordered set S of tuples (c, \hat{e}, d) where d represents the decisions that the colony forecasts will maximize the rewards, and where $s_0 = (c_0, \hat{e}_0, d_0)$ is the tuple with the current cognitive and emotional states and the next immediate decision an entity should make.

For each iteration, every ant colony compares solutions obtained by ants in order to select one solution that maximizes the formula (1). After all iterations are done, the solution that maximizes (1) is returned. Then the entity actually makes the decision of s_0 and the algorithm updates the models with the external reward received and the actual cognitive state that is reached.

$$Max \sum_{c \in S} R_{CR}(c) + \sum_{k=1, \hat{e} \in S}^{|S|} R_{ER_k}(\hat{e}) + \sum_{c,d \in S} R_{ext}(c,d) \quad (1)$$

In this way, entities select the decisions (and the emotional behaviors) that maximize the expected reward (learning of new knowledge by users).

3 Experiments and Results

In order to verify our ideas and the correctness of their implementation, we conducted a user study on a basic prototype. Our main goal was to examine users' reactions and feedback on the behaviour of eCoology emotional entities, firstly, to determine if users noticed the emotional behaviour of entities; secondly, to verify if this behaviour had some influence on users learning processes and, finally, to know which details of the implemented emotional features users liked most.

14 volunteers were selected, 2 females and 12 males, all of them students from the Polytechnic University of Valencia and with an average age of 24 years. 2 of them had some kind of experience with virtual reality, and only 1 with AR.

The prototype of eCoology used for this study was running on a TabletPC with a Pentium M 2.13GHz processor, 512MB of RAM and a nVidia GeForce Go 6600 TE with 128MB of RAM; a Logitech Quickcam Pro 5000 was used as a video capture

device. We set up the system to augment a 4x5 meters room with 3 animals: a dog, a chicken and a pig. We asked the volunteers to perform the following three tasks in sequence:

- a. Throw cakes to the augmented environment until you notice a change on any of the animals that ate the cakes you have thrown.
- b. Throw vegetables to the augmented environment until you notice a change on any of the animals that ate the cakes you have thrown.
- c. Throw any of the available food to the augmented environment and press the “D” key whenever you want.

Pressing the “D” key when eCoology is being executed makes animals start dancing in the augmented space; this last task allowed us to observe users reactions, to measure how much an eye-catching and cheerful behavior such as dancing may affect user engagement and enjoyment, and if users curiosity leads to a search for more behaviors and therefore to a self-teaching of how eCoology works.

After performing these tasks volunteers were asked to answer a short questionnaire with three questions; questions 1 and 2 were asked after tasks 'a' and 'b', respectively, and the last question after task 'c'.

1. Do you think animals should eat cakes? If not, how did you arrive to that conclusion?
2. Do you think animals should eat vegetables? If yes, how did you arrive to that conclusion?
3. What did you like most or remember most in eCoology? List 3 things beginning from what you liked the most.

With tasks a and b and its related questions we wanted to determine if users were able to notice emotions in the reactions of the animals, and whether these emotions had helped them to learn which food was more healthy for the animals. In the third question the goal is to know if the animations we have created to make animals visually express emotions are good enough. Finally, the last question allowed us to evaluate which characteristics of eCoology are the most appealing.

3.1 Results

In Fig. 3 we can see our interpretation of users answers for the first two questions. We have studied the users answers to determine if they have based their answers on visual clues only (e.g., a color or shape change), on emotions only, or on a combination of both. For example, in the first experiment users were asked to throw just cakes to feed animals until they detected any change on the animals. Eventually, animals got sick after eating several cakes, what was visually represented with a skin color change and an emotional behavior that made animals to seem sad and downhearted. Almost one third of the users answered that it was a color change what helped them to build their answers (e.g., “Its color is not a healthy one”, “It got a strange color”). But almost twice of the answers shown that a combination of emotional behavior and visual clues helped the users in a significant way (e.g., “It has changed its color and looks like weak and sad”, “They changed their color and looked bad”, “Pig changed its color and its face showed anguish”).

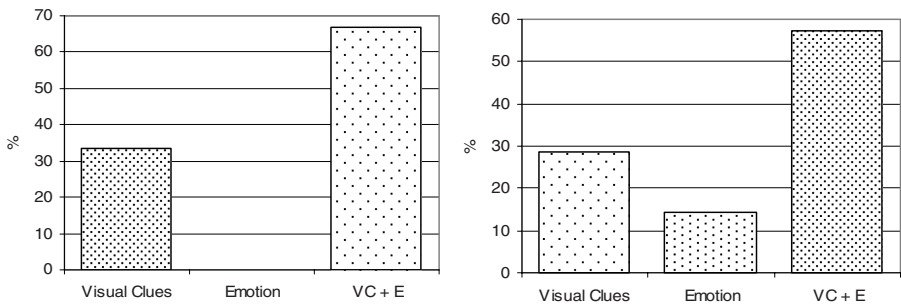


Fig. 3. Results for questions 1 (left) and 2 (right)

With the second experiment we wanted users to learn that feeding the sick animals of the AR environment with vegetables would normalize their nutritional levels and, hence, they would return to a healthy state that was visually represented with a skin color change and an emotional behavior that expresses joy and happiness. In the right figure of the Fig. 3 we can observe that more than a half of the users reported that the combination of emotional behavior and visual clues helped them to achieve the goal of the experiment.

The third experiment was designed to allow us to observe how users behave when they have the chance to keep on throwing food to the environment or just press a specific key without a known effect. Most users thrown more than 5 items of food, one of them more than 15 items, and the main reason they gave when asked was “because I like to see how they react”. Eventually users pressed the “D” key, which in eCoology triggers an eye-catching behavior that expresses joy and happiness: all animals start dancing. In almost all users the dance produced a positive emotional response as they started to laugh and some of them whistled the song. This explains why the dance appeared as one of the things users liked or remembered most in nearly all the answers given for the third question.

4 Conclusions and Future Works

In this paper we have presented a decision making algorithm based on ACO which is part of the emotional model we created for the entities of the AR learning environment eCoology. The algorithm allows entities to foresee most probable situations in the near future in order to select a decision, what allows an entity to reach or keep itself in a set of desired cognitive and emotional states.

We have also presented the results of a user study from which we concluded that having entities that visually express emotional behaviors may significantly ease the learning processes in an AR ubiquitous computing application for education. Moreover, the immediate reactions of animals to users’ interaction (with emotional behavior visually expressed when it is appropriate) generate a positive response on users that motivates them to keep on using the capabilities of the system. This allows us to infer that designing ubiquitous learning systems with an emotional component (e.g., giving emotional behaviors to the virtual entities) permits users to reach higher levels of involvement, as it increases the levels of the perceived transparency, although the system has a low level of “real” transparency.

Our plans for future work include to study the relationship between the accuracy of the forecasting and the amount of experience of the entity in order to permit a dynamic adaptation of the number of future sequential decisions that are simulated (i.e., if an entity has no enough experience to warrant a certain level of accuracy, reduce the number of future simulated decisions). Another enhancement will be the modification of the algorithm to permit emotion and cognition to have different weights on the decision selection.

References

1. Delle Fave, A., Bassi, M., Massimini, F.: Quality of Experience and Daily Social Context of Italian Adolescents. In A.L. Comunian, U.P. Gielen (Eds): *It's all about relationships* (2002)
2. Verma, S., Larsson, R.W.: *Examining Adolescent Leisure Time Across Cultures: Developmental Opportunities and Risks*. New Directions in Child and Adolescent Development Series (2003)
3. Csikszentmihalyi, M., Csikszentmihalyi, I.: *Optimal Experience*. Psychological Studies of Flow in Consciousness. Cambridge University Press (1988)
4. Deci, E.L., Ryan, R. M.: *Intrinsic Motivation and Self-Determination in Human Behavior*. Plenum Press (1985)
5. Damasio, A.R.: *Descartes' Error: Emotion, Reason, and the Human Brain*, Gosset/Putnam Press (1994)
6. Isen, A.M., Daubman, K.A., Nowicki, G.P.: Positive Affect Facilitates Creative Problem Solving. *Journal of Personality and Social Psychology*, 52, 6 (1987) 1122-1131
7. Isen, A.M.: Positive Affect and Decision Making. In M. Lewis and J. Haviland (eds.): *Handbook of Emotions* (2000)
8. Izard, C.E.: Four Systems for Emotion Activation: Cognitive and Noncognitive Processes. *Psychological Review*, 100, 1 (1993) 68-90
9. Picard, R.W.: Affective Computing: Challenges. *International Journal of Human-Computer Studies*, 59, 1-2 (2003) 55-64
10. Picard, R.W.: *Affective Computing*. MIT Press, Cambridge (1997)
11. Acosta, R., Catalá, A., Esteve, J.M., Mochol J.A., Jaén, J.: eCoology: un Sistema para Aprender Jugando. *NOVATICA*, 182 (2006) 63-67
12. Mochol J.A., Esteve, J.M., Jaén, J., Acosta, R., Xech, P.L.: An Emotional Path Finding Mechanism for Augmented Reality Applications. 5th International Conference on Entertainment Computing. Springer LNCS 4161 (2006) 13-24
13. Bechara, A., Damasio, H., Tranel, D., Damasio, A.: Deciding Advantageously Before Knowing the Advantageous Strategy. *Science*, 275 (1997) 1293-1295
14. Ahn, H., Picard, R.W.: Affective-Cognitive Learning and Decision Making: A Motivational Reward Framework for Affective Agents. The 1st International Conference on Affective Computing and Intelligent Interaction, Beijing, China (2005)
15. Deneubourg, J.-L., Aron, S., Goss, S., Pasteels, J.-M.: The Self-organizing Exploratory Pattern of the Argentine Ant. *Journal of Insect Behavior*, 3 (1990) 159-168
16. Dorigo, M., Maniezzo, V., Colomi, A.: Positive Feedback as a Search Strategy. Technical Report No. 91-016, Politecnico di Milano, Italy (1991)
17. Dorigo, M., Maniezzo, V., Colomi, A.: The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans. Systems, Man and Cybernetics, Part B* (1996) 26, 29-4.
18. Dorigo, M., Stützle, T.: *Ant Colony Optimization*. MIT Press (2004)

Image Restoration in Electron Cryotomography – Towards Cellular Ultrastructure at Molecular Level

J.J. Fernández^{1,2}, S. Li¹, and R.A. Crowther¹

¹ MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK

² Dept. Computer Architecture and Electronics, University of Almería, Almería 04120, Spain
jjfdez@ual.es

Abstract. Electron cryotomography (cryoET) has the potential to elucidate the structure of complex biological specimens at molecular resolution but technical and computational improvements are still needed. This work addresses the determination and correction of the contrast transfer function (CTF) of the electron microscope in cryoET. Our approach to CTF detection and defocus determination depends on strip-based periodogram averaging, extended throughout the tilt series to overcome the low contrast conditions in cryoET. A method for CTF correction that deals with the defocus gradient in images of tilted specimens is also proposed. These approaches to CTF determination and correction have been applied here to several examples of cryoET of pleomorphic specimens and of single particles. CTF correction is essential for improving the resolution, particularly in those studies that combine cryoET with single particle averaging techniques.

1 Introduction

Electron cryotomography (cryoET) has a unique potential to elucidate the structure of large biological specimens at molecular resolution [1], [2]. However, technical and computational advances are required to realize this potential [2]. One current computational limitations is determination of the contrast transfer function (CTF) of the electron microscope and restoration of its effects. The CTF models the linear image formation system of the microscope [3]. There has not been pressing need of CTF correction so far, because the resolution achieved was no better than 4.0nm [4], [5]. However, if molecular resolution is to become attainable, CTF correction will be critical [2].

For data acquisition in cryoET, a set of images from a single individual specimen is acquired at different orientations by following the so-called single-axis tilt geometry. Here, the specimen is tilted over a limited range, (typically $[-60^\circ, +60^\circ]$, or $[-70^\circ, +70^\circ]$) at small tilt increments (1° – 2°). An image of the same object area is then recorded at each tilt angle (Fig. 1(a)). The set of images is called a single-axis tilt series. Those images represent projections, or in other words, “radiographs”, of the specimen. To prevent radiation damage, images are taken at reduced electron doses, which makes signal-to-noise ratio (SNR) extremely low [1]. From those projection images, a 3D reconstruction (tomogram) can be obtained by tomographic reconstruction algorithms.

In the acquisition of images of tilted specimens, there is a defocus gradient in the direction of the electron beam. As the CTF is a function of defocus, this defocus gradient produces a CTF gradient and turns CTF correction into a spatially-variant restoration problem. This article presents an approach to CTF determination and restoration that overcomes the extremely low signal-to-noise ratio (SNR) in cryoET data and the CTF gradient in images of tilted specimens. CTF detection depends on strip-based periodogram averaging, extended throughout the tilt series to overcome the low SNR. Our approach to CTF correction decomposes the global spatially-variant restoration problem into multiple local spatially-invariant problems.

2 CTF Determination

The CTF gives rise to oscillations in the power spectrum of the image. The location of these oscillations is essential for accurate determination of defocus, but it is difficult in cryomicroscopy due to the extremely low SNR.

Periodogram averaging is a spectral estimation method that facilitates detection of the CTF oscillations. It subdivides the image into tiles whose power spectrum is computed by means of the periodogram, i.e. the squared magnitude of the Fourier transform. The spectra are then averaged to yield the “averaged periodogram”. Periodogram averaging allows a better estimate of the true power spectrum of the image, as the noise is significantly reduced [6].

Background subtraction then aims to extract the oscillatory component of the power spectrum. Usually it consists of fitting a curve to the positions of the local minima of the smoothed power spectrum [6]. Finally, the defocus is determined by maximization of a correlation coefficient between the background-subtracted smoothed power spectrum and a calculated CTF [7].

Our approach to reliable determination of the CTF parameters in the untilted plane depends on extensive periodogram averaging of areas with similar defocus throughout the tilt series. This significantly increases the number of tiles in the averaging, with a substantial noise reduction, thereby yielding a better estimate of the true power spectrum. We have also developed a new way, embedded in the optimization itself, to remove the background from the smoothed power spectrum.

2.1 Strip-Based Periodogram Averaging

The method consists of the following steps:

1. For each image in the tilt series, a strip is extracted where the CTF is similar to that in the untilted plane. This strip is located around the tilt axis, running parallel to it. Its width depends on a parameter ΔD that represents the maximum difference in defocus for which the CTF can be assumed invariant. The relationship between the strip width w and ΔD is given by $\tan \theta = \Delta D/w$, where θ denotes the tilt angle (Fig.1(b)). Typically, ΔD is the approximate thickness of the sample.
2. For each image, the strip is subjected to a tiling process. The tiles are overlapping with an optimal overlap of half the tile size [6]. For every tile, its power spectrum is computed.

3. The power spectrum at mean defocus is estimated as the average of the spectra of all the tiles in the tilt series. Finally, rotational averaging is also applied to further smooth the power spectrum.

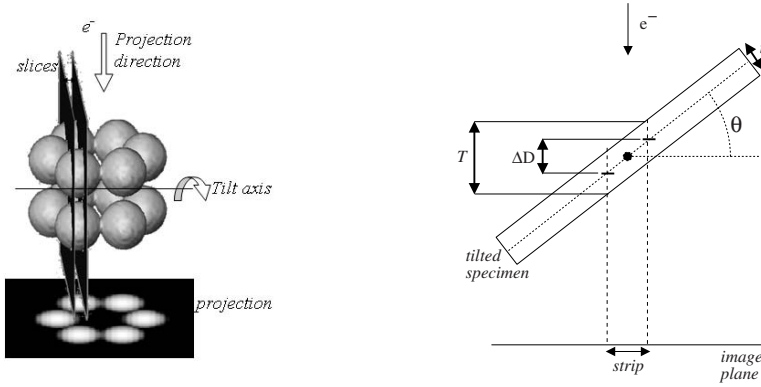


Fig. 1. (a) Single-tilt axis acquisition geometry. (b) Computation of a strip in an image of a tilted specimen. ΔD represents the defocus range considered as a single defocus value, T denotes the defocus range in the strip, t is the thickness of the specimen and θ is the tilt angle. The tilt axis runs perpendicular to the sheet and is marked by the black circle in the specimen slab.

2.2 Determination of Defocus from the Power Spectrum

We have embedded background subtraction into the fitting itself. The background curve is modeled by least-squares cubic spline fitting to the set of samples of the power spectrum located at the positions of the zeros of the theoretical CTF. The defocus is determined by maximizing a correlation coefficient [7] between the background-subtracted smoothed power spectrum and the squared theoretical CTF:

$$CC(D) = \frac{\sum_i (P(\rho_i) - B(\rho_i, D)) C^2(\rho_i, D)}{\sum_i (P(\rho_i) - B(\rho_i, D))^2 C^4(\rho_i, D)} \quad (1)$$

Where D denotes the defocus, $P(\rho_i)$ is the power spectrum estimate, $B(\rho_i, D)$ represents the spline function modeling the background, $C(\rho_i, D)$ is the CTF function [7] and ρ_i is the discrete spatial frequency. The set of knots used for the spline-based background fit is limited to the positions of the zeros of the theoretical CTF. The low resolution, specially up to the position of the first zero of the CTF, is excluded from the fitting. Fig.2 illustrates background subtraction and defocus estimation for a highly underfocused tilt series of 61 images in $[-60^\circ, 60^\circ]$ with 2Kx2K pixels and 10Å/pixel, using $\Delta D=200\text{nm}$ and tile size of 128x128.

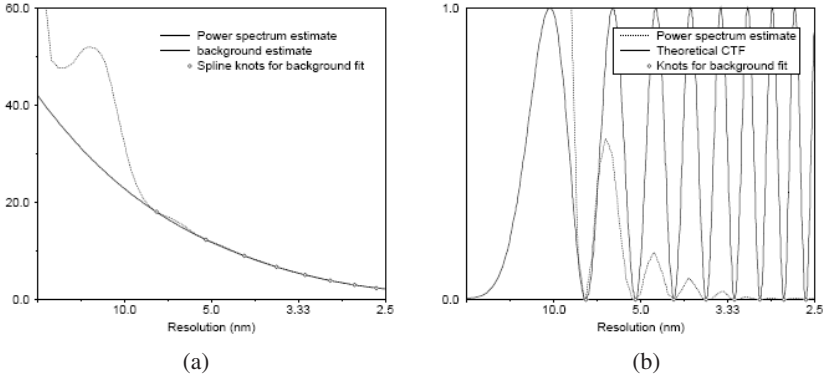


Fig. 2. Background subtraction and defocus estimation. (a) The rotationally-averaged power spectrum estimate is shown in dotted line. For a theoretical defocus of $26.65\mu\text{m}$, the knots for the background fitting are marked with small hollow circles, and the background estimate is shown in solid line. (b) The theoretical CTF is shown in solid line, the background-subtracted power spectrum estimate in dotted line, and the positions of the knots are marked with circles.

3 CTF Restoration

Once an estimate of the defocus D_0 at the untilted plane is available, the defocus at any point of any image in the tilt series can be determined by simple geometric rules. This can be analytically expressed, under the assumption that the tilt axis is located along the Y axis, as:

$$D(x)=D_0+d(x)\tan\theta \tag{2}$$

where x is the coordinate in the X axis, $D(x)$ is the defocus for the pixels along the x -line, $d(x)$ is the distance to the tilt axis and θ is the tilt angle.

To deal with the defocus gradient (Eq.2), we exploit a similar concept of strip to that described for CTF determination, except that now the strip is not restricted to be around the tilt axis (Fig.3(a)). Considering the parameter ΔD as described before, a strip can be extracted along any x -line of the image. A single constant defocus value can be considered for the strip, being computed by Eq.(2), where x is the index of the center line of the strip.

Our approach to CTF correction (Fig.3(b)) essentially decomposes the global spatially-variant restoration problem into multiple local spatially-invariant problems that are confined to the width of a strip. CTF correction of the strip (see Step (3) in Fig.3(b)) is merely a restoration problem with a constant CTF, as usual in electron cryomicroscopy. Standard correction techniques based on phase flipping or Wiener-like filters can then be applied [8]. The overall effect is that each part of each projected image is locally corrected as well as possible for the local value of defocus, prior to the tomogram being computed.

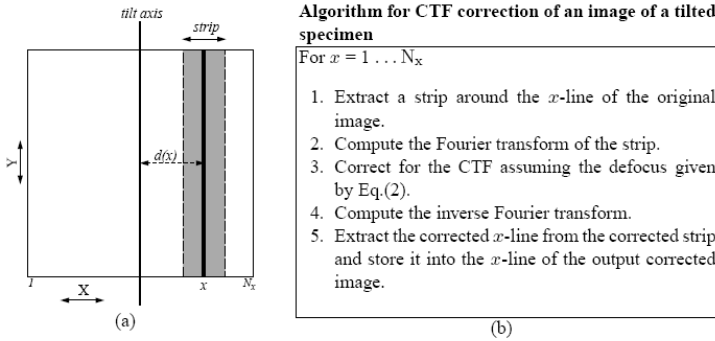


Fig. 3. Strip-based CTF correction. (a) Extraction of a strip with a single effective defocus value from a tilted specimen. The square represents an image acquired from the tilted specimen. The tilt axis runs along the Y axis. x denotes the index of the x -line around which the strip is extracted. $d(x)$ represents the distance from the x -line to the tilt axis. (b) Algorithm for CTF correction of an image.

4 Results

The evaluation of CTF determination and correction has been carried out using experimental cryoET datasets of yeast spindle pole body (SPB) [9], vaccinia virus (VV) [5] and hepatitis B virus (HBV) core [10].

4.1 CTF Determination

We have determined the CTF in a number of tilt series from SPB, VV and HBV core, covering a range of defocus from $27\mu\text{m}$ to $5\mu\text{m}$. All data sets were acquired on an FEI Tecnai F30 TEM (300 kV). The images had $2\text{K}\times 2\text{K}$ pixels and the number of images in a tilt series was around 60-80, in the ranges $[-60^\circ, 60^\circ]$ or $[-70^\circ, 70^\circ]$ at intervals of 1.5° or 2° . The pixel size for the SPB dataset was $10\text{ \AA}/\text{pixel}$, 8.2 \AA for VV and $5\text{ \AA}/\text{pixel}$ for HBV.

The influence of the parameter ΔD on CTF determination was evaluated using different values from 0 to $0.5\mu\text{m}$, in intervals of 50nm . In these tests, we used three different tile sizes (96×96 , 128×128 , 160×160) and an average defocus and a standard deviation were then computed. Fig 4. summarizes the results obtained for five tilt series. The figure shows steady estimates using values of ΔD larger than 100nm - 200nm , with standard deviations lower than 1% of the average defocus. However, CTF determination based on low values of ΔD , and particularly that based solely on the untilted specimen (i.e. $\Delta D=0\mu\text{m}$), produces inaccurate estimates with larger deviations. According to these results, setting ΔD to the approximate thickness of the sample is usually appropriate for reliable CTF estimation for typical specimens.

4.2 CTF Correction

CryoET is currently used for structural studies of large pleomorphic structures or, combined with single particle averaging techniques, of macromolecular assemblies at

medium resolution [1]. SPB was tested as a representative example of the former, and HBV core of the latter.

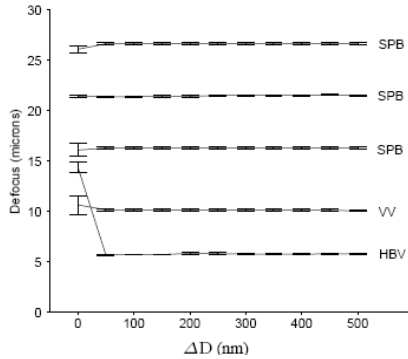


Fig. 4. Defocus estimate for five different tilt series as a function of ΔD . Every point of the curves reflects the average defocus and the standard deviation from three independent estimates that are computed using different tile size. The defocus estimates were around 26.6 μm , 21.6 μm , 15.5 μm , 10.2 μm and 5.7 μm respectively.

Spindle Pole Body. Two tilt series of SPB were selected. First, a highly underfocused tilt series (with $D_o=26.65\mu\text{m}$, hereinafter referred as the 26 μm tilt series), was used as a proof of concept. Second, we tested a closer-to-focus tilt series ($D_o=15.45\mu\text{m}$, referred as 15 μm tilt series in the sequel), which approaches the defocus used in standard cryoET studies of pleomorphic structures. In both cases, the parameter ΔD was set to 200nm. Both tilt series covered the tilt range $[-60^\circ, 60^\circ]$, with 61 and 82 images, respectively. For this test, the original images were binned to a pixel size of 20 Å.

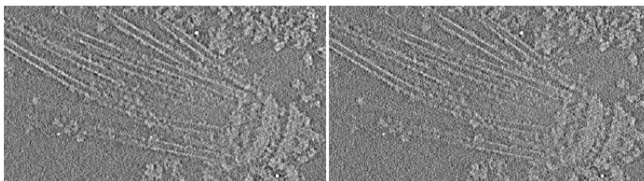


Fig. 5. Effect of CTF correction on tomograms. An area of the tomograms obtained from the original 26 μm tilt series (left), and from the restored tilt series (right) is shown. To increase the contrast in these images, ten slices were summed together.

From the 26 μm tilt series, the tomograms were computed with the standard reconstruction method. The effects of the CTF correction are subtle, as can be seen in Fig5. In general, the whole specimen looks sharper and less blurred and, in particular, the microtubules (MT) have thinner and better delineated walls. In the 15 μm tilt series the effects of the CTF correction could hardly be detected (results not shown here).

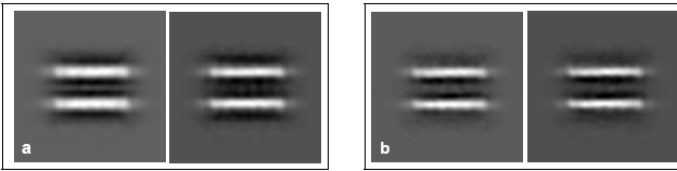


Fig. 6. Effect of CTF correction on the average MT motif. (a) 26 μ m series, (b) 15 μ m series. The average motifs from the original tomogram (left) and from the CTF-restored tomogram (right) are shown. The size of the motif was 48nm long.

To better identify the effects of the CTF correction, we derived an average MT motif from the tomograms. We extracted and aligned a total of 1200 central segments of MTs from the 2D slices of the tomograms. The average MT motifs generated are shown in Fig. 6. In the case of the 26 μ m tilt series, there are clear differences before and after CTF correction. The original average MT had blurred walls and a sharp artifact running along the center of the MT. As a result of CTF correction, the walls of the MT are thinner and sharper, and the artifact is substantially attenuated. In the case of the 15 μ m tilt series, the MT walls on the corrected tomograms are again sharper but the difference is less marked than in the 26 μ m series.

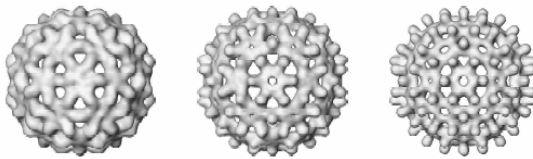


Fig. 7. View of the 3D structure of HBV computed from the tomograms. From left to right: averaged and symmetrized from the original tomogram; from the CTF-restored tomogram; the template used in the alignments.

Hepatitis B Virus Core. The tilt series of HBV core selected for this test had 52 images (tilt range [46°,56°], 2° interval) of size 2K \times 2K with a pixel size of 5 Å. The defocus D_0 was found to be 5.7 μ m using a ΔD of 200nm.

From each of the tilt series (original and CTF-corrected), eighty reconstructed HBV particles were extracted from tomograms, after 3D reconstruction with the standard method at a 20 Å limit. These 3D volumes were then aligned to a previous map of HBV core [10] that was filtered to 20 Å and sampled at 5 Å. The aligned particles were then summed to yield the average HBV core in each tomogram. To increase the SNR, icosahedral symmetry was then applied to the 3D maps. Fig. 7 shows 3D views of these symmetrized average maps. The map from the corrected tomogram shows an evident improvement over the original map. Quantitative resolution assessment showed that the maximum resolution attained in the original map is around 34 Å whereas the CTF-corrected map reaches 23Å[11]. Therefore, CTF correction has made it possible to boost the resolution of the structure.

5 Conclusion

This work has addressed CTF determination and correction, one of the current computational problems in cryoET that limits the resolution attainable. We have proposed an approach to CTF determination that overcomes the low SNR in cryoET by strip-based periodogram averaging extended throughout the tilt series and a spline-based strategy for background subtraction. The method of CTF correction deals with the defocus gradient in images of tilted specimens by decomposing the global restoration problem into multiple local spatially-invariant problems. The approach has been applied to examples of cryoET of pleomorphic specimens and of single particles. CTF correction has shown little effect on pleomorphic specimens due to the inherent resolution limits. However, CTF correction is essential for improving the resolution in those studies that combine cryoET with single particle averaging techniques, if molecular resolution is to be achieved.

Acknowledgements. Thanks to A. Roseman, S. Wynne, N. Grigorieff, J.L. Carrascosa, L. Amos and T. Horsnell for their help during this work. Work partially supported by the MRC and grants MEC-TIN2005-00447, EU-FP6-LSHG-CT-2004-502828, JA-P06-TIC1426.

References

1. Lucic, V., Foerster, F., Baumeister, W.: Structural Studies by Electron Tomography: from Cells to Molecules. *Annu. Rev. Biochem.* 74 (2005) 833–865
2. Jensen, G.J., Briegel, A.: How Electron Cryotomography is Opening a New Window Onto Prokaryotic Ultrastructure. *Cur. Opi. Struct. Biol.* 17 (2007) 260–267
3. Erickson, H.P., Klug, A.: Measurement and Compensation of Defocusing and Aberrations by Fourier Processing of Electron Micrographs. *Philos. Trans. R. Soc. B* 261 (1971) 105–118
4. Medalia, O., Weber, I., Frangakis, A.S., Nicastro, D., Gerisch, G., Baumeister, W.: Macromolecular Architecture in Eukaryotic Cells Visualized by Cryoelectron Tomography. *Science* 298 (2002) 1209–1213
5. Cyrklaff, M., Risco, C., Fernandez, J.J., Jimenez, M.V., Esteban, M., Baumeister, W., Carrascosa, J.L.: Cryo-Eelectron Tomography of Vaccinia Virus. *Proc. Natl. Acad. Sci. USA* 102 (2005) 2772–2777
6. Fernandez, J.J., Sanjurjo, J.R., Carazo, J.M.: A Spectral Estimation Approach to Contrast Transfer Function Detection in Electron Microscopy. *Ultramicroscopy* 68 (1997) 267–295
7. Mindell, J.A., Grigorieff, N.: Accurate determination of local defocus and specimen tilt in electron microscopy. *J. Struct. Biol.* 142 (2003) 334–347
8. Grigorieff, N.: Three-Dimensional Structure of Bovine NADH:Ubiquinone Oxidoreductase (complex I) at 22 Å in ice. *J. Mol. Biol.* 277 (1998) 1033–1046
9. Bullitt, E., Rout, M.P., Kilmartin, J.V., Akey, C.W.: The Yeast Spindle Pole Body Is Assembled around a Central Crystal of Spc42p. *Cell* 89 (1997) 1077–1086
10. Bottcher, B., Wynne, S.A., Crowther, R.A.: Determination of the Fold of the Core Protein of Hepatitis B Virus by Electron Cryomicroscopy. *Nature* 386 (1997) 88–91
11. Fernandez, J.J., Li, S., Crowther, R.A.: CTF Determination and Correction in Electron Cryotomography. *Ultramicroscopy* 106 (2006) 587–596

SeqTrim – A Validation and Trimming Tool for All Purpose Sequence Reads

Juan Falgueras¹, Antonio J. Lara², Francisco R. Cantó², Guillermo Pérez-Trabado³, and M. Gonzalo Claros⁴

¹ Lenguajes y Ciencias de la Computación, ETSI Informática,
Campus Universitario de Teatinos, s/n. E-29071, Málaga, Spain

² Biología Molecular y Bioquímica, Universidad de Málaga,
Campus Universitario de Teatinos, s/n. E-29071, Málaga, Spain

³ Arquitectura de Computadores, ETSI Informática,
Campus de Teatinos, E-29071 Málaga, Spain

⁴ Departamento de Biología Molecular y Bioquímica. Facultad de Ciencias Universidad de
Málaga. 29071 Málaga, Spain
Tel.: +34 95 213 72 84; Fax.: +34 95 213 20 41
claros@uma.es

Abstract. Bioinformatics tools are required to produce reliable, high quality data devoid of unwanted sequences in the preprocessing stage of current sequencing and EST projects. In this paper we describe SeqTrim, an algorithm designed to extract the insert sequence from any sequence read devoid of any foreign, contaminant or unwanted sequence, whatever the experimental process was. SeqTrim is easy to install and able to identify the sequence insert by removing low quality sequences, cloning vector, poly A or T tails, adaptors, and sequences that can be considered contaminants. It is easy to use and can be used as stand-alone application or as web page. The default parameters of the algorithm are best suited for most cases but a configuration file can be provided along with input sequences. SeqTrim admits several input and output formats (with and without quality values), which enables its inclusion in already or newly defined sequence processing workflows. SeqTrim is under continuous refinement due to collaboration between biologists and computer scientists which has succeeded in correct dealing with most sequence cases and opens the possibility to include new capabilities to manage new kinds of bad sequences.

1 Introduction

Sequencing projects and EST data have proven to be an important resource for gene discovery and mapping, and promise to be invaluable for the annotation of the eukaryotic and prokaryotic genomes by providing sequence information to identify novel genes, gene location and even intron-exon boundaries. They are helped by the availability of high throughput automated sequencing, which has enabled the exponential growth rate of these experimentally determined sequences, although not always with the desired quality. However, this exponential growth requires efforts to be made in increasing the quality and reliability of sequences incorporated into databases, since there is a high percent of nucleotides in the databases corresponding to contaminant sequences [1]. Hence, bioinformatic tools are required to produce

reliable, high quality data devoid of unwanted sequences with efficient preprocessing methods.

Preprocessing includes filtering of low-quality sequences, identification of sequence features, removal of contaminant sequences (from vector to any other artifacts) and trimming the undesired segments. Though there are some bioinformatic tools that serve to accomplish individual parts of the preprocessing (e.g. TrimSeq, TrimEST, VectorStrip, VecScreen, ESTPrep, crossmatch), currently used programs that deal with the complete preprocessing are, Lucy [2], SeqClean (<http://www.tigr.org/tdb/tgi/software>) or PreGap4 [3]. However, many of these programs are hard to configure, environment specific, or focused on specific needs (like only pre-process ESTs), and it is not always easy to connect them with further processing tools for annotation or assembling, for example. Moreover, using such programs typically requires a change in implementation and design of either the program or the protocols within the laboratory itself.

This paper presents a new program, SeqTrim, which has been designed to extract the insert sequence from any sequence read devoid of any foreign, contaminant or unwanted sequence, whatever the experimental process was. SeqTrim is easy to install and able to identify the sequence insert by removing low quality sequences, removing cloning vector, removing any special feature (like poly A or poly T tails if present, restriction sites used for cloning, terminal transferase tails if present, sequence adaptors, etc.). Configuration parameters can be customizable transiently or permanently by the user. SeqTrim is provided as a command line tool or as a web tool (<http://castanea.ac.uma.es/genuma/seqtrim>). Although it works as a standalone application, it is intended to work in a web server as a part of an automated workflow that, starting with raw sequences in a database finishes in the design of contigs and the annotation of them.

2 Overview

Although there are many DNA sequence comparison and analysis algorithms in the bioinformatics literature, getting them to work correctly may be very hard, as is the case of Lucy [2], [8], phrap or consed [9]. In other cases, getting them to process high throughput data requires an extra programming effort to consider unlikely special cases that appear when handling large amounts of sequences or the data quality are very low [2]. Due to inherently unpredictable nature of biological data, there may be some distance between the theoretical design of a bioinformatic solution and the successful implementation of the solution in a reliable, working program that can deal with large amounts of data. Since the only way to shorten such distance is to collaborate between computer scientists and biologists, SeqTrim emerged from such collaboration since 1999 in the University of Málaga (Spain). Along these years, the algorithm has been trained with real data obtained from previous, different researches ([10]; R Bautista, FR Cantó, C Avila and MG Claros, unpublished results). In other words, it has been under continuous development, testing and production, which make it well suited for the needs of large-scale sequence preprocessing, providing a time- and cost-effective solution. The use of SeqTrim has significantly reduced the time and complexity involved in numerous gene discovery projects at the University of Málaga.

Installation of SeqTrim does not require special knowledge since there is nothing to compile and the only previous requirements are: (i) presence of blast, (ii) presence of bioperl libraries and, (iii) presence of phred. SeqTrim itself only needs uncompression and moving the 'seqtrim' directory to /usr/local or create a soft link in the /usr/local directory to your preferred location of the seqtrim folder. This transparency is also extended to the configuration: SeqTrim is provided with databases that will be used for vector and contaminant detection, an editable list of restriction enzymes, adaptors and configuration parameters (see above). The configuration file contains the default parameters that can serve to run SeqTrim with results of many sequencing projects. In the seek of customization, a wide range of parameters are available even if we know that most of them will never be changed. Default parameter modification will be permanent only if it is done directly within the seqtrim.conf file; otherwise, default values are used for each run.

Sequence preprocessors are not expected to be used alone but in a pipeline with other programs [11], [12], [13], [14]. Sometimes, constructing a pipeline is not easy mainly due to input/output formats or any other program peculiarities. This has been considered in SeqTrim, since it transparently accepts several kinds of data formats, and outputs different formats too. Input sequences are accepted as follows: (i) one or more fasta file(s), each one containing one or more sequences, (ii) a fasta file and the corresponding quality value file in fasta format, (iii) a phd format file, (iv) a set of chromatograms, and (v) a zip file containing a set of chromatograms. Outputs are generic fasta files, or fasta files compatible with phred/phrap workflows. Hence, SeqTrim can be included in a workflow to remove low quality sequences, cloning vector, and any other special feature (like poly A or poly T tails if present, restriction sites used for cloning, terminal transferase tails if present, sequence adaptors, etc.) to obtain a clean insert sequence.

3 Implementation

The algorithm underlying SeqTrim was programmed in Perl 5.6 using BioPerl libraries, and can be executed as a command line tool (ask for code to jfalgueras@uma.es) or as a web tool (<http://castanea.ac.uma.es/genuma/seqtrim>). SeqTrim makes use of the external programs phred [4], [5] and blast [6] that must be installed in the same machine that SeqTrim. It will work in any unix/linux release, including OSX, but not on Windows. SeqTrim uses a set of files that are gathered in the '/usr/local/seqtrim' directory. In it, the file seqtrim.conf contains all configurable default parameters for analysis. The user can also create a copy into the home directory (~/.seqtrim) to modify it permanently according to the needs. These parameters can also be changed for a single run via command-line options or the web interface (Fig. 1). The '/usr/local/seqtrim' directory also contains an editable file called 'RE_sites.txt' that contains the restriction sites that are recognized by SeqTrim, and a third editable file named 'adaptorSeqs.txt' which contains a list of adaptor sequences that will appear in the popup menu for adaptors in the web interface. The content of every file in the ~/.seqtrim directory can be edited by the user (for local customizations) or the system administrator can modify the /usr/local/seqtrim files for permanent changes.

Enter sequence(s) in FASTA format (comment line is compulsory)

or choose a file to upload

ningún archi...seleccionado

Optional: File with qualities in fasta format

ningún archi...seleccionado

(uploading may take several minutes, please be patient)

Fig. 1. Web interface of SeqTrim that shows the default parameters

SeqTrim was designed to provide a final trimmed sequence without any unsuitable nucleotides throughout a series of main steps (Fig. 2). A sequence read can contain bases of very low quality, which can hinder detection of further trimming features. Hence the first trimming step will take the base-call quality assessment of each base into consideration (and/or the presence of N as nucleotide indetermination) to trim the original sequence in order to obtain the largest sequences with sufficiently high quality. If the input sequences are chromatograms, the external program phred is called automatically and in a transparently way in order to obtain the quality value for each individual base of the raw sequences and then it calculates the largest sequence with the highest quality. If the input sequences are in a text fasta format, SeqTrim contains a function to extract the largest subsequence without too many N (indetermination). Any of them provide a “first trim sequence” which corresponds to the reliable sequence obtained from a read. Parameters for determining the sequence quality can be changed.

The second step relates to cloning vector detection comparing the NCBI’s UniVec and the EMBL’s emvec vector/adaptor libraries against each “first trim sequence”

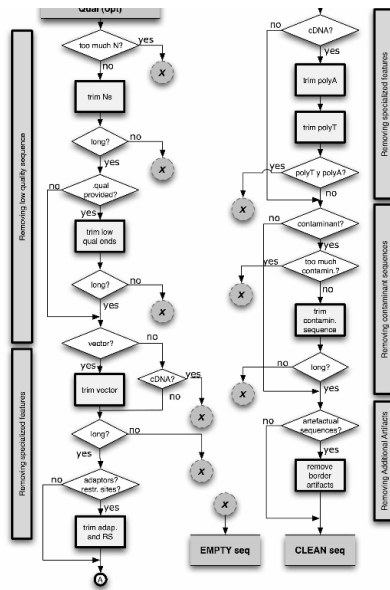


Fig. 2. Detailed data-flow diagram of steps involved in SeqTrim while processing a sequence

using blast so that users do not need to specify the cloning vector used for their experiment. Finding vector with blast seems to be better than locating it as a function of restriction site presence since this targets are subject of read or experimental error, even in high quality sequences, and they are very short to assure a good prediction when they are searched with error tolerance. However, since users are asked for the upstream and downstream restriction sites even if both are the same, SeqTrim tries to locate the upstream and downstream restriction sites without ambiguity as a reference to discard the cloning vector when no vector sequence was identified in the database.

The next step is to locate special features that are common to all sequences as they (i) provide false sequences, (ii) mislead assembling or clustering algorithms that can be further used with these sequences, (iii) mislead researchers that use these contaminated sequences. The special features to locate are adaptors, poly A tails (only for ESTs) and poly T tails (only for ESTs, which indicates that the sequence is in the reverse orientation). Due to the procedures used in cDNA library construction, the length of the poly A or T tail should contain at least 20 successive identical nucleotides unless they are at the end of the sequence. To manage such cases, more than one A at the 3' end of a sequence are always removed.

At this moment, the trimmed sequence can be thought to be correct, but there are a lot of possible experimental contaminations that can make several sequences to be useless. Contamination of the input sequences can come from many sources, such as *Escherichia coli* genomic DNA, the cloning vector, cell plasmids, organelle DNA, viruses, yeast (for tissue culture), human (due to handling), etc. Detection of such contaminants has been achieved launching blast with the trimmed sequences against a database – built from fasta sequence files – that contains sequences than can be considered potential contamination. SeqTrim is provided with the genomes of *E. coli*,

S. cerevisiae, lambda phage and several mitochondria. The vector database is re-screened again. Length, position and identity of the contaminant sequence is provided for user information. Finally, the sequence ends are regarded for presence of any N or artefactual terminal transferase adds.

Rejection criteria for the output sequences include insufficient length (less than 100 bp by default, but this is also modifiable by users), low quality sequence (or excessive amount of Ns) throughout the sequence, absence of insert between identified cloning vector boundaries, detection of two inserts (only in EST) by the presence of two poly A or poly T tails or the concomitant presence of poly A and poly T tails, or considering the complete insert as a contaminant sequence. Warnings are given when cloning vector is not found at the 5' end.

The command line output is a fasta file with the trimmed part from original sequences. Users can determine if the output file will contain or not user readable information for each sequence concerning the trimming events. A coloured output of each sequence can also be seen on the screen, a direct customizing interface being obtained with the SeqTrim web server (Fig. 3), where users can call current and previous executions, save the results according to what is shown on the screen (trimmed sequences, discarded sequences or both), and see the detailed information about each sequence. On the other hand, command line is more adequate for automatic batch processing or workflows. Instead of trimming the original sequence, SeqTrim can mask the unwanted sequences in order to make the output compatible with other programs like the sequence assemblers cap3 [7] or phrap.

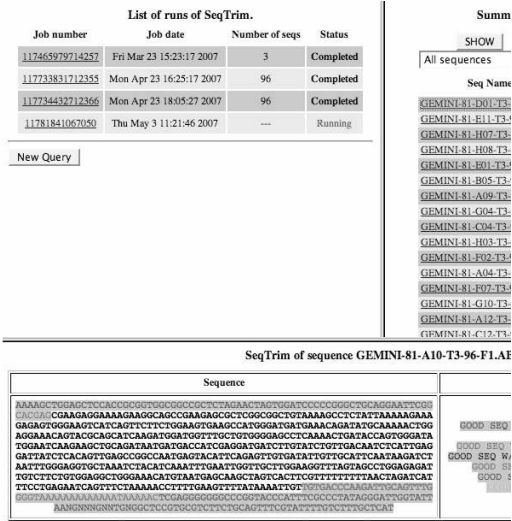


Fig. 3. Example of sequence trimming using different sets of sequences. The upper-left “List of runs” frame displays the status of all executions of SeqTrim made by the user ordered by date and time of run. The upper-right “Summary” frame displays the list of the sequences analyzed in a single session with the original length, and the positions and length of the trimmed part. The bottom frame shows the original sequence coloured according to the different parts that have been removed. Legend on the right explains what colours mean.

4 Discussion

SeqTrim is well suited for the need of large-scale sequence processing. The flexibility regarding input and output formats contrasts with other sequence preprocessors that admit only one single type of sequence file: SeqClean or ESTPrep accept fasta sequences while Lucy or pregap4 accept chromatograms. Concerning the output, the possibility of saving the final sequences as trimmed sequences or as masked sequences enables the possibility to easily include SeqTrim in other previous workflows like pred/crossmatch/repeatmasker/phrap, EGAssembler [12], or EST2UNI http://www.melogen.upv.es/genomica/web_estpipe/). With respect to parameter modification, SeqTrim offers more customization than SeqClean or ESTPrep, nearly pregap4, but do not present the more than forty parameters that can be modified in Lucy [8]. Databases can also be modified directly by the user, simply adding sequences in fasta format, or removing files from appropriate folders in the /usr/local/seqtrim/DB directory. Before each execution, SeqTrim verifies if there are changes in the databases and notifies it to the user, reformatting the database index when needed. Concerning the interface, other sequence preprocessors were created only as command line programs (SeqClean, ESTPrep, TrimSeq, phred/crossmatch), only as web pages (VecScreen, EMBVec Query) or as command line and a GUI interface (Lucy, pregap4). SeqTrim is the first one that provides a command line and web interface, although the creation of EGAssembler has recently provided an additional web interface to SeqClean [8]. Like Lucy, SeqTrim uses colors for the output, but in contrast to the rest of sequence preprocessor programs, SeqTrim colors trimmed regions in each sequence, the clean sequence is shown as black text (Fig. 3). This facilitates result interpretation as well as comparison between cleaned and original sequences since they are on the same string instead of two synchronized scrolling panels as occurs in Lucy. The ‘Info’ cell containing the legend explaining colors also deploys the trimmed range that passed each trimming step.

SeqTrim is under continuous refinement due to collaboration between biologists and computer scientists that have succeed in dealing with most sequence cases and open the possibility to include new capabilities to manage new unexpected sequence behaviors. At present, several changes are planned, like enabling users to select the contaminant databases to use (instead using all of them like now), or to establish a different order of action than described.

Acknowledgements. This work is supported by the Spanish MEC grants AGL-2006-07360/FOR and BIO2006-06216 as well as the Junta de Andalucía grant AGR-663 and foundings to the research groups CVI-114, TIC-160, and TIC-113.

References

1. Coker JS, Davies E (2004) Identifying adaptor contamination when mining DNA sequence data. *Biotechniques* 37, 194, 196, 198
2. Chou HH, Holmes MH (2001) DNA sequence quality trimming and vector removal. *Bioinformatics* 17:1093–1104

3. Bonfield JK, Smith K, Staden R (1995) A new DNA sequence assembly program. *Nucleic Acids Res* 23:4992–4999
4. Ewing B, Green P (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res* 8:186–194
5. Ewing B, Hillier L, Wendl MC, Green P (1998) Base-calling of automated sequencer traces using phred. I. Accuracy assessment. *Genome Res* 8:175–185
6. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. *J Mol Biol* 215:403–410
7. Huang X, Madan A (1999) CAP3: A DNA sequence assembly program. *Genome research* 9:868–877
8. Li S, Chou HH (2004) LUCY2: an interactive DNA sequence quality trimming and vector removal tool. *Bioinformatics* 20:2865–2866
9. Gordon D, Abajian C, Green P (1998) Consed: a graphical tool for sequence finishing. *Genome Res* 8:195–202
10. Cantó F, Le Provost G, Garcá V, BarréA, Frigerio JM, Paiva J, Fevereiro P, Ávila C, Mouret JF, de Daruvar A, Cánovas F, Plomion C (2003) Transcriptome analysis of wood formation in maritime pine. In *Sustainable Forestry, Wood products and Biotechnology*, S Espinel, Y Barredo, E Ritter, eds (Vitoria-Gasteiz: DFA-AFA Press)
11. Liang F, Holt I, Pertea G, Karamycheva S, Salzberg S, Quackenbush J (2000) An optimized protocol for analysis of EST sequences. *Nucleic acids research* 28:3657–3665
12. Masoudi-Nejad A, Tonomura K, Kawashima S, Moriya Y, Suzuki M, Itoh M, Kanehisa M, Endo T, Goto S (2006) EGAssembler: online bioinformatics service for large-scale processing, clustering and assembling ESTs and genomic DNA fragments. *Nucleic Acids Res* 34:W459–462
13. Miller RT, Christoffels AG, Gopalakrishnan C, Burke J, Ptitsyn AA, Broveak TR, Hide WA (1999) A comprehensive approach to clustering of expressed human gene sequence: the sequence tag alignment and consensus knowledge base. *Genome Res* 9:1143–1155
14. Scheetz TE, Trivedi N, Roberts CA, Kucaba T, Berger B, Robinson NL, Birkett CL, Gavin AJ, O’Leary B, Braun TA, Bonaldo MF, Robinson JP, Sheffield VC, Soares MB, Casavant TL (2003) ESTprep: preprocessing cDNA sequence reads. *Bioinformatics* 19:1318–1324

A Web Tool to Discover Full-Length Sequences – Full-Lengther

Antonio J. Lara^{1,2}, Guillermo P  ez-Trabado², David P. Villalobos¹, Sara D  az-Moreno¹, Francisco R. Cant  ¹, and M. Gonzalo Claros^{1,*}

¹ Departamento de Biolog  a Molecular y Bioqu  mica, Universidad de M  laga, Campus Universitario de Teatinos, E-29071 M  laga, Spain

² Arquitectura de Computadores, E.T.S.I. Inform  tica, Campus Universitario de Teatinos, E-29071 M  laga, Spain
claros@uma.es

Abstract. Many Expressed Sequence Tags (EST) sequencing projects produce thousands of sequences that must be cleaned and annotated. This research presents the so-called Full-Lengther, an algorithm that can find out full-length cDNA sequences from EST data. To accomplish this task, Full-Lengther is based on a BLAST report using a protein database such as UniProt. Blast alignments will guide to locate protein coding regions, mainly the start codon. Full-Lengther contains an ORF prediction algorithm for those cases which is not homologous to any sequence. The algorithm is implemented as a web tool to simplify its use and portability. This can be worldwide accessible via <http://castanea.ac.uma.es/genuma/full-lengther/>

1 Introduction

New biological technology produces a large amount of sequences in form of ESTs (Expressed Sequence Tags). These sequences have to be thoroughly annotated to uncover, for example, its function. Currently, the task of annotating EST sequences does not keep pace with the rate at which they are generated [1] since:

1. EST sequence annotation is computationally intensive and often returns no results;
2. EST data suffers from inconsistency problems (error rate, contaminant sequences, low complexity regions, etc.);
3. Gene identification programs perform inconsistently as they are sensitive to errors.

One of the most important and difficult tasks is to discover whether the EST sequence was derived from a full-length cDNA. Basically, a sequence is considered full-length when it contains the gene 5' end. In other words, if an EST is flagged as full-length EST, the plasmid where the EST comes from contains the complete cDNA. The final aim is to flag the “full-length” EST sequences among a long EST list, which will facilitate further study of contigs containing a complete gene for its functionality.

Due to the enormous number of ESTs, such a process is not feasible one by one by researchers, so automatic computer processing of ESTs is required [2]. There are

* Corresponding author: Tel.: +34 95 213 72 84, Fax.: +34 95 213 20 41

several algorithms that have been developed to predict ORFs in ESTs for which unknown orthologues are available, such as NetStart [3], ESTScan [4], ATGpr[5] or OrfPredictor [6], which use neural networks, Markov models and a linear discriminant approaches. These algorithms can predict coding regions but they have to be trained with organism-specific sequences [7]. Recently, web servers for identification of full-length cDNAs using BLAST have been described [8], [9] as they can receive various input formats and outputs a tab-delimited spreadsheet.

To avoid training and in the seek of generic behavior as well as workflow compatibility concerning a sequence analysis and/or annotation process, Full-Lengther was developed. Full-Lengther classifies ESTs as full-length, putative full-length or non full-length, based on matches (similarities) found by executing BLAST against a protein database [2], [9] (i.e. UniProt). BLAST alignments will guide detection of protein coding regions, mainly the start codon. The algorithm also cares about ESTs that do not show any sequence homology, and it offers an alternative method to detect whether or not it is full-length. A web interface for the algorithm has been implemented in order to improve its accessibility to the internet users.

2 Algorithm and Implementation Details

Any standard mRNA has two edges, a start 5'-UTR and an end 3'-UTR. Protein coding region is inside this range, and such region has a well defined start (ATG) and end codons (TAA, TAG or TGA). Clones are produced by a method which guarantees the presence of the 3' end, but not the 5' end. Since natural 5' end can be absent, its presence enables to mark a sequence as complete and to call it "full-length". Our mission is to identify the presence of the 5' end.

The algorithm will mark sequences with one of the following tags:

- **Full-Length:** It will appear when the algorithm has unambiguously detected that the 5' end exists.
- **Non Full-Length:** It will appear when the algorithm has unambiguously determined that the 5' end is not present.
- **Putative Full-Length:** Sometimes it is not clear whether the sequence is full-length. Due to problems like short sequence length, bad quality, lack of reliability... In these cases, the result does not give too much information except that "maybe the sequence is full-length, but it is not sure".

2.1 Selection of Significant Alignments

First of all, it is necessary to determine if an EST is homologous to any known protein sequence. This is quite easy using BLASTX against a protein database like UniProt. BLASTX provides a list of alignments, each one associated to an expect value (quality value), which is a real number, and the smaller it is, the better. The list of alignments has to be then filtered by a cut off value to remove the least significant homologous. The cut off value is set by default to 10^{-6} , although users can modify it.

The SwissProt part of UniProt is the current database used for the BLAST analysis since it is not too large and offers high quality annotations. However, when no hits are

found in SwissProt, Full-Lengther algorithm uses the poorly annotated TrEMBL though it contains a lot of incomplete sequences.

The set of alignments is filtered once again taking into account the best expect value. A new cut off value is generated dynamically to implement this new filter. The expect value is a positive real number whose values closer to zero are the better, the most useful values being very close to zero. The selection criteria is based on the exponent value of the expect value: it only passes alignments with expect values that have a similar order of magnitude than the best one. The cut off is chosen using the expression $10^{e-(e/10)}$, where e is the best expect exponent.

At this point a filtered list of the best alignments is obtained, but this list may be empty if an EST sequence does not produce any significant match in the database. The biological reason for this absence of matches is beyond the scope of this section. To deal with these cases, Full-Lengther provides two ways of analysis, one using the alignments and another taking into account just the primary sequence.

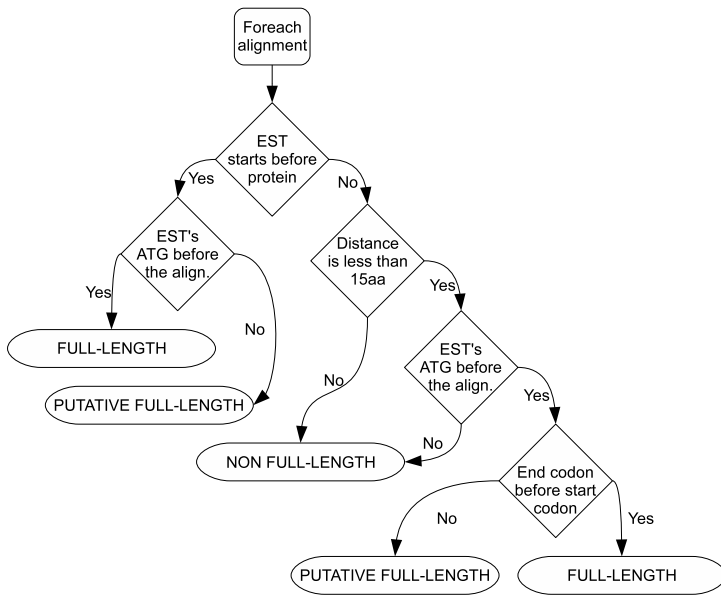


Fig. 1. Processing steps for each alignment

2.2 Analysis of Alignments

Here it is described the analysis Full-Lengther performs on the significant alignments found for an EST. Each EST-subject sequence alignment is processed and classified in one of the three categories (full-length, putative full-length or non full-length) described above. Matches with the most repeated expect value are used to classify the sequence. When it is ambiguous, the average expect value is calculated for every selected category and the best one is then chosen. Even if there is more than one category with the same expect value, the best expect value in each category is used to determine which one is chosen.

To process an alignment (Fig. 1), first, we have to distinguish between three cases:

1. The EST sequence starts before the subject protein. In this case, the alignment will be full-length if there is a start codon in the sequence before the alignment (Fig.2a), or will be putative full-length if there is no in-frame ATG codon or the first is within the aligned sequence.
2. The EST sequence starts after the protein which has the alignment (Fig. 2b), but within a short distance from the beginning. This distance is set to 15 amino acids by default, although users can modify it. If there is an in-frame ATG codon, the EST will then be tagged as putative full-length, but if there is also an in-frame end codon, EST will be tagged as full-length.
3. Any other case will result in a non full-length tag.

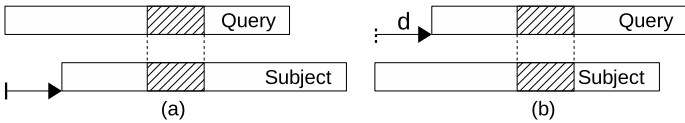


Fig. 2. The query (EST sequence) starts before (2a) or after (2b) the subject (aligned protein), with distance d . Gray regions correspond to the aligned sequence.

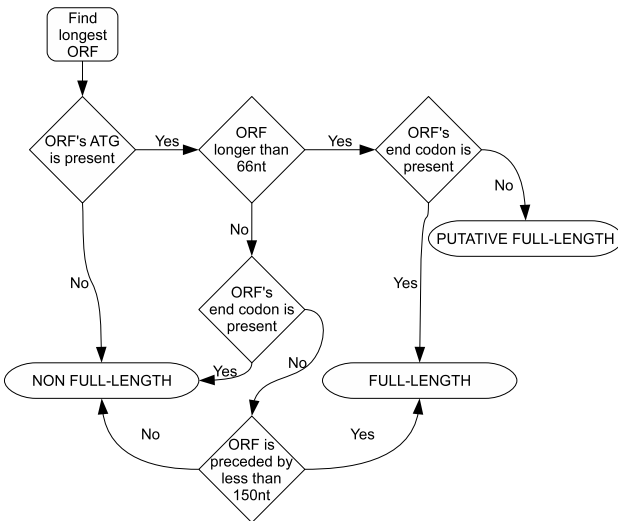


Fig. 3. Processing steps when there are no alignments

2.3 Analysis Without Alignments

When BLAST renders no similarities for an EST, the analysis must consider only the features of that sequence. The EST is supposed to contain a high quality sequence since all automatic sequencers provide a quality value to qualify the removal of any ambiguous fragments [10, 11, 12, 13]. Hence, Full-Lengther will look for the longest possible open reading frame (ORF). All possible ORFs are determined by a simple

analysis locating start and stop codons, and the longest one is then selected. The rationale for ORF classification is explained in Fig. 3. ORFs longer than 66 nucleotides are full-length when start (ATG) and end codons are present or when the ATG is at no longer than 150 nucleotides from the 5' end without an in-frame end codon.

The minimum length to consider an acceptable ORF is 66 nucleotides (22 codons) because the probability to reach an end codon is 1/21 codons, so that shorter ORFs cannot be considered significant.

3 Interface

Full-Lengther has a web interface that can process a set of sequences at once. It is actually very easy to use since users only have to fill out a web form (Fig. 4). The sequences to analyze can be provided by pasting text or choosing a file to upload. The valid input format for sequences is a FASTA file containing the set of sequences to be processed. The default parameters for the algorithm are suitable for most purposes and hence, it is not compulsory to change them. The meaning of parameters is as follows:

Enter sequence(s) in FASTA format (comment line is compulsory)

or choose a file to upload

Fig. 4. Full-Lengther form

- A check-box to mark whether we want to use TrEMBL when there are no hits found using Swiss-Prot. By default the use of TrEMBL is disabled.
- The expect cutoff is a BLAST output filter that removes any subject sequence in the alignment that have an expect value greater than specified (10^{-6}).
- The last parameter is the number of amino acid that will be considered as maximum distance from the beginning of the sequence to the protein which is aligned to, to be considered full-length. By default this value is 15 amino acids.

Seq.Name	Min. Start	# Hits found	# Families found	Seq. Length (nt)
GEM-05-1D07-T3-A1.ab1	28	4	1	698
GEM-48-D07-T3-96-F1.ab1	59	6	6	640
GEM-04-1C08-T3-A1.ab1	25	2	1	689
GEMINI-81-D07-T3-96-F1.ab1	174	3	1	562
GEM-48-D10-T3-96-F1.ab1	19	5	4	570
GEM-05-1E01-T3-A1.ab1	261	1	4	693
GEMINI-82-G02-T3-96-F1.ab1	87	3	2	647
GEM-49-F06-T3-96-L1.ab1	158	2	4	710
GEM-05-1E02-T3-A1.ab1	379	4	1	658
GEM-05-1G08-T3-A1.ab1	428	2	3	721
GEM-48-H05-T3-96-F1.ab1	49	20	4	685
GEM-05-1C09-T3-A1.ab1	-	0	0	629

# FULL-LENGTH	2
# Putative full-length	2
# Non full-length	8
# Total seqs	12

Fig. 5. Summary of sequences

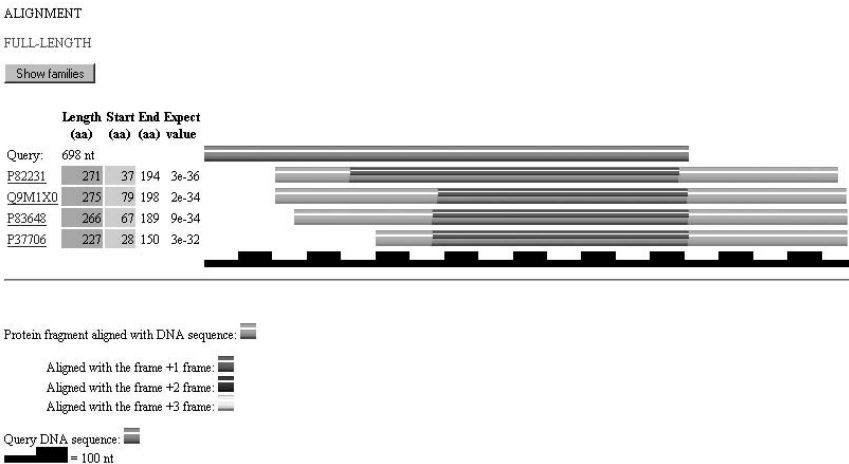


Fig. 6. EST sequence alignment result. Aligned hits are shown in green, unpaired sequences are in gray and the putative ORF appears in blue.

Once the form is submitted and the process is finished, access to the results under the form of a list of sequences (Fig. 5) is allowed. Sequences are highlighted according to its category, including information about length and position of start

codon. A detailed view of each EST sequence can be presented, showing several sections (Fig.6):

1. The first section shows the ORFs result, where the best and the longest ORF are highlighted.
2. The second section shows the Testcode result for the best ORF.
3. The last section is the graphical list of all the EST alignments. Different colours are used to identify information.

4 Discussion

Full-Lengther has been evaluated with real data to verify the correctness and accuracy of its results. In contrast to similar programs such as TargetIdentifier, Full-lengther shows a simple and intuitive output, making easy to form an accurate idea of the results. Moreover, it is designed as a library so that it can be easily integrated in other algorithms to build workflows for EST annotation.

Since Full-Lengther emerged from collaboration between biologists and computer scientists, it is continuously improved and new features are included. For example, it is planned to use databases of fully sequenced genomes in order to avoid partial sequences that can lead to erroneous interpretations.

Acknowledgements. This work is supported by the Spanish MEC grants AGL-2006-07360/FOR and BIO2006-06216 as well as the Junta de Andalucía grant AGR-663 and founding to the research groups CVI-114, TIC-160, and TIC-113.

References

1. Chen Y, Carlis J, Shoop E, Riedl J. International Joint Conference on Artificial Intelligence 2001, Workshop on Inconsistency in Data and Knowledge, Seattle, WA (2001)
2. Terol J, Conesa A, Colmenero JM, Cercos M, Tadeo F, Agusti J, Alos E, Andres F, Soler G, Brumos J, Iglesias DJ, Gotz S, Legaz F, Argout X, Courtois B, Ollitrault P, Dossat C, Wincker P, Morillon R, Talon M. Analysis of 13000 unique Citrus clusters associated with fruit quality, production and salinity tolerance. *BMC Genomics* 8, 31 (2007)
3. Pedersen AG, Nielsen H. Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis. *Proc Int Conf Intell Syst Mol Biol* 226–233 (1997)
4. Iseli C, Jongeneel CV, Bucher P. ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. *Proc Int Conf Intell Syst Mol Biol* 138–148 (1999)
5. Salamov A, Nishikawa T, Swindells MB. Assessing protein coding region integrity in cDNA sequencing projects. *Bioinformatics* 14:384–390 (1998)
6. Min XJ, Butler G, Storms R, Tsang A. OrfPredictor: predicting protein-coding regions in EST-derived sequences. *Nucleic Acids Res* 33:677–680 (2005)
7. Nadershani A, Fahrenkrug SC, Ellis LBM. Comparison of computational methods for identifying translation initiation sites in EST data. *BMC Bioinformatics* 5, 14 (2004)

8. Nishikawa T, Ota T, Isogai T. Prediction whether a human cDNA sequence contains initiation codon by combining statistical information and similarity with protein sequences. *Bioinformatics* 16:960–967 (2000)
9. Min XJ, Butler G, Storms R, Tsang A. TargetIdentifier: a webserver for identifying full-length cDNAs from EST sequences. *Nucleic Acids Research* 33:669–672 (2005)
10. Walther D, Bartha G, Morris M. Basecalling with LifeTrace. *Genome Res* 11:875-888 (2001)
11. Gordon D, Abajian C, Green P. Consed: a graphical tool for sequence finishing. *Genome Res* 8:195-202 (1998)
12. Huang X, Madan A. CAP3: A DNA sequence assembly program. *Genome research* 9:868-877 (1999)
13. Falgueras J, Lara A, Cantó FR, Pérez-Trabado G, Claros MG. SeqTrim: a validation and trimming tool for all purpose sequence reads. *This issue* (2007)

Discovering the Intrinsic Dimensionality of BLOSUM Substitution Matrices Using Evolutionary MDS

Juan Mádez, Antonio Falcó, Mario Hernández, and Javier Lorenzo

Intelligent Systems Institute
Univ. Las Palmas de Gran Canaria, Spain

Abstract. The paper shows the application of the multidimensional scaling to discover the intrinsic dimensionality of the substitution matrices. These matrices are used in Bioinformatics to compare amino acids in the alignment procedures. However, the methodology can be used in other applications to discover the intrinsic dimensionality of a wide class of symmetrical matrices. The discovery of the intrinsic dimensionality of substitutions matrices is a data processing problem with applications in chemical evolution. The problem is related with the number of relevant physical, chemical and structural characteristic involved in these matrices. Many studies have dealt with the identification of relevant characteristic sets for these matrices, but few have concerned with establishing an upper bound of their cardinality. The methodology of multidimensional scaling is used to map the substitution matrix information in a virtual low dimensional space. The relationship between the quality of this process and the dimensionality of the mapping provides clues about the number of characteristics which better represents the matrix. To avoid the local minima problem, a genetic algorithm is used to minimize the objective function of the multidimensional scaling procedure. The main conclusion is that the number of effective characteristics involved in substitution matrices is small.

1 Introduction

The molecular evolution predicts that variations in spices are highly related to the physical-chemical factors involved in protein function and folding. The modelling of evolution at protein level is usually accomplished by using matrices of mutation probability among amino acids. These collect the co-occurrence probability of each amino acid pair in homologous sequences, which have some defined evolutionary distance or rate of conserved residues. Therefore, substitution rates at molecular level and physicochemical properties seem to be of central interest in biological evolution. In special, the knowledge about the effects of amino acid properties in the substitution probability can be of great interest in order to understand the evolution mechanisms.

In practical computational tasks, substitution matrices are introduced to score the substitution of amino acid residues in sequence alignment procedures to reveal homologies. Different substitution matrices can be constructed according with the selection of the set of representative sequences in a biological framework. Eg. PAM matrices[1] are constructed from sequences with evolutionary relations, while BLOSUM matrices[2] are constructed from block sequences that have a similarity ratio.

In Data Mining, the problems concerning with dimensionality reduction or dimensionality discovery must deal with the concept of *intrinsic dimensionality* [3] of their data. In intrinsic dimensionality research, most approaches [4, 5] deal with high dimensional databases and try to reduce the data into a few dimensions by applying methods of multidimensional scaling while the precision in database continues to be high [6]. Intrinsic dimensionality of substitution matrices can be obtained by using two approaches: characteristic independent or dependent. Characteristics dependent analysis are weak techniques because they involve some problems. The main one is related with the choice of the property set. To avoid undesired exclusions, its cardinality must be high. There are some exhaustive compiled sets of amino acid characteristics which cardinality is about several hundreds; one of the best is the AAindex database [7]. The second problem is related to how obtain the intrinsic dimensionality from these massive sets. Feature selection procedures as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are the basic approaches. These procedures provide results as linear combinations of the characteristics contained in the original set. PCA is optimal in applications where the second order statistical parameters – as in gaussian case– define the probabilistic distribution. In the PCA procedure the results are eigenvalues and eigenvectors, that are relevant properties of the characteristics set. A set of orthogonal linear combination of amino acid characteristics for clustering has been obtained [8] from the eigenvectors of the PCA selection procedure from a wide characteristic set. They are significative among the characteristic set itself, but there are doubts about if they are significative in the relationship with the substitution matrices.

The main motivation of this paper is that the discovery of the intrinsic dimensionality of substitution matrices provides an upper bound about the cardinality of the set of relevant properties. Also, the main hypothesis is that the discovery can be directly obtained –independently of the set of properties– from the substitution data itself by using procedures of multidimensional scaling. This paper is mainly concerning with *how many* rather than *what* properties are important in the substitution matrices.

The following of this paper is organized in sections covering the methods, results and conclusion. In the methods section, the substitution matrices are coded based on derived distance matrices. Also, it includes the use of non-linear multidimensional scaling procedures to map the distance in a dimensional space. The result and conclusion sections argue about the intrinsic dimensionality of substitutions matrices.

2 Methods

Substitutions matrices are computed as the log-odds between the relation probability q_{ab} between amino acids pairs in sets of proteins sequences and the independent probability $p_a p_b$

$$s(a,b) = \log \frac{q_{ab}}{p_a p_b} \quad (1)$$

These matrices are symmetrical $s(a, b) = s(b, a)$ and at usual mutation rates verifies $s(a, a) \geq s(a, b)$, but in general the diagonal terms are different: $s(a, a) \neq s(b, b)$. They are like similarity functions, but are not full similarity functions. Many heuristic distance expressions can be obtained from these matrices. The used in this paper is:

$$d(a, b) = s(a, a) + s(b, b) - 2s(a, b) \quad (2)$$

That has the properties of a distance matrix:

$$d(a, b) = d(b, a) \quad d(a, b) \geq 0 \quad d(a, a) = 0 \quad (3)$$

But it is no metric in the general case. The verification of additional properties required to be a metric, which are the if-only-if and triangular properties, depends on the $s(a, b)$ values. Eg. the if-only-if metric property, which requires the following property: $d(a, b) = 0 \leftrightarrow a = b$, is verified if the inequality $s(a, b) \leq s(a, a)$ can be transformed in the most restrictive condition $s(a, b) < s(a, a)$.

This distance is not a general purpose distance among amino acids. It is an evolutionary distance in a defined biological environment, as general or specific as the substitution matrix from which is obtained. A distance matrix among amino acids is a dimensional-less relationship, nothing relates it with a multidimensional coordinates system.

2.1 Multidimensional Scaling

Multidimensional scaling [9], or mapping, is the process of finding a configuration of points in a multidimensional space as lower dimensionality as possible, whose interpoint distances correspond, with the lowest error possible, to some previous existent similarities or dissimilarities data [10]. In some cases, the original data have a dimensional representation and mapping tries to find a good representation in a lower dimensional space. In this case, feature reduction procedures as PCA and ICA techniques are used. In other cases –as in this paper– the original distance does not imply a coordinate representation. It is a set of interrelations among concepts without any explicit dimensional counterpart.

Two distance types are involved in the mapping process. The first, $d(a, b)$, is the original coordinate-less measure. The second, $d_X(a, b)$, is the distance in a dimensional space where X denotes the coordinates system of amino acids. The problem to be solved is how to compute $d_X(a, b)$ as a good approximation of $d(a, b)$, which implies the computation of the vector set: $X(a)$. The Sammon method [11] is a non-linear mapping procedure that provides a good ratio of result quality to computational complexity [12–14]. It maps a distance function to a reduced dimensionality space based on the minimization of an objective function by assigning trial coordinates to each amino acid.

The goal function is related to the relative error between the original dimensionless distances, $d(a, b)$, and the dimensional ones, $d_X(a, b)$. Consequently, several solutions can be obtained if some local minima exist. The Sammon method requires the minimization of the goal function $G(X)$ which likes a relative error of the mapping process:

$$\min_X G(X) = \frac{\sum_a \sum_{b < a} \frac{[d_X(a,b) - d(a,b)]^2}{d(a,b)}}{\sum_a \sum_{b < a} d(a,b)} \quad (4)$$

The optimal solution X is not unique. There are some freedom degrees related to the geometrical transformations that preserve the distance d_X , eg. translations, rotations and sign inversion. The dimensional distance function is based on the L_2 norm:

$$d_X(a,b) = \left[\sum_{i=1}^n |X_i(a) - X_i(b)|^2 \right]^{\frac{1}{2}} \quad (5)$$

To increase the efficiency of computational procedures, the vector $X(a) \in \mathfrak{R}^n$ provided by the optimization procedure is transformed into the $Y(a)$ vector in the integer range $[0, 255]$ by using geometric transformations of translation and scaling. This discrete version can be coded by using integer arithmetic, more efficient than the float point one. The translation to coordinates origin does not modify the distances, but the scaling to fit the $[0, 255]$ range modifies the distance with a constant factor ρ related with the scaling factor. The relationship between the distances computed by mean of the two vector type is: $d_X(a, b) = \rho d_Y(a, b)$.

Some optimization methods, such as evolutionary and gradient based, can be used to achieve the minimization of the goal function. Gradient procedures have better convergence around a local minimum, while genetic procedures allow a better global optimization by considering several local minima. Many solutions are expected in the proposed problem covering a wide range of local minima due to non-linearity and geometrical transformations. In this paper, a genetic algorithm is used to obtain a solution which is afterward refined by applying a gradient procedure based on the Quasi-Newton algorithm. Genetic algorithms are good to explore the space domain, avoiding the local minima problem. However, in practice after a large number of iterations they are mainly working in the refinement of a local minimum, but the gradient procedures are more efficient for this task.

3 Results

The proposed methodology can be applied to any symmetrical matrix which can generate a distance matrix. The substitution matrices used in Bioinformatics for alignments procedures verify this property. The most used substitution matrices are the BLOSUM family. Figure 1 shows the graphical representation of the optimal value $G(n)$ of the goal function in Equation (4) vs the dimensionality n of the mapping space. A fast convergence toward null values is obtained when the dimensionality increases, which implies a high decreasing in the marginal relevance of additional dimensions. Therefore, after small dimensionality values of three or four, few additional gain can be obtained with additional dimensions. This could be interpreted as most of the information contained in the substitution matrix is related with a few orthogonal –independent– factors.

Table 1 shows coordinates from 1 to 5 dimensionality. These dimensions are the most relevant. Remark that due to the random nature of genetic algorithms, two different runs of the code can provide different solutions in the X vector, but similar – no too much different – values in G .

The coordinates generated by the mapping process are virtual meaning-less data. An arbitrary set of rotations, translations and sign inversions can be involved in the optimization process because these transformations are distance invariant. No control exists on the spatial organization of the solutions, and it is a fact that random conditions are involved in their generation. Therefore, a first appreciation is that no relationship exists among the mapping coordinates and any variable with meaning at biological, physical or chemical level. However, without a full refusing of these appreciations, some kind of semantic organization can be found in the mapped space.

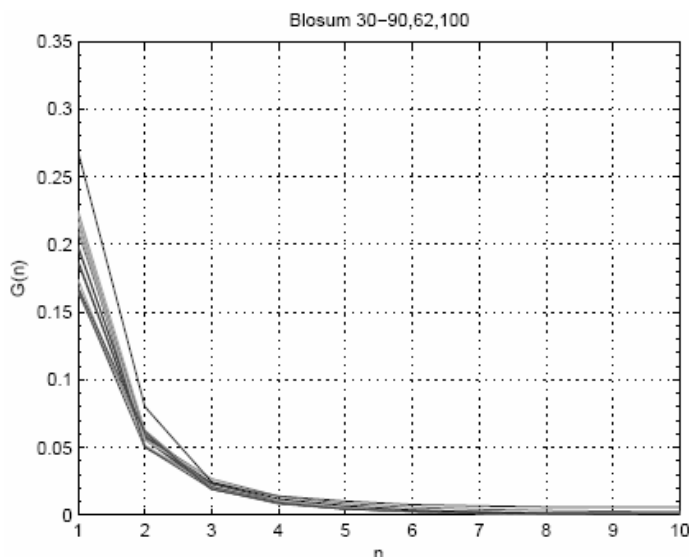


Fig. 1. Optimal goal value $G(n)$ vs the dimensionality n of the mapping space for BLOSUM matrices from 30 to 90 every 5, and also for 62 and 100 values. The result suggests that after dimensionality three or four few gain is obtained by introducing additional dimensions.

The general formulation of the data mining and pattern analysis problem to be solved in order to discover the meaning of virtual coordinates is as follows: how to relate the virtual coordinates $X_i(a)$ or $Y_i(a)$ obtained from the mapping of the distance $d(a, b)$ with a set of characteristic $Q_j(a)$ with previous semantic, which in general are no orthogonal. This is an open problem which requires future studies.

As an illustrative contribution on the discovery of some semantic in the virtual coordinates, a high spatial organization of the amino acid groups can be found in the provided results. Amino acids can be grouped according with their physical-chemical properties. Some groups –aliphatic or aromatic– are related to the chemical structure; others –tiny or small– are grouped with the molecular size; the polar and charged groups are related to the electric activity, and hydrophobic group is related with their

Table 1. The Y coordinates from 1 to 5 dimensionality of BLOSUM 62 matrix

n	1	2	3	4	5
aa	Y ₁	Y ₁ Y ₂	Y ₁ Y ₂ Y ₃	Y ₁ Y ₂ Y ₃ Y ₄	Y ₁ Y ₂ Y ₃ Y ₄ Y ₅
A	140	111 84	148 142 97	102 105 63 116	102 50 67 207 45
R	180	69 175	21 48 115	72 2 162 96	200 45 15 202 156
N	203	3 134	74 166 199	42 95 174 47	235 40 72 136 64
D	216	0 80	45 212 139	64 114 100 0	184 147 92 97 43
C	31	220 37	229 160 0	95 44 26 251	39 34 255 225 54
Q	164	65 131	52 93 154	134 44 149 51	178 127 85 181 158
E	190	31 112	24 138 144	104 56 123 22	190 136 62 138 123
G	228	31 34	166 209 170	0 160 92 99	127 8 7 113 0
H	239	35 219	38 61 239	115 103 252 63	187 26 47 47 181
I	97	173 99	174 57 68	182 102 95 155	40 20 86 242 126
L	89	171 125	148 30 59	170 76 116 175	41 70 92 241 144
K	173	53 153	24 92 90	89 0 120 73	173 106 0 209 117
M	115	142 137	108 34 62	162 44 142 157	87 65 88 241 180
F	59	177 181	176 1 148	150 169 158 187	12 24 65 127 184
P	255	87 0	0 156 18	129 73 0 31	86 199 65 205 16
S	154	74 99	103 149 119	59 88 103 97	147 66 94 164 59
T	131	107 57	103 139 47	63 57 83 146	171 39 137 228 86
W	0	200 255	238 22 255	38 128 241 255	0 86 170 0 180
Y	71	133 213	127 0 193	139 165 207 140	78 0 103 101 215
V	105	158 90	166 78 65	178 96 89 133	62 10 84 248 115
ρ	0.144	0.113	0.088	0.089	0.075

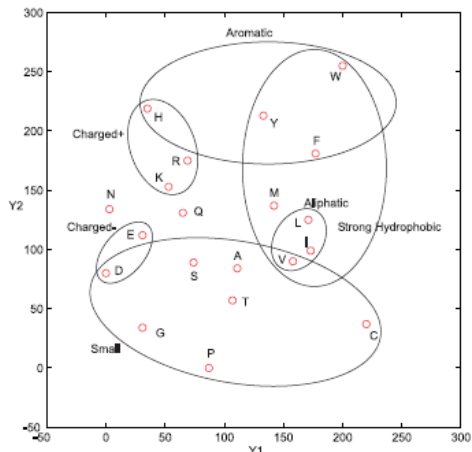
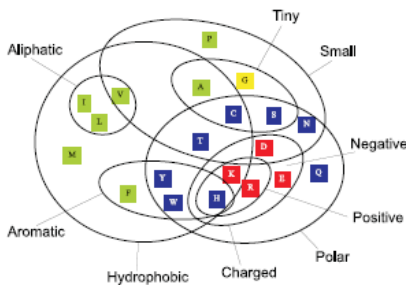


Fig. 2. At left the biochemical groups of amino acids. At right a representation for the $n=2$ mapping of BLOSUM 62 showing the spatial organization of some amino acid groups. Relevant groups are spatially organized as clusters. Small group tends to be in low Y_2 values while hydrophobicity tends to high Y_1 values.

affinity with water. It has been shown [15] that the amino acid groups have a high level of spatial organization when they are mapped in a two dimensional space obtained from the reduction of the whole AAindex database to two index according

with their correlations. In this line, but from a different approach, Figure 2 shows the two dimensional mapping of the amino acids using the Y coordinates obtained from Table 1.

The small group –except N amino acid– is mapped at the bottom of the map. It forms a region at low Y_2 coordinate and extends along the whole range of Y_1 one. The aromatic group –VIL– conforms a cluster included into the strong hydrophobic group –WFYMVIL– located at higher values of the Y_1 coordinate. The opposite groups –charged– have the lower values in this coordinate. The hydrophobic –or its opposite hydrophilic– property of amino acids is fundamental in the dynamics and structure of proteins[16]. Due to that the biological matter is basically an aqueous solution, the water affinity is essential in the relation of a protein with its environment. The mutations with significative changes in the water affinity have a high probability of generate disfunctions, and consequently they have a low survival probability, therefore, there are lost in the evolution process.

4 Conclusion

The aim to be exhaustive in the discovery of relevant characteristics in the substitution matrices increases the cardinality of the characteristic set. Finding the most relevant characteristics is more critical as the number of properties is increased, thus the definition of an upper limit is a good choice to avoid the computational explosion. This assertion expresses qualitatively the advantage of determining the intrinsic dimensionality of substitution matrices from themselves, instead of estimates it from a big and unclosed set of characteristic. This paper hypothesizes that the computation of the intrinsic dimensionality – the *how many characteristics* problem– is better achieved as a characteristic independent procedure. However, this does not exclude that the *what characteristic* analysis is necessary.

A lot of factors could be implied in the substitutions matrices, but the main result of this paper suggested that no too much must be considered. About three or four factors are important for the BLOSUM test case. Multidimensional mapping of substitution matrices could be considered an useful methodology to know about how many independent factors are involved in these matrices. The minimizing of a goal function related with the relative error of mapping has been used. The fast decreasing of goal function at small dimensionality suggests a small number of independent characteristics, and also that additional factors have a small contribution in the substitution matrix.

References

1. Dayhoff, M., Schwartz, R., Orcutt, B.: Atlas of Protein Sequence and Structure. Volume 5. Nat. Biomed. Res. Found. (1978)
2. Henikoff, S., Henikoff, J.: Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. 89 (1992) 10915–10919
3. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Morgan Kaufmann (1990)
4. Chakrabarti, K., Mehrotra, S.: Local dimensionality reduction: A new approach to indexing high dimensional spaces. In: The VLDB Journal. (2000) 89–100

5. Kanth, K.V.R., Agrawal, D., Abbadi, A.E., Singh, A.: Dimensionality reduction for similarity searching in dynamic databases. *Computer Vision and Image Understanding: CVIU* 75(1–2) (1999) 59–72
6. Aggarwal, C.C.: On the effects of dimensionality reduction on high dimensional similarity search. In: *Symposium on Principles of Database Systems*. (2001)
7. Kawashima, S., Ogata, H., Kanehisa, M.: Aaindex: amino acid index database. *Nucleic Acids Res.* 27 (1999) 368–369
8. Venkatarajan, M.S., Braun, W.: New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical-chemical properties. *J. Mol. Model* 7 (2001) 445–453
9. Cox, T., Cox, M.A.: *Multidimensional Scaling*. Chapman and Hall (1994)
10. Duda, R., Hart, P., Stork, D.: *Pattern Classification*. John Wiley and Sons (2001)
11. Sammon, J.: A nonlinear mapping for data structure analysis. *IEEE Trans. Computers* 18 (1969) 401–409
12. Li, S., de Vel, O., Coomans, D.: Comparative performance analysis of non-linear dimensionality reduction methods. Technical report, James Cook Univ. (1995)
13. Backer, S.D., Naud, A., Scheunders, P.: Nonlinear dimensionality reduction techniques for unsupervised feature extraction. *Pattern Recognition Letters* 19 (1998) 711–720
14. Scheunders, P., Backer, S.D., Naud, A.: Non-linear mapping for feature extraction. *Lecture notes in computer science* 1451 (1998) 823–830
15. Hagerty, C., Kulikowski, C., Muchnik, I., Kim, S.: Two indices can approximate 402 amino acid properties. In: *Proc. IEEE Int. Symp. Intelligent Control, Intelligent Systems and Semiotics*. (1999) 365–369
16. Gerstein, M., Levitt, M.: Simulating water and the molecules of life. *Scientific American* (1998) 100–105

Autonomous FYDPS Neural Network-Based Planner Agent for Health Care in Geriatric Residences

Juan F. de Paz, Yanira de Paz, Javier Bajo, Sara Rodríguez,
and Juan M. Corchado

Departamento Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n 37008, Salamanca, Spain
{fcofds, yanira, jbajope, srg, corchado}@usal.es

Abstract. This paper presents an autonomous intelligent agent developed for health care in geriatric residences. The paper focuses on the construction of an autonomous agent which incorporates a model of human thinking, such as reasoning based on past experiences. The work here presented focuses in the development of the CBP internal structure. The planning mechanism has been implemented by means of a novel FYDPS neural network. The system has been tested and this paper presents the results obtained.

Keywords: Multi-agent System, CBR, FYDPS neural network.

1 Introduction

In this article we present a novel planning system based on the combination of neuronal networks with CBP (Case-based planning) systems [7]. Case-based planning allows us to retrieve past experiences to create a new plan which lends the system a large capacity for learning and adaptation [7]. The neuronal networks proposed within this research framework are self-organised, based on Kohonen networks [10], but present certain improvements (FYDPS Neural Network) [11]. These improvements allow the network to reach a solution much more rapidly. Besides, once a solution has been reached, it is possible to make modifications taking restrictions into account (in this study time restrictions). The new planning mechanism is integrated within AGALZ (Autonomous aGent for monitoring ALZheimer patients) [5], [6], a planning agent that works in conjunction with complementary agents into a prototype multi-agent system (ALZ-MAS: ALZheimer Multi-Agent System). The results obtained are compared to those obtained with the previous geodesic planner used by AGALZ.

This work focuses in the development of deliberative agents using a case-based planning [1] architecture, as a way to implement adaptive systems to improve assistance and health care support for elderly and people with disabilities, in particular with Alzheimer's. Agents in this context must be able to respond to events, take the initiative according to their goals, communicate with other agents, interact with users, and make use of past experiences to find the best plans to achieve goals, so we propose the development of a deliberative agent that incorporates a CBP mechanism, specially designed for planning construction. CBP-BDI facilitates learning and

adaptation, and therefore a greater degree of autonomy than that found in pure BDI (Believe, Desire, Intention) architecture [2].

During the last three decades the number of Europeans over 60 years old has risen by about 50%. Today they represent more than 25% of the population and it is estimated that in 20 years this percentage will rise to one third of the population, meaning 100 millions of citizens [3]. This situation is not exclusive to Europe, since studies in other parts of the world show similar tendencies. The importance of developing new and more reliable ways to provide care and support to the elderly is underlined by this trend [3], and the creation of secure, unobtrusive and adaptable environments for monitoring and optimizing health care will become vital. Tomorrow's health care institutions will be equipped with intelligent systems capable of interacting with humans. Multi-agent systems and architectures based on intelligent devices have recently been explored as supervision systems for medical care for the elderly or Alzheimer patients, these intelligent systems aim to support them in all aspects of daily life, predicting potential hazardous situations and delivering physical and cognitive support.

In the next section the autonomous planner agent is presented. Then the new planning mechanism is described, finalizing with a case study and results and conclusions obtained after the implementation of a prototype into a real scenario.

2 Autonomous Health Care Agent

The CBP is based on the generation of plans from cases. The deliberative agents, proposed in the framework of this investigation, use this concept to gain autonomy and improve their guiding capabilities. The relationship between CBP systems and BDI agents can be established by implementing cases as beliefs, intentions and desires which lead to the resolution of the problem. In a CBP-BDI [5], [6] agent, each state is considered as a belief; the objective to be reached may also be a belief. The intentions are plans of actions that the agent has to carry out in order to achieve its objectives [8], so an intention is an ordered set of actions; each change from state to state is made after carrying out an action (the agent remembers the action carried out in the past, when it was in a specified state, and the subsequent result). A desire is any of the final states reached in the past (if the agent has to deal with a situation, which is similar to one in the past, it will try to achieve a similar result to the one previously obtained). Below, the guiding mechanism, used by the CBP-BDI agent, is presented: Let $E = \{e_0, \dots, e_n\}$ the set of the possible rooms and places in the residence.

$$a_j : E \begin{array}{c} \rightarrow \\ \rightarrow \end{array} \begin{array}{c} E \\ a_j(e_i)=e_j \end{array} \quad (1)$$

An Agent plan is the name given to a sequence of actions (1) that, from a current state e_0 , defines the path of states through which the agent passes in order to offer to the nurse the better path. Below, in (2), the dynamic relationship between the behaviour of the agent and the changes in the environment is modelled. The behaviour of agent A can be represented by its action function $a_A(t) \forall t$, defined as a correspondence between one moment in time t and the action selected by the agent,

$$Agent\ A = \{a_A(t)\}_{t \in T \subseteq N} \tag{2}$$

From the definition of the action function $a_A(t)$ a new relationship that collects the idea of an agent’s action plan (3) can be defined,

$$p_A : \begin{matrix} Tx_A \\ (t, a_A(t)) \end{matrix} \begin{matrix} \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} A \\ p_A(t) \end{matrix} \tag{3}$$

in the following way,

$$p_A(t_n) = \sum_{i=1}^n a_{iA}(t_i - t_{i-1}) \tag{4}$$

Given the dynamic character that we want to print onto our agent, the continuous extension of the previous expression (4) is proposed as a definition of the agent plan, in other words (5) -

$$p_A(t_n) = \int_0^n a_A(t) dt \tag{5}$$

The variation of the agent plan $p_A(t)$ will be provoked essentially by: the changes that occur in the environment and that force the initial plan to be modified, and the knowledge from the success and failure of the plans that were used in the past, and which are favoured or punished via learning. The planning is carried out through a neural network based on the Kohonen network [9]. The neurons are organized in a two-layer unidirectional architecture. The learning method is presented as follows: (The equations are presented in the order in that they should be executed).

- To present the input vector $X^p = (x_1^p, \dots, x_i^p, \dots, x_N^p)^T$ in the input layer.
- The weightings initially take random weightings in (0,1).
- To calculate the intensity of the neurons of the output layer. The Euclidean distance:

$$y_k = \sqrt{\sum_{i=1}^N (x_i - w_{ki})^2} \tag{6}$$

- To determine the winning neuron that will be that of smaller Euclidean distance.
- To upgrade the weights of the neurons that connect the input layer with the output neuron:

$$w_{ki}(t+1) = w_{ki}(t) + \eta(t) g(k, h, t)(x_i(t) - w_{ki}(t)) \tag{7}$$

Where:

$$g(k, h, t) = e^{\frac{-k-h^2}{2R(t)^2}} \tag{8}$$

(Gaussian function), therefore the formula is the following:

$$w_{ki}(t + 1) = w_{ki}(t) + \eta(t)e^{\frac{-|k-h|^2}{2R(t)^2}} (x_i(t) - w_{ki}(t)) \tag{9}$$

Where w_{ki} is the weight of the connection between the input neuron i and the output neuron k ; t is the iterations; η is the learning rate; h is the position of the winning neuron; k is the neuron of the output layer; and i is the neuron of the input layer. In $k-h$ a distance is calculated between the neurons. The Euclidean distance has been used.

3 Defining Self-Organising Neural Network in a Novel Way: FYDPS Neural Network

The basic Kohonen network [10] cannot be used to resolve our problem since it attempts to minimise distances without taking into account any other type of restriction, such as time limits. In the present study a planner is described that is based on Kohonen networks but with a number of improvements (FYDPS Neural Network) [13] that allow us to reach a solution far more rapidly. Furthermore, once a solution has been reached, it is re-modified in order to take restrictions into account.

3.1 Objective 1: Reach a Solution More Rapidly

As such, for this modification of the basic algorithm (FYDPS), we are aiming to make the solution search more agile and in order to achieve this, the basic vicinity function used in the Kohonen network is modified and the number of neurones in the output layer corresponds to the places that the subject wishes to visit. The topology of the neural network being considered is described below. The input layer is formed by two neurons, each one of those receives one of the coordinates of the patient's room presented as input. A vector of neurons is used of size the same as the number of places to visit by nurses of the problem in the output layer, as in [9] [11]. The number of neurons in the output layer isn't modified. Let $x_i \equiv (x_{i1}, x_{i2}) \quad i = 1, \dots, N$ the coordinates from the patient's room i and $n_i \equiv (n_{i1}, n_{i2}) \quad i = 1, \dots, N$ the coordinates the neurons i in \mathfrak{R}^2 . N rooms will be visited by the fixed nurse. Then there will be: Two neurons in the input layer and N neurons in the output layer. It will be considered a vicinity function decreasing with the number of iterations.

$$g(k, h, t) = \text{Exp} \left[\left(-\frac{|k-h|}{N/2} \right)^{\frac{\text{Máx}_{\substack{i,j \in \{1, \dots, N\} \\ i \neq j}} \{f_{ij}\}} - \sqrt{(n_{k1} - n_{h1})^2 + (n_{k2} - n_{h2})^2}}{\text{Máx}_{\substack{i,j \\ i \neq j}} \{f_{ij}\}} - \lambda \frac{|k-h|t}{\beta N} \right] \tag{10}$$

λ and β are determined empirically, their respective values are: 5 and 50. t is the current iteration. $\text{Exp}[x] = e^x$. N is the number of the rooms that are visited by a fixed nurse and f_{ij} is the distance given by the Floyd Algorithm [14].

The radius of final vicinity should be similar to 0 so that the winner is only upgraded. Iteratively the group of rooms will be presented, so that the weights of the neurons approach the coordinates of the rooms. When concluding the process, there will be a neuron associated to each room. To determine the route to follow, we will leave the room associated to the neuron i to the associated to $i+1$, for $i=1, 2, \dots, N$, passing the whole vector of neurons. To close the road, the last tract will be given by the route that joins the patient's room associated to the neuron N with the associated to the neuron 1 . The distance of the road will be given by the sum of the distances between the successive couples of rooms of the road. The learning rate is a decreasing function:

$$\eta(t) = \text{Exp} \left[-\sqrt[t]{\frac{t}{\beta N}} \right] \quad (11)$$

The function of activation of the neurons is the identity. When the system stops, the route to continue will be given by the weights of the neurons that will be very closer to the coordinates of the rooms. To know which the following room in the journey, we pass to the following element of the vector of neurons. The neurons are stored in a vector that contains the weights of each one in the current instant. So that the vector defines a ring, the neuron n_1 is the following to the n_N considered.

With a big radius of vicinity, in the first iterations of the algorithm the victory of a neuron affects great part of the map, so that a global self-organization takes place. If the radius decreases, the effect of a victory affects every time a part smaller in the map, so that the criterion to stop the learning of the network is that the distance among rooms cannot be optimized more. The initial number of total iterations is of $T_1 = \beta N$ (first phase). When $t = \beta N$, all the couples of possible neurons are exchanged (exchanging their weights) in the obtained ring of neurons, if the distance is optimized then the learning isn't finished.

In general, in the phase Z , the total number of interactions to carry is:

$$T_Z = T_{Z-1} - \frac{T_{Z-1}}{Z} \quad (12)$$

The aim of these phases is to eliminate the crossings. Concluded the iterations of each phase is proven if the distance is already optimized, in such a way that in the phase that stops the learning, the distance is minimum. The primary objective - to achieve artificial neural networks that are faster than basic Kohonen networks, applied to the problems that the basic networks resolve - has been achieved. In the apparatus below the necessary modifications are introduced into the algorithm so that the network can take restrictions into account, and therefore be able to resolve other problems that cannot be resolved by basic Kohonen networks.

3.2 Objective 2: Taking Time Restrictions into Account

Instead of using the Euclidean distance, a different distance is used that we call "temporal distance". Without losing any generality it can be supposed that a distance unit is equivalent to a time unit.

$$dt_{ij} \equiv dt(x_i, x_j) \triangleq \text{Máx}\{f_{ij} + t_i, b_j\} \tag{13}$$

Where t_i is the time hended to get to room “ i ” from the previous room plus the time taken on tasks to be carried out within room “ i ” (in other words, the service time in “ i ”) and b_j is the time limit for carrying out the tasks in room “ j ”. In this way, the vicinity function of the network modified from the FYDPS network is:

$$g(k, h, t) = \text{Exp} \left[\left(-\frac{|k-h|}{N/2} \right)^{\frac{\text{Máx}_{i,j \in \{1, \dots, N\}, i \neq j} \{d_{ij}^*\} - \sqrt{(n_{k1} - n_{h1})^2 + (n_{k2} - n_{h2})^2}}{\text{Máx}_{i,j \in \{1, \dots, N\}, i \neq j} \{d_{ij}^*\}}} - \lambda \frac{|k-h|t}{\beta N} \right] \tag{14}$$

With:

$$d_{ij}^* \equiv d^*(x_i, x_j) \triangleq \frac{f_{ij} + t_i}{dt_{ij}} \tag{15}$$

When the network obtains as a result the optimum plan p^* . If the plan p^* is not interrupted, the agent will reach a desired state $e_j \equiv e^*$. In the learning phase, a weighting $w_f(p)$ is stored. With the updating of weighting $w_f(p^*)$, the planning cycle of the CBP motor is completed. Let’s suppose that the agent has initiated a plan p^* but at a moment $t > t_0$, the plan is interrupted due to a change in the environment. The solutions given by the neural network meet the conditions of the Bellman Principle of Optimality [1], in other words, each on of the plan’s parts is partially optimum between the selected points. This guarantees that if g_0 is optimum route for interrupted e_0 in t_1 , because e_0 changes to e_1 , and g_1 optimum route to e_1 that is begun in the state where g_0 has been interrupted, it follows that: $g = g_0 + g_1$ is optimum route to $e = e_0(t_1 - t_0) + e_1(t_2 - t_1)$.

4 Case Study

The Alzheimer ST Residence of Salamanca has been interested in improving the services offered to its patients and has collaborated in the development of a prototype [5], [6]. Figure 1 shows a diagram of the technology implemented in the first floor of the ST Residence of Salamanca. We selected 30 patients to test the system, so the hardware implemented at the Residence basically consisted of 42 RFID door readers, one on each door and elevator, 4 controllers, one at each exit, one in the first floor hall and another in the second floor hall, and 36 bracelets, one for each patient and the nurses. The ID door readers get the ID number from the bracelets and send the data to the controllers which send a notification to the Manager agent. To test the system 30 patient agents, 10 nurse agents, 2 doctor agents and 1 manager agent were instantiated. AGALZ agent is an autonomous planner agent which schedules the nurse’s working day obtaining dynamic plans depending on the tasks needed for each assigned patient. More details can be seen in [5], [6].

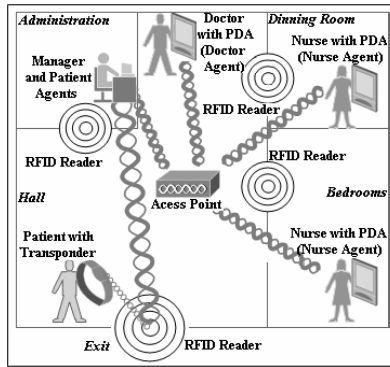


Fig. 1. ALZ-MAS wireless technology organization schema

5 Results and Conclusions

The planning model proposed has been integrated within a previously Developed multiagent system [5], [6]. The tasks executed by nurses were divided in two categories, direct action tasks and indirect action tasks. AGALZ can act on the indirect action tasks. Averages of the time spent by nurses in the carrying out of the tasks for every patient were obtained, having into account that a task depends on the dependency level of a patient and the nurse skill. For the direct action tasks, the following times were obtained for each patient: 35' cleaning, 18' feeding, 8' oral medication, 30' parenteral medication, 25' posture change, 8' toileting, 60' exercise and 10' others. We are especially interested on time spent on indirect action tasks; daily average times obtained for every kind of task before and after the implementation for each planner can be seen on Table 1.

Table 1. Time (minutes) spent on indirect tasks

	Monitoring	Reports	Visits	Other	TOTAL
Before	167	48	73	82	370
Geodesic Planner	105	40	45	60	250
FYDPS Planner	95	37	45	52	229

The system facilitates the more flexible assignation of the working shifts at the residence; since the workers have reduced the time spent on routine tasks and can assign this time to extra activities. Their work is automatically monitored, as well as the patients' activities. The new planning mechanism schedules the tasks in a more efficient way. Table 2 shows a comparison an average time used by each kind of planner for generating a plan.

Table 2. Planners execution time

Planner	Execution Time (seconds)
Geodesic Planner	167
FYDPS Planner	95

In the future, health care will require the use of new technologies that allow medical personnel to carry out their tasks more efficiently [3]. We have shown some potential of deliberative CBP-BDI agents in a distributed multi-agent system focused on health care. In addition, the use of RFID technology [15] on people provided a high level of interaction among users and patients through the system.

Acknowledgements. This work has been partially supported by the MCYT TIC2003-07369-C02-02 and the JCYL-2002-05 project SA104A05. Special thanks to Sokymat by the RFID technology provided and to Telefónica Móviles (Movistar) for the wireless devices donated.

References

1. Bellman R.E., *Dynamic Programming*. (Princeton University Press, Princeton, New Jersey, 1957).
2. Bratman, M.E.: *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, M.A. (1987)
3. Camarinha-Matos L., and Afsarmanesh H., *Design of a Virtual Community Infrastructure for Elderly Care*. PRO-VE'02. Sesimbra, Portugal. (2002).
4. Corchado J.M. and Laza R.: *Constructing Deliberative Agents with Case-based Reasoning Technology*, *Int. Journal of Intelligent Systems*. Vol 18, No. 12, pp. 1227-1241 (2003)
5. Corchado, J.M., Bajo, J., Tapia, D.I.: *ALZ-MAS: Alzheimer's Special Care Multi-Agent System*. *Proceedings of the workshop on Health Care*. ECAI'06. (2006)
6. Corchado J.M., Bajo J., de Paz Y. and Tapia D. I.: *Intelligent Environment for Monitoring Alzheimer Patients*, *Agent Technology for Health Care*. *Decision Support Systems*. Elsevier Science. DOI 10.1016/j.dss.2007.04.008. In Press (2007)
7. Cox M.T., Muñoz-Avila, H. and Bergmann R. *Case-based Planning*. *The Knowledge Engineering Review*, Vol. 00:0, 1–4. c 2005, Cambridge University Press
8. Glez-Bedia M. and Corchado J. M.: *A Planning Strategy based on Variational Calculus for Deliberative Agents*. *Computing and Information Systems Journal*. Vol 10, No 1, 2002. ISBN: 1352-9404, pp. 2-14 (2002)
9. Jin H.D., Leung K.S., Wong M.L., Xu Z.B.: *An Efficient Self-Organizing Map Designed by Genetic Algorithms for the Traveling Salesman Problem*. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 33, no 6. (2003). pp 877-888.
10. Kohonen T.: *Self-Organization and Associative Memory*, Springer Verlag (1984).
11. Leung K.S., Jin H.D., Xu Z.B.: *An Expanding Self-Organizing Neural Network for the Traveling Salesman Problem*. *Neurocomputing*, vol. 62. (2004). pp 267-292.
12. Marrow P. *Nature-Inspired Computing Technology and Applications*. *BT Technology Journal*, vol. 18, n° 4. (2000)
13. Martín Q., De Paz J.F., De Paz Y., Pérez E., *Solving TSP with a Modified Kohonen Network*, *European Journal of Operational Research*, (2007).
14. Martín Q., Santos M.T., De Paz Y., *Operations Research: Resolute Problems and Exercises*, Pearson, (2005) 189-190.
15. Sokymat. Sokymat. <http://www.sokymat.com>. (2006)

Structure-Preserving Noise Reduction in Biological Imaging

J.J. Fernández^{1,2}, S. Li¹, and V. Lucic³

¹ MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK

² Dept. Computer Architecture, University of Almería, Almería 04120, Spain

³ Dept. Structural Biology, Max Planck Institute of Biochemistry, Martinsried, Germany
jjfdez@ual.es

Abstract. An approach for noise filtering based on anisotropic nonlinear diffusion is presented. The method combines edge-preserving noise reduction with a strategy to enhance local structures and a mechanism to further smooth the background. The performance is illustrated with its application to electron cryotomography, a leading imaging technique for visualizing the molecular architecture of complex biological specimens. A challenging task in this discipline is to increase the extremely low signal-to-noise ratio to allow visualization and interpretation of the three-dimensional structures. The filtering method presented here succeeds in substantially reducing the noise with excellent preservation of the structures.

1 Introduction

The advent of biological imaging technology has made it possible to observe, directly or indirectly, the molecular and cellular architecture and interactions that underlie essential functions within cells and tissues [1], [2]. The availability of imaging techniques (optical microscopy, confocal microscopy, electron microscopy, electron tomography, just to name a few) in biology laboratories is growing rapidly. So does the need for sophisticated image processing methods that facilitate analysis and interpretation at different scales of resolution and complexity. Image processing is even needed for analysis of other biological data, e.g. microarrays.

Noise reduction is paramount for proper interpretation or post-processing of the images. Standard linear filtering techniques based on local averages or Gaussian kernels succeed in reducing the noise, but at expenses of blurring edges and features. Anisotropic nonlinear diffusion (AND) is currently one of the most powerful noise reduction techniques [3]. AND achieves better feature preservation as it adaptively tunes the strength and direction of the smoothing to the local structures found in the image. It preserves edges and enhances some features, thus considerably increasing the signal-to-noise ratio (SNR) with no significant distortions of the signal. Pioneered in 1990 by Perona and Malik [4], in the last decade AND has grown up to become a well-established tool for denoising multidimensional images [3], [5], [6], [7].

Electron cryotomography (cryoET) is the leading technique to elucidate the molecular architecture of biological specimens in their native state [8]. CryoET produces extremely low contrast three-dimensional (3D) density maps (known as

“tomograms” in the field). The poor SNR severely hinders visualization and interpretation. Therefore sophisticated filtering techniques are indispensable [7].

This article presents an approach to anisotropic nonlinear filtering for cryoET. The method combines structure-preserving noise reduction with a strategy for enhancement of planar and curvilinear local structures. We illustrate the method with its application to several 3D maps of biological specimens obtained by cryoET. The applicability of the method to other biological imaging techniques is straightforward.

2 Review of Anisotropic Nonlinear Diffusion

Conceptually speaking, AND accomplishes structure-preserving denoising by tuning the strength of the smoothing along different directions based on the local structure estimated at every point of the multidimensional image.

2.1 Estimation of Local Structure

The *structure tensor* allows estimation of the local structure in a multidimensional image. The structure tensor of a 3D image I is a symmetric positive semi-definite matrix given by:

$$J(I) = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{bmatrix} \quad (1)$$

where $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_z = \frac{\partial I}{\partial z}$ are the derivatives of the image with respect to x , y

and z , respectively. The components of \mathbf{J} are usually averaged with an Gaussian convolution kernel in order to represent the local structure at a higher scale.

The eigen-analysis of the structure tensor allows determination of the local structural features in the image [3]:

$$J(I) = [v_1 v_2 v_3] \cdot \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \cdot [v_1 v_2 v_3]^T \quad (2)$$

The orthogonal eigenvectors v_1 , v_2 , v_3 provide the preferred local orientations, and the corresponding eigenvalues μ_1 , μ_2 , μ_3 (assume $\mu_1 \geq \mu_2 \geq \mu_3$) provide the average contrast along them. The first eigenvector v_1 represents the direction of the maximum variance, whereas v_3 points to the direction with the minimum variance. Based on the relative values of μ_i , basic local structures can be characterized (Fig.1):

- Line-like structures have a preferred direction (v_3) exhibiting a minimum variation whose eigenvalue is much lower than the other two, i.e. $\mu_1 \approx \mu_2 \gg \mu_3$. v_1 and v_2 are directions perpendicular to the line.
- Plane-like structures have two preferred directions exhibiting similar small contrast variation, whose eigenvalues are much lower than the first one, i.e. $\mu_1 \gg \mu_2 \approx \mu_3$. v_1 represents the direction perpendicular to the

- plane-like structure, whereas v_2 and v_3 define the plane that better fits the local structure.
- Isotropic structures. When the two previous conditions do not hold, then the local structure is considered isotropic or unstructured. In general, for these structures, the eigenvalues have values of similar magnitude or order, i.e. $\mu_1 \approx \mu_2 \approx \mu_3$.

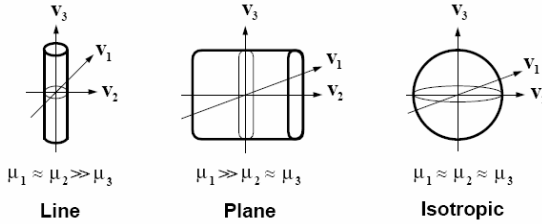


Fig. 1. Basic local structures found by eigen-analysis of the structure tensor. μ_1, μ_2, μ_3 are the eigenvalues. v_1, v_2, v_3 are the corresponding eigenvectors.

2.2 Concept of Diffusion in Image Processing

Diffusion is a physical process that equilibrates concentration differences as a function of time, without creating or destroying mass. In image processing, density values play the role of concentration. The *diffusion equation* [3] is given by:

$$I_t = \text{div}(D \cdot \nabla I) \tag{3}$$

Where $I_t = \frac{\partial I}{\partial t}$ denotes the derivative of the image I with respect to the time t , ∇I is the gradient vector, D is a square matrix called *diffusion tensor* and div is the *divergence operator*.

The diffusion tensor D allows us to tune the smoothing (both the strength and direction) across the image. D is defined as a function of the structure tensor J :

$$D = [v_1, v_2, v_3] \cdot \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \cdot [v_1, v_2, v_3]^T \tag{4}$$

where v_i denotes the eigenvectors of the structure tensor. The values of the eigenvalues λ_i define the strength of the smoothing along the direction of the corresponding eigenvector v_i . The values of λ_i rank from 0 (no smoothing) to 1 (strong smoothing). In AND, the λ_i s are normally set up independently so that the smoothing is anisotropically adapted to the local structure of the image. Consequently, AND allows smoothing along the edges so that they are not only preserved but smoothed and enhanced. AND has turned out, by far, the most effective denoising method by its capabilities for structure preservation and feature enhancement [3], [6], [7].

2.3 Common Diffusion Approaches

AND may function differently, by either filtering noise or enhancing some structural features, depending on the definition of λ_i of the diffusion tensor \mathbf{D} . Currently, the most common ways of setting up \mathbf{D} give rise to the following diffusion approaches:

EED: Edge Enhancing Diffusion

The primary effects of EED are edge preservation and enhancement [3]. Here strong smoothing is applied along the direction corresponding to the minimum change (the third eigenvector, v_3), while the strength of the smoothing along the other eigenvectors depends on the gradient: the higher the value is, the lower the smoothing strength is. The λ_i s are then set up as:

$$\lambda_1 = g(|\nabla I|) \quad \lambda_2 = g(|\nabla I|) \quad \lambda_3 = 1 \quad (5)$$

with g being a monotonically decreasing function [3]

CED: Coherence Enhancing Diffusion

CED is able to connect interrupted lines and improve flow-like structures [5] and also enhance plane-like structures [7]. The strength of the smoothing along v_2 must be tightly coupled to the plane-ness, given by $(\mu_1 - \mu_2)$, whereas the smoothing along v_3 depends on the anisotropy $(\mu_1 - \mu_2)$. So, the λ_i s are then set up as:

$$\lambda_1 \approx 0 \quad \lambda_2 = h(\mu_1 - \mu_2) \quad \lambda_3 = h(\mu_1 - \mu_3) \quad (6)$$

with h being a monotonically increasing function [5].

3 Anisotropic Nonlinear Diffusion in cryoET

3.1 Diffusion Approach

In cryoET a hybrid diffusion approach is used in order to combine the advantages of both EED and CED simultaneously [6], [9], [7]. The strategy is based on the fact that the anisotropy $(\mu_1 - \mu_3)$, reflects the local relation of structure and noise. Therefore, we use this value as a switch: CED is applied if the anisotropy is larger than a suitably chosen threshold, otherwise EED is applied. The threshold t_{ec} is derived *ad hoc* as the maximum anisotropy found in a subvolume of the image containing only noise. This approach carries out an efficient denoising which highlights the edges and connects lines and enhances flow-like and plane-like structures.

3.2 Smoothing the Background with Gaussian Filtering

In our diffusion approach, we have included a strategy to further smooth out the background. Since the interesting structural features usually have higher density levels than the background, those voxels with density values below a threshold are considered as background, and hence linear Gaussian filtering is applied. The threshold t_g is computed from the average grey level in a subvolume of the tomogram

that contains only noise, i.e. only background. As a consequence, those voxels that are considered background are significantly smoothed thanks to the Gaussian filtering.

3.3 Numerical Discretization of the Diffusion Equation

The diffusion equation, Eq.(3), can be numerically solved using finite differences. The term $I_t = \partial I / \partial t$ can be replaced by an Euler forward difference approximation. The resulting explicit scheme calculates subsequent versions of the image iteratively:

$$I^{(k+1)} = I^{(k)} + \tau \cdot \left(\begin{aligned} &\frac{\partial}{\partial x}(D_{11}I_x) + \frac{\partial}{\partial x}(D_{12}I_y) + \frac{\partial}{\partial x}(D_{13}I_z) + \frac{\partial}{\partial y}(D_{11}I_x) + \frac{\partial}{\partial y}(D_{12}I_y) + \frac{\partial}{\partial y}(D_{13}I_z) + \\ &\frac{\partial}{\partial z}(D_{11}I_x) + \frac{\partial}{\partial z}(D_{12}I_y) + \frac{\partial}{\partial z}(D_{13}I_z) \end{aligned} \right) \tag{7}$$

where τ denotes the time step size, $I^{(k)}$ denotes the image at time $t_k = k\tau$ and the D_{mn} terms represent the components of the diffusion tensor \mathbf{D} .

3.4 The Stopping Criterion: Noise Estimate Variance

AND works iteratively, yielding successive smoother versions of the image, gradually removing noise and details. The process should stop before the signal in the image is significantly affected. In this work, we use the *noise estimate variance* (NEV) stopping criterion [7]. Here, the noise that has been filtered at time t is estimated as the difference between the original noisy image, I^0 , and its current filtered version, I^t . The variance of this noise estimate increases monotonically from 0 to $\text{var}(I^0)$ during diffusion. The optimal stopping time is the time slot where $\text{var}(I^0 - I^t)$ reaches the variance of the noise subvolume in the original noisy image $\text{var}(I_N^0)$:

$$t_{stop} = \arg \min_t \left\{ \text{var}(I_N^0) - \text{var}(I^0 - I^t) \right\}$$

3.5 Scheme of Our Diffusion Approach

The outline of our AND approach is the following:

-
- (0.) **Compute NEV threshold from the subvolume containing noise.**
It computes the threshold $\text{var}(I_N^0)$ used for the stopping criterion.
 - $\text{var}(I_N^0)$ is the variance found in the noise subvolume.
 - (1.) **Compute t_{ec} and t_g from the statistics of the subvolume containing noise.**
 - (2.) **Compute the structure tensor \mathbf{J} .**
 - (3.) **Compute the diffusion tensor \mathbf{D} .**
For every voxel:
 - (3.1.) **Analysis of the local structure.**
It decides if the voxel is to be processed as EED, CED or background.
 - The voxel is considered background if its grey level is lower than t_g .
 - CED is to be applied, if the local anisotropy $(\mu_1 - \mu_3)$ is larger than t_{ec} .
 - Otherwise, EED is to be applied.
 - (3.2.) **Computation:**
 - Linear Diffusion.
If background, linear diffusion (i.e. Gaussian filtering) is applied.
 - EED: Edge Enhancing Diffusion.
If EED, the diffusion tensor \mathbf{D} is computed according to Eqs. (4), (5).
 - CED: Coherence Enhancing Diffusion.
If CED, the diffusion tensor \mathbf{D} is computed according to Eqs. (4), (6).
 - (4.) **Solve the partial differential equation of diffusion, Eqs. (3) and (7).**
 - (5.) **Iterate: go to step (1).**
-

4 Experimental Applications

The AND approach presented here is illustrated with its application to tomograms of four different biological specimens: human immunodeficiency virus (HIV) [10], vaccinia virus (VV) [11], microtubules (MTs) [12] and *Dictyostelium discoideum* cells [13]. Fig.2 shows visual results obtained from noise reduction applied to the tomograms. A single slice extracted from the 3D tomograms is shown. All the results clearly show significant noise reduction with excellent structure preservation. The structural features that are of interest from the biological point of view are smoothed and enhanced substantially thanks to the hybrid EED/CED diffusion process. In particular, the CED approach plays an essential role in the enhancement of the membranes and other linear and planar features of the specimens. The strategy to further smooth the background has a remarkable performance whereby the specimens' features are successfully highlighted over the background.

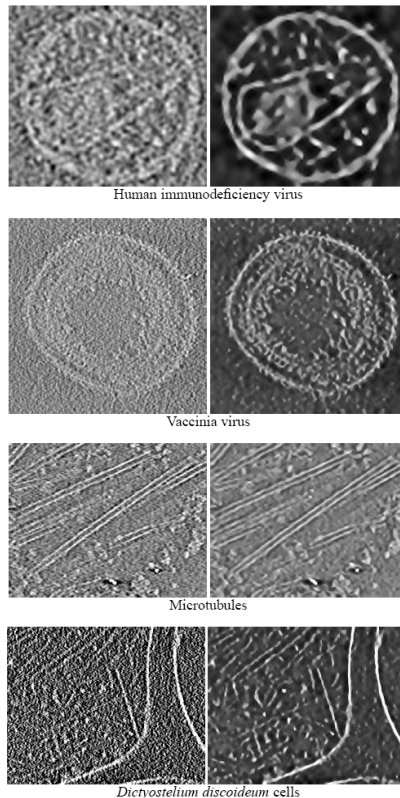


Fig. 2. Results from denoising. Left: a slice extracted from the original tomogram. Right: the same slice extracted from the denoised tomogram.

Fig.2 clearly shows the benefits of denoising for interpretation of the biological structures. In the case of HIV, there is strong enhancement of the outer membrane and

the core's surface, as well as some other bodies inside the core. In the case of VV, denoising has significantly improved planar features, allowing the interpretation of the architecture of the virus, e.g. the outer membrane and the core made up of a membrane and a palisade. With regard to the MTs, the continuity along them and their interactions are apparent. Finally, denoising has emphasized the membranes of the cell and the fibrous structures that compose the cell's cytoplasm in the *D. discoideum* tomogram.

Furthermore, AND allows visualization and interpretation of the structures in the 3D space. Also, it makes possible the application of multidimensional image analysis techniques, such as segmentation or pattern recognition, or other further post-processing techniques that facilitates quantitative interpretation [10, 11,13]. Fig.3(left) shows the 3D structure of Vaccinia virus after segmentation of its components (the core in blue, lateral body in orange, the outer membrane in yellow) [11]. Fig.3(right) shows the volume texture representation of a piece of the *D. discoideum* tomogram, where the actin filament network is now visible in the 3D space, bringing out the interaction between individual filaments and the assembly at the membrane.

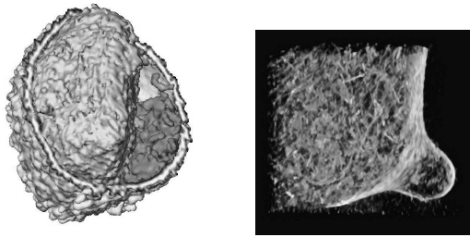


Fig. 3. Three dimensional structure of Vaccinia virus (left) and *D. discoideum* cell (right)

5 Conclusion

We have presented a method to perform structure-preserving denoising based on anisotropic nonlinear diffusion. The AND approach relies on a hybrid strategy that combines noise reduction and feature enhancement. A strategy to further smooth out the background and highlight structural features has been included. This anisotropic noise reduction method has been applied to CryoET, and the results show that it succeeds in filtering noise and emphasizing the features of interest. Therefore, this method facilitates interpretation of the structural information concealed in the noisy cryo-tomograms. The applicability of the method to other biological imaging techniques is straightforward, and it is expected to successfully denoise multidimensional images obtained from them as well.

Acknowledgments. The authors thank Dr. R.A Crowther for fruitful discussions; Drs. O. Medalia for the *D. discoideum* dataset; Drs. J.L. Carrascosa for the VV dataset.

The HIV dataset was obtained from the EBI-MSD database. Work partially supported by the MRC and grants MEC-TIN2005-00447, EU-FP6-LSHG-CT-2004-502828, JA-P06-TIC1426.

References

1. Rudin, M. *Molecular Imaging: Basic Principles and Applications in Biomedical Research*. Imperial College Press, 2005.
2. Hawkes, P.W., Spence, J.C.H, editors. *Science of Microscopy*. Springer, 2007.
3. Weickert, J. *Anisotropic Diffusion in Image Processing*. Teubner, 1998.
4. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intel.*, 12 (1990) 629–639.
5. Weickert, J.: Coherence-enhancing diffusion filtering. *Int. J. Computer Vision*, 31 (1999) 111–127.
6. Frangakis, A.S., Hegerl, R.: Noise reduction in electron tomographic reconstructions using nonlinear anisotropic diffusion. *J. Struct. Biol.*, 135 (2001) 239–250.
7. Fernandez, J.J., Li, S.: Anisotropic nonlinear filtering of cellular structures in cryo-electron tomography. *Computing in Science and Engineering*, 7(5) (2005) 54–61.
8. Sali, A., Glaeser, R., Earnest, T., Baumeister, W.: From words to literature in structural proteomics. *Nature*, 422 (2003) 216–225.
9. Fernandez, J.J., Li, S.: An improved algorithm for anisotropic nonlinear diffusion for denoising cryo-tomograms. *J. Struct. Biol.*, 144 (2003) 152–161.
10. Briggs, J.A.G., Grunewald, K., Glass, B., Forster, F., Krausslich, H.G., Fuller, S.D.: The mechanism of HIV-1 core assembly: Insights from 3D reconstructions of authentic virions. *Structure*, 14 (2006) 15–20.
11. Cyrklaff, M., Risco, C., Fernandez, J.J., Jimenez, M.V., Esteban, M., Baumeister, W., Carrascosa, J.L.: Cryo-electron tomography of vaccinia virus. *Proc. Natl. Acad. Sci. USA* 102 (2005) 2772–2777
12. Hoog, J.L., Schwartz, C., Noon, A.T., O’Toole, E.T., Mastronarde, D.N., McIntosh, J.R., Antony, C.: Organization of interphase microtubules in fission yeast analyzed by electron tomography. *Dev. Cell*, 12 (2007) 349–361.
13. Medalia, O., Weber, I., Frangakis, A.S., Nicastro, D., Gerisch, G., Baumeister, W.: Macromolecular architecture in eukaryotic cells visualized by cryoelectron tomography. *Science* 298 (2002) 1209–1213

Ensemble of Support Vector Machines to Improve the Cancer Class Prediction Based on the Gene Expression Profiles

Ángela Blanco¹, Manuel Martín-Merino¹, and Javier De Las Rivas²

¹ Universidad Pontificia de Salamanca (UPSA)C/Compañ 5, 37002, Salamanca, Spain
ablancogo@upsa.es, mmartinmac@upsa.es

² Cancer Research Center (CIC-IBMCC, CSIC/USAL) Salamanca, Spain
jrvivas@usal.es

Abstract. DNA microarrays provide rich profiles that are used in cancer prediction considering the gene expression levels across a collection of samples. Support Vector Machines (SVM), have been applied to the classification of cancer samples with encouraging results. However, they are usually based on Euclidean distances that fail to reflect accurately the sample proximities. Besides, SVM classifiers based on non-Euclidean dissimilarities fail to reduce significantly the errors. In this paper, we propose an ensemble of SVM classifiers in order to reduce the errors. The diversity among classifiers is induced considering a set of complementary dissimilarities and kernels. The experimental results suggest that that our algorithm improves classifiers based on a single dissimilarity and a combination strategy such as Bagging.

1 Introduction

DNA Microarray technology allows us to monitor the expression levels of thousands of genes simultaneously across a collection of samples. This technology has been applied particularly to the prediction of different type of cancer with encouraging results [10].

Support Vector Machines (SVM) [20] are powerful machine learning techniques that have been applied to the classification of cancer samples [8]. However, common SVM algorithms are based on Euclidean distances that fail to reflect accurately the proximities among the sample profiles [7, 13, 15]. Besides, SVM classifiers based on non-Euclidean dissimilarities are not able to reduce significantly the misclassification errors.

Combining SVM classifiers that exhibit different features can help to reduce the error and particularly the variance of the predictor [19]. To this aim, different versions of the classifier are usually built by sampling the patterns or the features [5]. However, this kind of resampling techniques reduce the size of the training set, so it may increase the bias of individual classifiers and the error of the combination [19].

In this paper, we propose an ensemble of SVM classifiers that avoids the bias introduced by resampling techniques. The diversity of classifiers is generated considering several values of the kernel parameters and a set of complementary

dissimilarities. The kernel parameters determine the generalization ability of the classifiers and can be used to induce diversity among the base predictors [20]. In the same way, there are a large variety of non-Euclidean dissimilarities that reflect different features of the data. Hence, the resulting predictors will misclassify a different set of patterns.

The classifiers induced using the previous approach take advantage of the whole sample avoiding the bias introduced by resampling techniques such as Bagging. The base classifiers have been modified to be able to work with non-Euclidean dissimilarities via the kernel trick [20]. Finally, the classifiers have been aggregated using a voting strategy [14]. The method proposed has been applied to the prediction of different type of cancer samples using the gene expression profiles with remarkable results.

This paper is organized as follows: *Section 2* introduces the dissimilarities considered to build the diversity of classifiers. *Section 3* comments how the classifiers can be extended to work from a dissimilarity matrix. *Section 4* presents our combination strategy. *Section 5* illustrates the performance of the algorithm in the challenging problem of gene expression data analysis. Finally, *section 6* gets conclusions.

2 Dissimilarities for Gene Expression Data Analysis

An important step in the design of a classifier is the choice of a proper dissimilarity that reflects the proximities among the objects. However, the choice of a good dissimilarity is not an easy task. Each measure reflects different features of the data and the classifiers induced by the dissimilarities misclassify a different set of patterns. Therefore, no dissimilarity outperforms the others.

In this section, we comment shortly the main differences among several dissimilarities proposed to evaluate the proximity between biological samples considering their gene expression profiles. For a deeper description and definitions see [7], [13], [9].

The Euclidean distance evaluates if the gene expression levels differ significantly across different samples.

An interesting alternative is the cosine dissimilarity. This measure will become small when the ratio between the gene expression levels is similar for the two samples considered. It differs significantly from the Euclidean distance when the data is not normalized by the L_2 norm.

The correlation measure evaluates if the expression level of genes change similarly in both samples. Correlation based measures tend to group together samples whose expression levels are linearly related. The correlation differs significantly from the cosine if the means of the sample profiles are not zero. This measure is more sensitive to outliers.

The Spearman rank dissimilarity is less sensitive to outliers because it computes a correlation between the ranks of the gene expression levels.

An alternative measure that helps to overcome the problem of outliers is the Kendall- τ index which is related to the Mutual Information probabilistic measure [9].

Due to the large number of genes, the sample profiles are codified in high dimensional and noisy spaces. In this case, the dissimilarities mentioned above are affected by the ‘curse of dimensionality’ [1, 16]. Hence, most of the dissimilarities become almost constant and the differences among dissimilarities are lost [12]. To avoid this problem, it is recommended to reduce the number of features before computing the dissimilarities.

3 Incorporating Non-euclidean Dissimilarities into the SVM

In this section we introduce shortly the Classical Support Vector Machines (SVM) [20]. Next, the SVM is extended to work from a dissimilarity matrix using the kernel trick.

Let $\{(x_i, y_i)\}_{i=1}^n$ be the training set codified in \mathbb{R}^d . We assume that each x_i belongs to one of the two classes labeled by $y_i \in \{-1, 1\}$. The SVM algorithm looks for the linear hyperplane $f(x) = w^T x + b$ that maximizes the margin $g = 2/|w|^2$. γ determines the generalization ability of the SVM. The slack variables ξ_i allow to consider classification errors.

The hyperplane that minimizes the prediction error is obtained solving the following optimization problem [20]:

$$\begin{aligned}
 \min_{w, \{\xi_i\}} & \quad \langle w, w \rangle + C \sum_{i=1}^n \xi_i^2 \\
 \text{s.t.} & \quad y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \quad i = 1, \dots, n \\
 & \quad \xi_i \geq 0 \quad i = 1, \dots, n
 \end{aligned}
 \tag{1}$$

where C is a regularization parameter that achieves a balance between the empirical error and the generalization ability of the classifier. The optimization problem can be solved efficiently in the dual space and the discriminant function can be written exclusively in terms of scalar products,

$$f(x) = \sum_{\alpha_i > 0} \alpha_i y_i \langle x, x_i \rangle + w_0
 \tag{2}$$

where α_i are the Lagrange multipliers in the dual optimization problem. The SVM algorithm can be easily extended to the non-linear case substituting the scalar products by a Mercer kernel [20].

Non-Euclidean dissimilarities can be incorporated into the SVM algorithm using the kernel trick [20]. Kernels transform non-linearly the data to a feature space where the Euclidean distance induces a non-Euclidean measure in input space. However, it is often impossible to know the properties of the dissimilarities induced in input space [20]. Fortunately, the kernel of dissimilarities allow us to incorporate a particular dissimilarity into the SVM algorithm [18].

Let d be a dissimilarity [18] and $R=\{p_1, \dots, p_n\}$ a subset of representatives drawn from the training set. Define the mapping $D(z,R): F \rightarrow R^n$ as:

$$D(z, R) = [d(z, p_1), d(z, p_2), \dots, d(z, p_n)] \tag{3}$$

This mapping defines a dissimilarity space where feature i is given by $d(., p_i)$. The set of representatives R determines the dimensionality of the feature space. The choice of R is equivalent to select a subset of features in the dissimilarity space. Due to the small number of training patterns in our application, we have considered the whole sample as a representative set.

The SVM optimization problem can be easily written in the dissimilarity space as follows:

$$\begin{aligned} \min_{w, \{\xi_i\}} & \langle w, w \rangle + C \sum_{i=1}^n \xi_i^2 \\ \text{s.t.} & \quad y_i f(D(x_i, R)) \geq 1 - \xi_i \quad i = 1, \dots, n \\ & \quad \xi_i \geq 0 \quad i = 1, \dots, n \end{aligned} \tag{4}$$

where f is the linear decision function implemented by the SVM.

The optimization problem can be solved efficiently in the dual space, as in the classical version:

$$\max_{\alpha} -\frac{1}{2} \alpha^T \text{diag}(y) K \text{diag}(y) \alpha + \alpha^T 1 \tag{5}$$

$$\text{s.t.} \quad \alpha^T y = 0 \tag{6}$$

$$0 \leq \alpha_i \leq C \quad i = 1, 2, \dots, n \tag{7}$$

where K is an $n \times n$ kernel matrix of dissimilarities defined as:

$$K_{ij} = \langle D(x_i, R), D(x_j, R) \rangle \tag{8}$$

$\langle .., .. \rangle$ denotes the scalar product in the feature space. Thus, the kernel matrix is written as $K=DD^T$. This matrix is positive definite and keeps the nice properties of the optimization problem in the original SVM algorithm. With this definition, the decision function can be written as:

$$f(x) = \sum_{\alpha_i > 0} \alpha_i y_i K(x_i, x_j) + w_0 \tag{9}$$

4 An Ensemble of SVM Classifiers

In this section, we present the ensemble of classifiers proposed to reduce the misclassification errors. Our method builds the diversity of classifiers considering

several dissimilarities and different non-linear kernels. Each dissimilarity introduced in section 2 reflects different features of the data and the resulting classifiers will produce different errors. In the same way, each kernel gives rise to a classifier with different degree of generalization increasing the diversity.

The base classifiers are generated considering the whole training sample. In this way, we avoid to reduce the size of the training set which may induce bias in the individual classifiers. Notice that the combination strategies are not able to reduce the bias of single classifiers [19].

Figure 1 illustrates in an intuitive way how the combination of classifiers reduces the misclassification errors. For instance bold patterns are assigned to the wrong class by C_3 but are correctly classified by C_2 and C_1 . Therefore using a voting strategy the bold patterns will be assigned to the right class.

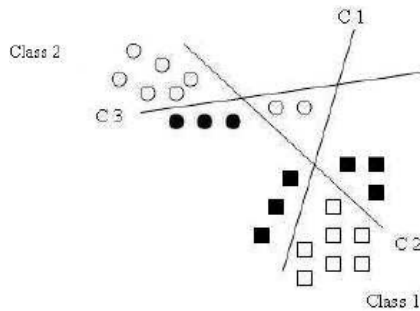


Fig. 1. Aggregation of classifiers using a voting strategy. Bold patterns are misclassified by C_3 but not by the combination.

Our combination algorithm proceeds as follows: First, the set of complementary dissimilarities introduced in section 2 are computed. Each dissimilarity is incorporated in the SVM algorithm through the kernel of dissimilarities. Then the kernels corresponding to different parameters and dissimilarities are computed. Next, the SVM is optimized in the usual way. The ensemble of classifiers is aggregated by a standard voting strategy [14].

A related technique to combine classifiers is the Bagging [5, 3]. This method generates a diversity of classifiers considering several bootstrap samples as training sets. Next, the classifiers are aggregated using a voting strategy. Nevertheless there are three important differences between Bagging and the method proposed in this section:

1st) Our method generates the diversity of classifiers by considering the whole sample. Bagging trains each classifier using around 63% of the training set. In our application, the size of the training set is very small and neglecting part of the patterns may increase the bias of each classifier. It has been suggested in the literature that Bagging does not help to reduce the bias [19] and so, the aggregation of classifiers will hardly reduce the misclassification error.

2st) The method here proposed is able to work directly with a dissimilarity matrix.

3st) The combination of several dissimilarities avoids the problem of choosing a particular dissimilarity for the application we are dealing with. This is a difficult and time consuming task.

5 Experimental Results

In this section, the ensemble of classifiers proposed is applied to the identification of several cancer human samples using microarray gene expression data.

Three benchmark gene expression datasets have been considered. The first one consisted of 72 bone marrow samples (47 ALL and 25 AML) obtained from acute leukemia patients at the time of diagnosis [10]. The RNA from marrow mononuclear cells was hybridized to high-density oligonucleotide microarrays produced by Affymetrix and containing 6817 genes. The second dataset consisted of 49 samples from breast tumors [21]: 25 classified as positive to estrogen receptors (ER+) and 24 negative to estrogen receptors (ER-). Those positive to estrogen receptors require a different treatment. The RNA of breast cancer cells were hybridized to high-density oligonucleotide microarrays produced by Affymetrix and containing 7129 genes. Finally the third dataset consists of 40 tumor and 22 normal colon samples, analyzed with an Affymetrix oligonucleotide array complementary to more than 6,500 human genes. The number of genes provided by the authors in the original dataset was 2000 [2].

Due to the large number of genes, samples are codified in a high dimensional and noisy space. Therefore, the dissimilarities are affected by the 'curse of dimensionality' and the correlation among them becomes large [16]. To avoid this problem and to increase the diversity among dissimilarities, we have reduced aggressively the number of genes using the standard F-statistic [9]. The percentile of genes considered was 14%.

The dissimilarities have been computed without normalizing the variables because as we have mentioned in section 2 this operation may increase the correlation among them.

The algorithm chosen to train the Support Vector Machines is C-SVM. The C regularization parameter has been set up by ten fold-crossvalidation [17, 4].

The algorithms have been evaluated considering the global errors and the false negative errors. Both have been estimated by ten-fold cross-validation that gives good experimental results for the problem at hand [17].

Table 1 compares the combination algorithm proposed with the best single classifier and with Bagging. We report the following conclusions:

- The combination strategy proposed improves significantly the misclassification errors of the best single classifier. In particular, the ensemble of classifiers improves the SVM algorithms based on a single dissimilarity for the three problems considered.
- The ensemble of classifiers proposed improves significantly the Bagging algorithm particularly for Breast Cancer. This result supports the idea that our algorithm performs better than the resampling techniques when the sample size is small.

- The dissimilarity that minimizes the error depends strongly on the particular dataset considered. Hence, the choice of a proper dissimilarity is not an easy task for human experts.

Table 1. Empirical results for the combination of SVM classifiers. The best SVM classifier for several dissimilarities and Bagging have been considered as reference.

Technique	Datasets	Error %	F. Neg. %
SVM (correlation)	Golub	1.94%	1.38%
SVM (Tau)	Breast	5%	2%
SVM (correlation)	Colon	11%	5%
Bagging (SVM)	Golub	8.33%	6.94%
	Breast	6.12%	2.04%
	Colon	12.9%	4.83%
Combination	Golub	1.38%	1.38%
	Breast	2%	2%
	Colon	10%	4%

6 Conclusions

In this paper, we propose an ensemble of classifiers based on a diversity of dissimilarities and kernels. Our approach aims to reduce the errors of classical Support Vector Machines. The algorithm has been applied to the classification of complex cancer human samples.

The experimental results suggest that the method proposed improves the misclassification error of classifiers based on a single dissimilarity. We also report that our method compares favorably with a combination algorithm such as Bagging.

References

1. Aggarwal, C. C.: Re-designing Distance Functions and Distance-Based Applications for High Dimensional Applications, in Proc. of the ACM International Conference on Management of Data and Symposium on Principles of Database Systems (SIGMOD-PODS), vol. 1, March 2001, pp. 13-18.
2. Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. Proc. Nat'l Acad Sci USA, 96:6745–6750, 1999.
3. Bauer, E., Kohavi, R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants, Machine Learning, vol. 36, pp. 105-139, 1999.
4. Braga-Neto, U., Dougherty, E.: Is Cross-Validation Valid for Small-Sample Microarray Classification? Bioinformatics, vol. 20, no. 3, pp. 374-380, 2004.
5. Breiman, L.: Bagging predictors, Machine Learning, vol. 24, pp. 123-140, 1996.
6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge: Cambridge University Press, 2000.
7. Drăghici, S.: Data Analysis Tools for DNA Microarrays. New York: Chapman & Hall/CRC Press, 2003.

8. Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., Haussler, D.: Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data, *Bioinformatics*, vol. 16, no. 10, pp. 906-914, 2000.
9. Gentleman, R., Carey, V., Huber, W., Irizarry, R., Dudoit, S.: *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Berlin: Springer Verlag, 2006.
10. Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E.: Molecular classification of cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, *Science*, vol. 286, no. 15, pp. 531-537, 1999.
11. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene Selection for Cancer Classification Using Support Vector Machines, *Machine Learning*, vol. 46, pp. 389-422, 2002.
12. Hinneburg C. C. A., Keim, D. A.: What is the Nearest Neighbor in High Dimensional Spaces? In *Proc. of the International Conference on Database Theory (ICDT)*. Cairo, Egypt: Morgan Kaufmann, September 2000, pp. 506-515.
13. Jiang, D., Tang, C. Zhang, A.: Cluster Analysis for Gene Expression Data: A survey, *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, November 2004.
14. Kuncheva, L. I.: *Combining Pattern Classifiers*. John Wiley, New Jersey, 2004.
15. Martín-Merino, M., Muñoz, A.: Self Organizing Map and Sammon Mapping for Asymmetric Proximities, *Neurocomputing*, vol. 63, pp. 171-192, 2005.
16. Martín-Merino, M., Muñoz, A.: A New Sammon Algorithm for Sparse Data Visualization, In *International Conference on Pattern Recognition (ICPR)*, vol. 1. Cambridge (UK): IEEE Press, August 2004, pp. 477-481.
17. Molinaro, A., Simon, R. Pfeiffer, R.: Prediction Error Estimation: a Comparison of Resampling Methods, *Bioinformatics*, vol. 21, no. 15, pp. 3301-3307, 2005.
18. Pekalska, E., Paclik, P., Duin, R.: A Generalized Kernel Approach to Dissimilarity-Based Classification," *Journal of Machine Learning Research*, vol. 2, pp. 175-211, 2001.
19. Valentini, G., Dietterich, T.: Bias-Variance Analysis of Support Vector Machines for the Development of Svm-Based Ensemble Methods, *Journal of Machine Learning Research*, vol. 5, pp. 725-775, 2004.
20. Vapnik, V.: *Statistical Learning Theory*. New York: John Wiley & Sons, 1998.
21. West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J., Marks, J., Nevins, J.: Predicting the Clinical Status of Human Breast Cancer by Using Gene Expression Profiles, *PNAS*, vol. 98, no. 20, September 2001.

NATPRO-C13 – An Interactive Tool for the Structural Elucidation of Natural Compounds

Roberto Theron², Esther del Olmo¹, David Díaz², Miguel Vaquero², José Francisco Adserias³, and José Luis López-Pérez¹

¹ Departamento de Química Farmacéutica-Facultad de Farmacia

² Departamento de Informática y Automática-Facultad de Ciencias

³ Fundación General. Universidad de Salamanca, Spain

theron@usal.es, lopez@usal.es

Abstract. This paper describes the characteristics and the improvements of the free web-based spectral database NATPRO-C13, containing ¹³C NMR spectra data from more than 5.000 natural compounds and related derivatives. It provides tools that facilitate the structural identification of natural compounds even before their purification. This database allows for searches by chemical structure, substructure, name, family compounds, and by spectral features i.e. chemical shifts and multiplicities. These capabilities are used together with visual interactive tools, which enable the structural elucidation of known and unknown compounds by comparison of their ¹³C NMR data.

Keywords: structural elucidation, ¹³C NMR spectral database, natural compounds, chemoinformatics, SMILES.

1 Introduction

Natural products from microbial, plant, marine, or even mammalian sources have traditionally been a major drug source and continue to play a significant role in today's drug discovery environments [1], [2]. In fact, in some therapeutic areas, for example, oncology, the majority of currently available drugs are derived from natural products.

In the natural products research, tedious purifications of extracts are needed in order to isolate their constituents. These procedures are often performed with the main purpose of molecular structure identification or elucidation. The molecular structure of a substance is its main characteristic because it determines all of its properties: chemical, physical, biological, and pharmacological.

When a researcher in natural products isolates and purifies a compound he needs to know its skeleton, total structure and if it has been previously described.

If the compound under investigation has been described before, it can be identified by a direct comparison with an original sample or by property comparison of those described for this compound. If this is not the case, it will be necessary to carry out a structure elucidation. The difficulty of identifying the structure of a compound depends on several factors, such as the substance's origin and its complexity.

Since, in the case of synthetic or semi-synthetic procedures, the researcher knows beforehand the starting structures and their resulting products, he only needs to confirm that the desired target has been reached. Nevertheless, the process can be much more complicated if it is necessary to identify a natural compound, isolated from an extract of species that either grow only in unique geographical areas or belong to species without previous phytochemical studies. In this case, the difficulty will mainly reside in the complexity of the structure.

Yet if the structures of natural extract constituents can be known in advance, the isolation efforts could be focused on truly novel and interesting components, avoiding re-isolation of known or trivial constituents and their increasing productivity [3].

^{13}C NMR spectroscopy is the most powerful tool [4] as regards identification and elucidation of natural compounds. This is largely due to the well-known and exquisite dependence of the ^{13}C chemical shift of each carbon atom on its local chemical environment and its number of attached protons. Furthermore, the highly resolved spectra, provided by a large chemical shift range and narrow peak width, could be easily converted into highly reduced numerical lists of chemical shift positions with minimal loss of peak intensity and width information.

The analysis of spectral data for the determination of unknown compound structure remains a usual and laborious task in chemical practice. The results of these spectroscopic experiments need to be compared with those of the previously described compounds.

This methodology provides highly interesting challenges for chemoinformatic practitioners. If a database of natural products and their NMR spectral data were available, searching databases would allow for quick identifications by comparison with previously registered compounds. They would also provide insight into the structural elucidation of unknown compounds.

Since the advent of computers many efforts have been directed toward facilitating the solution to structural identification [5]. Libraries of such spectral lists of data are common for synthetic organic compounds and are an invaluable tool for the confirmation of the identity of known compounds [6]. However, the methods for structure elucidation of compounds outside a database have not been exhaustively studied. In the field of natural products, where hundreds of thousands of compounds have been reported in the literature, most compounds are absent from commercially available spectral libraries. A library about Marine Natural Products is available [7]. Despite of the help of databases, the comparison of such a large number of compounds remains still a very difficult task that can be more easily performed with the aid of both numerical algorithms and interactive visual tools.

2 What Is NATPROC-13?

NATPROC-13 is a web-based application that holds a database for Natural Products and their ^{13}C Nuclear Magnetic Resonance Spectra. It can be accessed and used without restrictions by a standard browser at <http://c13.usal.es>. It allows for the search of spectral data and substructures. The implemented search algorithms can scan the database for trivial and semi-systematic name, formula, structure, substructures, and

chemical shifts. It also generates a Hit List of best matches. These capabilities have been enhanced with advanced interactive visualization tools in order to provide means of data manipulation.

This database is constantly growing and it can store hundreds of thousands of compounds. This tool is designed for researchers who are willing to enter the information of their own compounds and to share information over the Net. We have also developed scripts to automatically parse input data, run different tests and populate the database.

2.1 Implementation

In order to develop NATPROC-13, MySQL (<http://www.mysql.com>) has been chosen given its fast performance, high reliability and ease of use. MySQL is an open-source relational database manager, based on SQL (Structured Query Language). We use the open-source Apache Tomcat Web server and JSP to create Web pages that show contents which are dynamically generated. By means of proper JSP programming we bring about the communication between applets and the database. As for the interactive visualization tools Java applets have also been integrated in this application.

The database contains natural compounds, mainly terpenoids. The largest number of heavy atoms of a compound in the database is 99, and its molecular formula is $C_{66}H_{106}O_{33}$. Structures and spectral data collected in the database are mainly compiled from papers in the following research journals: Journal of Natural Products, Phytochemistry, Planta Medica, Chemical & Pharmaceutical Bulletin, Chemistry of Natural Compounds, Helvetica Chimica Acta, Magnetic Resonance in Chemistry.

2.2 Database Schema and Data Format

The numbering system of “Dictionary of Natural Products” has been applied to every family skeleton. The homogeneity in the numbering within the same family compounds enables the comparison of spectral data for a variety of related structures.

The basic database scheme is relationally organized and the molecular structures are defined and stored in the database with SMILES (Simplified Molecular Input Line Entry Specification) [8], [9] code. This format of structural specification, that uses one line notation, is designed to share chemical structure information over the Internet [5]. The substructural searches are performed by SMARTS code (SMILES ARbitrary Target Specification), a variation of the SMILES code. While SMILES defines the molecules in the form of alphanumeric chains, that enable an easy manipulation, SMARTS is a more complex code. It uses a boolean operator that allows choosing all-purpose atoms, groups of alternative atoms, donor and acceptor groups of hydrogen bonds or lipophilic atoms. The spectral ^{13}C NMR data in the form of a numerical list of chemical shift and their multiplicity is always associated with every compound structure.

An applet, JME [10], is used to convert these notations into a graph that represents a substructure that will be interpreted by organic chemists without specific training in this area.

3 Searches

NATPROC-13 allows for flexible searches by chemical substructure of structures as well as for spectral features, chemical shifts and multiplicities. Searches for trivial and semi-systematic names, molecular forms, families, types and groups of compound according to the IUPAC classification and other parameters are also provided for.

3.1 Search by Category/Name/Molecular Formula/Names

Pull-down lists of defined categories (Family, Type, Group) of compounds present in the database are available (see Figure 1). After selecting a specific Family, a pull-down list with the Type compound available in this database for this Family appears. Once a Type has been selected, another pull-down list of Groups available for this Type in the database appears. In this way a selective selection of Family, Type or Group is possible.

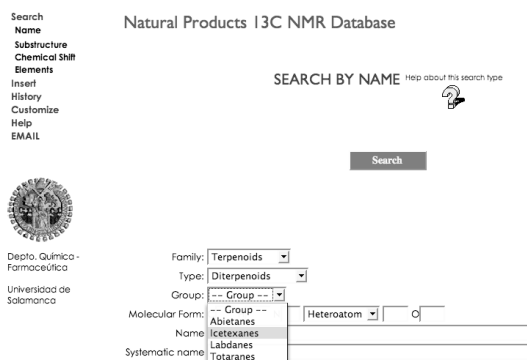


Fig. 1. Search by Category. Pull-down lists of categories present in the database

A partial or total molecular formula can be defined in the query by introducing independently the number of C, H, N, O or other heteroatoms. This search type proves very useful when the exact molecular formula cannot be exactly determined.

3.2 Substructure Search

From the spectroscopic analysis of several two-dimensional experiments of NMR, like COSY (Correlated Spectroscopy), HMQC (Heteronuclear Multiple Quantum Coherence), HMBC (Heteronuclear Multiple Bond Coherence), we can infer the presence of different structural fragments. These fragments can be drawn with the editor in order to find all the compounds that simultaneously fulfill these requirements [11].

In order to search molecular structures in the database, it is first necessary to interactively draw a substructure into the JME¹ applet window. Buttons with

¹ JME, version 1.2. The authors wish to thank the courtesy of Dr. Peter Ertl for consenting to the use in this non-profit database.

functional defined groups are available. We can also specify the stereochemistry according to the IUPAC notations by using solid (up) and dashed (down) wedged bonds. At the same time the search could be restricted to one specific Family, Type or Group of Natural Compounds by selecting one of these categories in specific pull-down lists. Once the desired substructure has been drawn the search can be performed.

3.3 Search by Chemical Shift (Spectrum Search)

This database enables a search for a given set of chemical shifts. The experimentally obtained chemical shift must be entered in "Input Fields" (one resonance per line). The multiplicity for each chemical shift is always required and this constitutes a useful search restriction. This means that only chemical shifts of the signals that adjust to the specified multiplicity will be considered during the search.

NATPROC-13 offers several alternatives of the chemical shift search process. The search can be undertaken for one specific position in the molecule. Furthermore, the user can formulate the enquiry with the required number of carbons, by one carbon or more, up to the totality of the carbons of the compound. There is a default established deviation (± 1 ppm) for all chemical shifts, but the user can specify a particular deviation for every carbon. It is convenient to repeat the search with different deviations and to select the search that provides the best results. If the deviation is too small, it may occur that an interesting compound will not be selected. In this way the researcher will obtain a reasonable and manageable number of compounds.

Even a search based only on the most significant carbons of the studied compound ^{13}C NMR spectrum will lead to the identification of the family they belong to.

If the skeleton of the studied substance is already known, and if some distinctive chemical shifts of the most important signals are also available, a search by shifts in each particular position of the molecule can be carried out. Therefore the researcher will only obtain the compounds of the family whose shifts at particular positions match those of the problem compound.

The iterative search, together with the search by proximity, are probably the most characteristic and idiosyncratic search of this application. The user can include in his/her search from one to the totality of chemical shifts. This tool will initially carry out a search of all the entered chemical shifts. If it does not find any compound that does not fulfill the full requirements, it will undertake a new iterative search by all the shifts except one. Thus, the tool will perform all the possible combinations until it finds a compound that fulfills some of the requirements.

After the input spectrum list has been entered and the search algorithm has been chosen, the application will execute the search.

3.4 Search by Proximity

This kind of search is visually directed. (see Figure 2) This feature provides an intuitive and flexible way to perform exploratory searches. The expert can insert a set of chemical shifts which correspond to neighbouring carbons (heteroatoms in between carbons can be disregarded). The underlying search algorithm will try to maximize

the number of hits in the database that matches the given chemical shifts set. The results obtained in this search can be further explored with the visual analytical tool included in NATPROC-13.

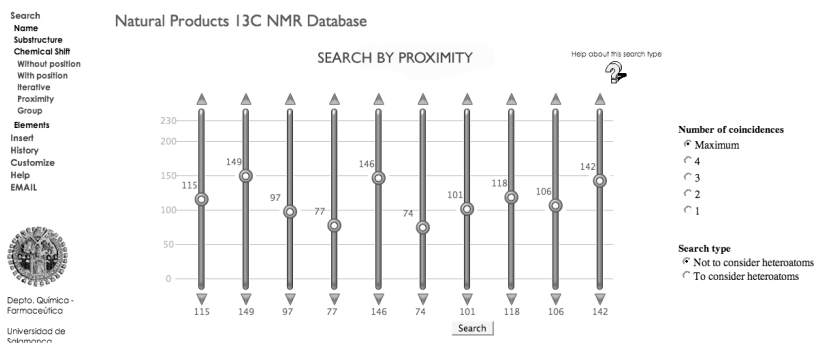


Fig. 2. Search by proximity. The chemical shift can be selected visually.

3.5 Interactive Visual Analytical Tool

As stated above, NATPROC-13 features a built-in visual analytical tool (see Figure 3). It is a highly interactive interface integrated by four linked views (left to right, top to bottom): a) ^{13}C NMR Spectrum, b) structure, c) Parallel coordinates plot and d) taxonomic information.

The main advantage of this approach is that the expert can deal with a great number of compounds that have matched a particular search. Thanks to interaction, the user can explore this result set, focusing on particular details of a given compound, (name-family-type-group, structure, spectrum) while maintaining the context, i.e. the characteristics of the rest of the compounds in the result set. Parallel

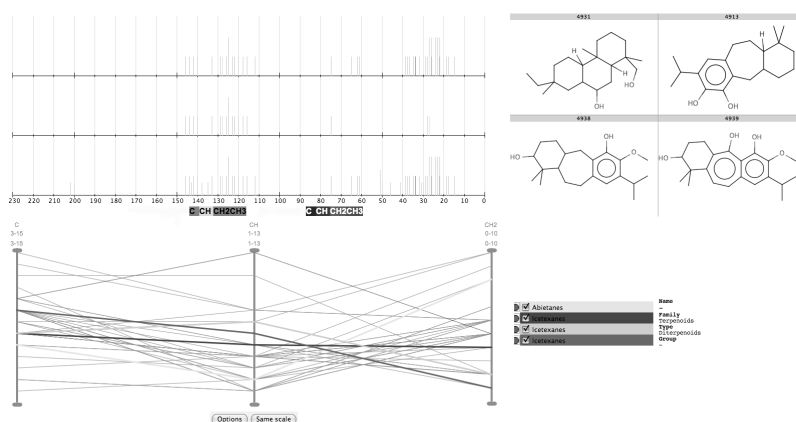


Fig. 3. Visual interactive exploration of results

coordinates [12] provide a way of representing any number of dimensions in the 2D screen space. Each compound is drawn as a polyline passing through parallel axis, which represent the number of particular elements or groups. Thus, it is possible to discover patterns, i.e. a number of polylines (compounds) with similar shapes.

A similar, although visually different, approach is taken with the representation of the spectra. Initially, all the compound spectra are shown as overlapped. Thus a global pattern can be discovered in the result set.


The user can interact with any of the four views and, as a result, the other three views will change accordingly. For instance, the expert may select any number of polylines and the corresponding spectra will be shown as overlapped, and their structures are shown in order to facilitate their comparison.

Further inspection can be achieved by filtering the result set according to different criteria (e.g. a range in the number of occurrences of an element or a particular area in the spectrum). The filtered data is visually maintained in the background in order to keep always the context of the exploration.

All these features are a great means that foster the discovery of knowledge and provide insight into the vast number of compounds in the database.

Natural Products ¹³C NMR Database

Search
Name
Substructure
Chemical Shift
Without position
With position
Iterative
Proximity
Group
Elements
Insert
History
Customise
Help
EMAIL



Depto. Química -
Farmacéutica
Universidad de
Salamanca

History



Search Num.	Search Code	Results	Criteria	Show Results	Delete
☐ 1	1	19	Family=Terpenoids, Type=Diterpenoids, Group=Icetaxanes	View	☐
☐ 2	2	15		View	☐
☐ 3	3	4	Carbon type=CH, Chemical shift=121, Tolerance=6, Carbon type=CH, Chemical shift=74, Tolerance=6, Carbon type=C, Chemical shift=34, Tolerance=6	View	<input checked="" type="checkbox"/>
☐ 4	(2 and 3)	3	 and Carbon type=CH, Chemical shift=121, Tolerance=6, Carbon type=CH, Chemical shift=74, Tolerance=6, Carbon type=C, Chemical shift=34, Tolerance=6	View	☐
Search Num.	Search Code	Results	Criteria	Show Results	Delete
Combine	and or				Delete

Fig. 4. History of searches. Combination of searches 2 and 3.

4 The Search History

The search history window can be accessed from the main menu “History” (see Figure 4). It will keep a history of all searches performed in the current session. Every single search is defined in one row with a code (Search code), the number of entries (Results) and the criteria used during the search (Criteria). The results of a single search shown in the history window can be displayed by pushing the corresponding

button "View" in the column "Show Results" and it can be deleted by clicking on the button "Delete" after the activation of the corresponding box in the column "Delete". If multiple searches are selected, one can choose between "and" and "or". The "and" option means that all datasets fulfill all conditions at the same time; the "or" option means that all datasets fulfill at least one condition. "And" should be used if one wants to combine different categories, e.g., a substructure search and a Family of compounds. "Or" will be used in order to combine several possibilities of the same category, e.g., several Groups of a Family. One can also choose the "not" option. This will permit us to select everything that does not fulfill the "not" conditions, that is, the "rest" of the database.

This enables a combined and simultaneous search by substructure and by chemical shifts, a feature that undoubtedly enhances the search capacity and increases the possibilities of finding compounds related with the problem substance.

4.1 Browsing and Displaying the Results

The matching records retrieved resulting from a search can be displayed in the "Results" pane in the form of molecular structure. These results can be browsed page by page. The chemical shifts of the matching records can be viewed in tables or in the compound structures by clicking the buttons at the top right " δ (ppm) in tables" or " δ (ppm) in structures", respectively. In order to view the details of a particular record, click on the "Properties". To see the Spectrum graphically, click on "Spectrum".

The user can selectively eliminate any number of records. In order to adapt the information search results to the characteristics of the screen, that is, size and resolution, the user can interactively define the size of the structure window and the number of structures per page. He can also define the number of records per page.

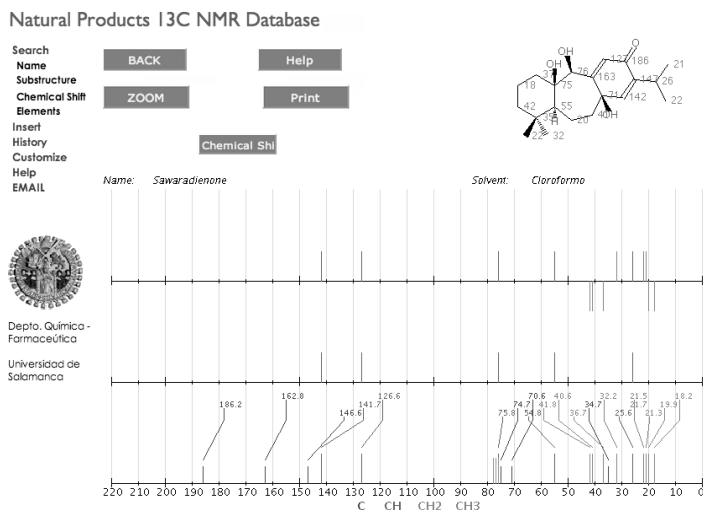


Fig. 5. ^{13}C NMR Spectrum. Multiplicities of the carbons are codified by colors.

Below of each structure window there are two labels: Properties and Spectrum. A “Properties” window can be shown, which contains the record of this particular compound. Similarly, a Spectrum page (see figure 5) which contains graphic ^{13}C NMR spectra can be shown. This representation is similar to the one obtained experimentally showing the decoupled proton (broad band) and the DEPTs (Distortionless Enhancement by Polarization Transfer). Multiplicities of the carbons are codified by colors: quaternary carbons in blue, CH in red, CH_2 in green and CH_3 in magenta. Besides, signals corresponding to the deuterated solvent used in the experiment appear in gray color. Also the numerical shift of every signal can be printed in the screen. This graphical representation can be zoomed so as to enable a better appreciation of the populated area of the spectrum with signals whose shifts are very similar. Also, spectrum measurement conditions as the deuterated solvent are available.

5 Conclusions and Further Work

The development of this database, provides an excellent tool to deal with complex chemical problems such as structure elucidation, which necessitates the joint efforts of information science and chemistry experts.

At present we are increasing the number of stored compounds belonging to different families. We are improving the tool to generate new entries and to speed up the process. We are also developing new search methods. We attempt to improve information visualization techniques that give a deeper insight into the analytical processes. We plan to include supervised and unsupervised machine learning methods that will lead to interesting predictions for assignment of the NMR spectral data of new compounds.

Acknowledgements. Financial support came from the Ministerio de Educación y Ciencia, project TIN2006-06313 and the Junta de Castilla y León, projects SAO30A06 and US21/06.

References

1. Grabley, S. and Thiericke, R.: Bioactive agents from natural sources: trends in discovery and application. *Adv. Biochem. Eng. Biotechnol.* 64 (1999) 101-54.
2. Harvey, A.: Strategies for discovering drugs from previously unexplored natural products. *Drug. Discov. Today* 5, (2000) 294-300.
3. Clarkson, C., Stärk, D., Hansen, S.H., Smith, P.J., Jaroszewski, J.W.: Discovering New Natural Products Directly from Crude Extracts by hplc-spe-nmr: Chinane diterpenes in *harpagophytum procumbens*. *J. Nat. Prod.* 69 (2006) 527-30.
4. Breitmaier, E.; Woelter, W. *Carbon-13 NMR Spectroscopy. High-Resolution Methods and Applications in Organic Chemistry*, Third ed.; VCH Publishers: Weinheim, Germany, 1987.
5. Gasteiger, J.: Chemoinformatics: a New Field with a Long Tradition. *Anal. Bioanal. Chem.* 384 (2006) 57-64.

6. Robien, W.: NMR Data Correlation with Chemical Structure. In v. R. Schleyer, P., Allinger, N.L., Clark, T., Gasteiger, J., Kollman, P.A., III, H.F.S., Schreiner, P.R., eds.: *Encyclopedia of Computational Chemistry*. Volume 3. John Wiley & Sons, Limited, Chichester, England (1998) 1845–57.
7. Lei, J., Zhou, J.: *J. Chem. Inf. Comput. Sci.* 42 (2002) 742-8.
8. Weininger, D.: Smiles, a Chemical Language and Information System. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* 28 (1988) 31–6
9. Weininger, D., Weininger, A., Weininger, J.L.: Smiles. 2. Algorithm for Generation of Unique Smiles Notation. *J. Chem. Inf. Comput. Sci.* 29 (1989) 97–101
10. Ertl, P., Jacob, O.: WWW-based Chemical Information System. *Theochem*, 419 (1997) 113-120
11. Kochev, N., Monev, V., Bangov, I.: Searching Chemical Structures. In: *Chemoinformatics: A textbook*. Wiley-VCH (2003) 291–318.
12. Inselberg, A.: The plane with parallel coordinates. *The Visual Computer* 1 (1985), 69–91

Application of Chemoinformatic Tools for the Analysis of Virtual Screening Studies of Tubulin Inhibitors

Rafael Peláez, José Luis López, and Manuel Medarde

Departamento de Química Farmacéutica-Facultad de Farmacia,
Campus Unamuno. Universidad de Salamanca, Spain
pelaez@usal.es

Abstract. Virtual screening (VS) experiments were applied to rank more than 700000 candidate lead-like virtual molecules in order of likelihood of binding to the colchicine site of tubulin, which is an important antitumor target. The best ranked compounds were clustered and classified by means of “ad hoc” semiautomatic chemoinformatic tools. The results obtained in this way were compared with those achieved by visual inspection protocols and the best were selected for synthesis and screening stages.

Keywords: drug design, virtual screening, tubulin.

1 Introduction

The development of new drugs is a very complex and demanding interdisciplinary process. Despite the huge efforts devoted to this enterprise and the continuous methodological advances, the number of successful events is greatly surpassed by the number of failures. The completion of the human genome project has expanded the number of possible drug targets, but it has already become evident that a deeper understanding of these targets and more accurate tools and techniques for the discovery and improvement of new drug candidates are needed. [1]

A variety of computational approaches can be applied at the different stages during the process of designing and developing drugs. The appropriate choice will depend on the information available, the oddities of the problem and the resources. An important tool of the drug design armamentarium is virtual screening (VS), which attempts to rank the candidate molecules in descending order of likelihood of biological activity, hence reducing the number of compounds for experimental high-throughput screening (HTS) assays which are very costly. [2] When the structure of the target and the binding site are known, one choice of virtual screening strategies is docking. This involves a positioning of the compounds in the target site (posing) and a ranking of the resulting complexes (scoring). [3] The docking approach attempts to simulate several complex phenomena with a minimal time investment, thus resulting in inconsistent performance. Several approaches attempt to alleviate the inherent limitations of the method, with different degrees of success. A combined use of different scoring functions tries to compensate for their simplified character, explicit incorporation of ligand flexibility, simulation of protein flexibility, etc. [4]. The best

ranked compounds are clustered and classified and often visually inspected by experienced chemists in order to confirm their goodness, before the final selection for the screening stages.

Microtubules are the basic components of cell structure, which take part in a number of pivotal cell functions such as division, motility, intracellular transport, and shape maintenance. Microtubules are polymers made of α - and β -tubulin heterodimers, which show a characteristic, high dynamic behaviour. Drugs interfering with the dynamics of microtubules are called antimicrotubule agents and are in clinical use as antitumour, antiprotozoal or antifungal chemotherapeutics. They validate the role of tubulin as a drug target. [5] The structure of the tubulin dimer complexed with several ligands (Figure 1) has been recently determined [6] and it has opened the door to the application of structure based drug design methodologies.

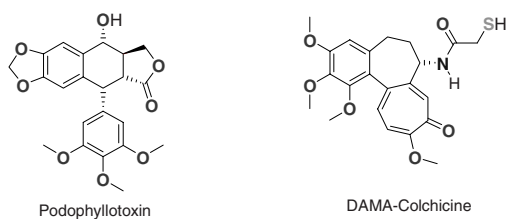


Fig. 1. Estructures of podophyllotoxin and DAMA-Colchicine

Three components are needed for virtual screening: the target structure with a defined binding site, the virtual library of hypothetical ligands and the docking-scoring programs.

2 The Target

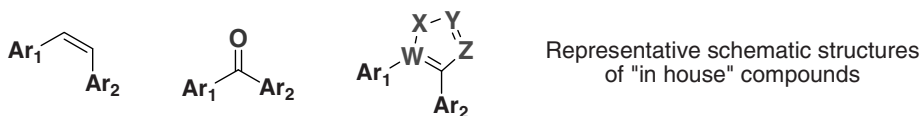
We decided to work with the so called colchicine site of tubulin. Drugs binding at the colchicine site are structurally simpler and more amenable to drug design and synthesis than those binding to other possible sites. [5] The resolutions of the experimentally determined complexes of tubulin with DAMA-colchicine or podophyllotoxin (Figure 1) are low for docking purposes, requiring some further manipulation [6]. The coordinates of the complexes were retrieved from the Brookhaven Protein Data Bank (1sa0 and 1sa1). [7] After an initial restrained minimization aimed to reduce bad contacts, the complexes were relaxed by molecular dynamics simulations using the Discover® program (Figure 2) [8]. Afterwards, the relaxed complexes overlapped so as to minimize the RMS of the binding site residues of the two proteins. The two aligned complexes were placed side by side by translating the podophyllotoxin complex 30Å along the X axis. The two complexes were subsequently and simultaneously used in docking experiments, a strategy which attempts to somewhat compensate for the rigid modelling of the protein target (cross-docking). [9], [10]



Fig. 2. Refined binding site of tubulin occupied by colchicine

3 The Virtual Library

The dataset of hypothetical ligands was constructed by combination of virtual compounds from three different sources: the ZINC free electronic database of commercially-available compounds for virtual screening [11], the ACD molecules employed by A. N. Jain in Surfex docking validation, [12] and a collection of compounds synthesized in our laboratory and tested against tubulin. From the Zinc (a electronic database which contains over 4.6 million compounds in ready-to-dock, in 3D formats) we selected a subset which has appropriate characteristics for a lead candidate, that is, molecular weight between 150 and 350 Daltons; xlogP between 4.00 and -2.00; the number of hydrogen bond donors less than or equal to three; and the number of hydrogen bond acceptors less than or equal to six. [13] Under these conditions 700.000 compounds were selected. The ACD compounds employed in Surfex docking validation were 990 randomly chosen nonreactive organic molecules, which can be considered as negative controls, without biological activity. The dataset of compounds synthesized in our laboratory is a congeneric set of more than 300 related compounds displaying different degrees of activity against the desired target. It represents a more demanding test of the virtual screening protocol, which usually does not accurately discriminate active from inactive compounds in structurally related series. The compounds were drawn in 2D and converted to 3D structures with MarvinBeans [14]. The 3D structures were energetically minimized using Macromodel [15].



Representative schematic structures of "in house" compounds

Fig. 3. Representative schematic structures of "in house" compounds

4 Docking Scoring Program

The docking-scoring task was carried out with the Surflex docking program [12]. Surflex is a fully automated flexible docking program that combines the scoring function from the Hammerhead docking system with a search engine that relies on a surface-based molecular similarity method in order to generate poses. Surflex proved a significantly better performance when compared with competing methods. The cross docking protocol was implemented by independent generation of one protocol for each site and subsequent combination of both files. For comparison purposes with the cross docking protocol, the docking of the in house library and the ACD dataset was performed separately on each site.

Surflex was run on Intel Pentium IV based desktop computers, Apple Power PC G3 computers, and SGI Octane workstations. The docking calculations were split in sets of 1000 compounds, manually generated from the global dataset. The combined output of the Surflex docking calculations was processed with in house written perl scripts so as to extract the scoring values, calculate different scoring indexes, rank the ligands and the conformations of each ligand, extract the 3D coordinates of the highest docking conformation of each ligand, and finally establish their ranking. Additional information such as the preference (or lack) of each ligand for the colchicine or podophyllotoxin sites, the importance of different combinations of scores, the performance of cross docking versus independent docking, etc. was also derived and quantified. These results were analyzed for the congeneric series of known binding affinity, and the best performing scheme (as determined by receiver operating characteristic curves) was applied to the whole dataset. The results of cross docking experiments were totally consistent with those performed independently on each protein complex.

5 Analyzing the Virtual Screening Results

The results of the docking studies were analyzed by constructing receiver operating characteristic curves (ROC) with the results obtained for the control datasets (Figure 4) [16]. The compounds were binary classified as active (value 1) or inactive (value 0) according to the following criteria: the ACD compounds were all considered inactive, and the house compounds were considered active if they present a CI_{50} lower

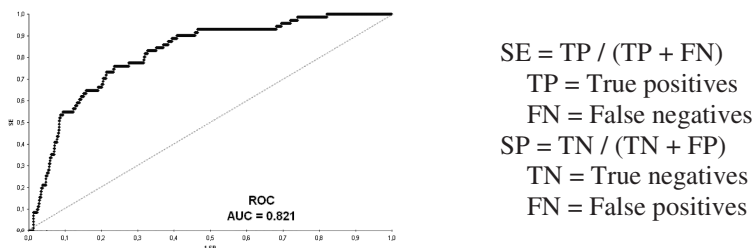


Fig. 4. ROC for the “in house” dataset and the ACD dataset. SE = selectivity. SP = specificity. The diagonal line represents random selection.

than 40 μM in tubulin polymerization assays. The resulting ROCs agree with the initial expectations. If we consider only the inactive compounds of the ACD dataset, the enrichment factor is better than in the case in which the inactive compounds of skeleton similar to the active compounds are also included.

The visual inspection of the complexes is out question, due to the huge number of 3D structures generated during the virtual screening procedure (10 conformations for each of the more than 700,000 molecules make $7 \cdot 10^6$ conformations). We therefore decided to attempt a semiautomated means of comparison of the complexes (Figure 4). We assumed that similar poses should put similar atoms in similar positions. Therefore we generated a grid that would sample the occupancy of the binding site by the poses. In order to consider both those poses occupying the podophyllotoxin and those occupying the colchicine site, we shifted the second group 30 Å along the X axis, by means of in house generated perl scripts. [17] We then generated a grid of points (we have tested point separations from 0,3 to 2 Å) encompassing the two overlapping sites. Using Insight macros, [8] we classified the non hydrogen atoms of the ligands in three groups: aliphatic carbon atoms, aromatic atoms and aliphatic hetero-atoms. For each ligand pose a bitstring was calculated: for each grid point we calculated if there was an atom of the ligand –of each of the three types separatedly- within Van der Waals contact. As a result, each pose was represented by a binary bitstring, of which three positions corresponded to each grid point (a 1 indicating occupation by the appropriate atom type and a 0 indicating vacancy). The bitstrings were used to obtain similarity half-matrices by calculating, using in house written perl scripts, the pairwise similarity (measured as similarity coefficients) of every pose in the dataset to the rest of poses. Initially, Tanimoto (alias Jaccard) coefficients were chosen for the similarity measures, since lack of occupancy (d) was considered to convey no information. [18] The Tanimoto coefficients were calculated as follows:

$$T = c/a + b + c \quad (1)$$

where

a is the number of bits with a 1 value in the first pose and 0 in the second

b is the number of bits with a 0 value in the first pose and 1 in the second

c is the number of bits with a 1 value in the first and second poses

d is the number of bits with a 0 value in the first and second poses

all the comparisons are position dependent.

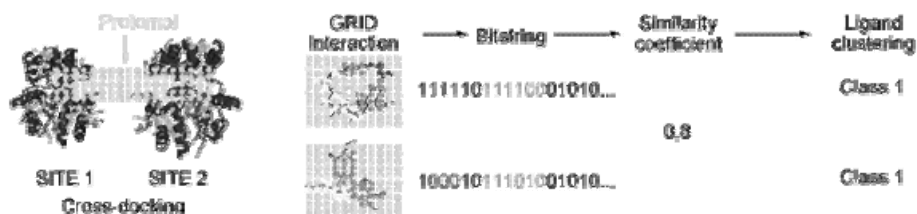


Fig. 5. Schematic representation of the scoring and clustering protocols

In order to generate hierarchical similarity trees, the half matrices thus generated were input to the XCluster software accompanying Macromodel® [15]. When only a fraction of compounds were studied (either the top scored 100-300 compounds in their best pose each, or the combination of the ACD and in house datasets of compounds in their best pose each, ...) the clustering procedure led to chemically reasonable clusterings of the compounds. However, while the number of compounds increased, more scattered results were found. The employment of different similarity coefficients with different relative weightings did not substantially improve the results.

A manual clustering of the top ranked poses of each compound of the ACD and in house datasets was performed in order to assess the best behaved results of the semiautomated procedure by means of visual methods. [19] The poses were manually classified in eight distinct clusters, with different numbers of compounds in each of them. One of the most populated clusters corresponded to a naive superposition of the trimethoxyphenyl rings (or an equivalent moiety) of the ligands with those of colchicine and podophyllotoxin. This cluster was particularly enriched in compounds of the in house dataset that presented activity in tubulin polymerization assays. Although producing chemically reasonable clusters, the semiautomated protocol did not reproduce the visually generated clusters. This discrepancy probably arises from a biased chemical viewpoint –due to unconscious application of known structure activity relationships for this class of compounds–, which is not matched by a more automatic clustering. Means for the incorporating this bias in cluster generation are now being pursued in our group.

6 Conclusions

Analyzing the top results found by the virtual screening protocol, we have identified among the Zinc dataset several compounds with structures similar to known tubulin polymerization inhibitors, as well as molecules displaying interesting new features. The combination of the results from the virtual screening protocol with the manual clustering results has yielded a chemical hindsight for the selection of several molecules as targets for chemical synthesis and assays. Some of these molecules have already been synthesized and they have shown promising inhibitory activities in tubulin polymerization assays.

Acknowledgements. We thank the MEC (Ref CTQ2004-00369/BQU), the Junta de Castilla y León (Refs. SA090A06, SA030A06 and US21/06) and the EU (Structural Funds) for the financial support.

References

1. Kubinyi, H. *Nat. Rev. Drug. Disc.* 2003, 2, 665-9. Drug Research: Myths, Hype and Reality.
2. Soichet, B. K. *Nature* 2004, 432, 862-865. Virtual Screening of Chemical Libraries.
3. Leach, A. R.; Shoichet, B. K.; Peishoff, C. E. *J. Med. Chem.* 2006, 49, 5851-5855. Prediction of Protein-Ligand Interactions. Docking and Scoring: Successes and Gaps

4. Warren, G. L.; Andrews, C. W.; Capelli, A.-M.; Clarke, B.; LaLonde, J.; Lambert, M. H.; Lindvall, M.; Nevins, N.; Semus, S. F.; Senger, S.; Tedesco, G.; Wall, I. D.; Woolven, J. M.; Peishoff, C. E.; Head, M. S. J. *Med. Chem.* 2006, 49, 5912-5931. A Critical Assessment of Docking Programs and Scoring Functions.
5. Jordan, M. A.; Wilson, L.; *Nat. Rev. Cancer* 2004; 4, 253–265. Microtubules as a Target for Anticancer Drugs.
6. Ravelli R. B.; Gigant, B.; Curmi, P. A.; Jourdain, I.; Lachkar, S.; Sobel, A.; Knossow, M. *Nature* 2004, 428, 198–202. Insight into Tubulin Regulation from a Complex with Colchicine and a Stathmin-Like Domain.
7. <http://www.wwpdb.org/>
8. <http://www.accelrys.com/>
9. Carlson, H. A.; McCammon, J. A. *Mol. Pharm.* 2000, 57, 213-218. Accommodating Protein Flexibility in Computational Drug Design.
10. Sotriffer, C. A.; Dramburg, I. J. *Med. Chem.* 2005, 48, 3122-3. “In Situ Cross-Docking” To Simultaneously Address Multiple Targets
11. Irwin, J. J.; Shoichet, B. K. *J. Chem. Inf. Model.* 2005; 45, 177-82. ZINC - A Free Database of Commercially Available Compounds for Virtual Screening.
12. Jain, A. N. *J. Med. Chem.* 2003, 46, 499-511. Surflex: Fully Automatic Flexible Molecular Docking Using a Molecular Similarity-Based Search Engine.
13. Carr, R. A. E.; Congreve, M.; Murray, C. W.; Rees, D. C. *Drug Disc. Dev.* 2005, 14, 987-92. Fragment-based Lead Discovery: leads by design
14. MarvinBeans 4.1.2, 2006, ChemAxon (<http://www.chemaxon.com>)
15. Macromodel v. 5.1, Schrodinger, LLC, New York, NY, 1998.
16. Duda, R.; Hart; Stark, P. *Pattern Classification*; John Wiley and Sons: New York, 2001.
17. <http://perldoc.perl.org>
18. Kochev, N., Monev, V., Bangov, I.: Searching Chemical Structures. In: *Chemoinformatics: A textbook*. Wiley-VCH, 2003, 291–318.
19. Visualization of the Superposed Complexes was done with Jmol: <http://www.jmol.org>

Identification of Glaucoma Stages with Artificial Neural Networks Using Retinal Nerve Fibre Layer Analysis and Visual Field Parameters

Emiliano Hernández Galilea, Gustavo Santos-García,
and Inés Franco Suárez-Bárcena

Universidad de Salamanca, Dept. Surgery, Ophthalmology,
Alfonso X el Sabio s/n, Campus Miguel de Unamuno,
37007 Salamanca, Spain
{egalilea, santos, ifranco}@usal.es

Abstract. For the diagnosis of glaucoma, we propose a system of Artificial Intelligence that employs Artificial Neural Networks (ANN) and integrates the analysis of the nerve fibres of the retina from the study with scanning laser polarimetry (NFAII;GDx), perimetry and clinical data. The present work shows an analysis of 106 eyes of 53 patients, in accordance with the stage of glaucomatous illness in which each eye was found. The groups defined include stage 0, which corresponds to normal eyes; stage 1, for ocular hypertension; 2, for early glaucoma; 3, for established glaucoma; 4, for advanced glaucoma and 5, for terminal glaucoma. The developed ANN is a multilayer perceptron provided with the Levenberg-Marquardt method. The learning was carried out with half of the data and with the training function of gradient descent w/momentum backpropagation and was checked by the diagnosis of a glaucoma expert ophthalmologist. The other half of the data served to evaluate the model of the neuronal network. A 100% correct classification of each eye in the corresponding stage of glaucoma has been achieved. Specificity and sensitivity are 100%. This method provides an efficient and accurate tool for the diagnosis of glaucoma in the stages of glaucomatous illness by means of AI techniques.

Keywords: Glaucoma Diagnosis, Bioinformatics, Artificial Neural Networks, Nerve Layer Analysis, Visual Field Parameters, Laser Polarimetry.

1 Introduction

Glaucoma is one of the principal causes of blindness in the world. It is an illness which has an asymptomatic form until advanced stages, thus early diagnosis represents an important objective to achieve with the aim that people who present glaucoma maintain the best visual acuity throughout life, thereby improving their quality of life [2], [14], [19].

In our study we contribute the inclusion of artificial intelligence and neuronal networks in the diverse systems of clinical exploration and autoperimetry and laser polarimetry [12], [13], with the objective of facilitating the adequate staging in a rapid and automatic way and thus to be able to act in the most adequate manner possible.

There are different methods available to the ophthalmologist to classify patients in different stages of glaucoma. To establish the different stages in clinical practice, the

data of exploration and the degree of affectedness of the visual field (V.F.) are routinely employed. The application of Artificial Intelligence to Ophthalmology is relatively recent [4-7], [15], [16], [18], [25], orientated toward the analysis of the visual field.

More recently the analysis of the retinal nerve fibre layer (RNFL) by laser polarimetry has been included [2], [9-11], [21], [26]. All these diagnostic methods compose the criteria to be taken into account at the time of clinically classifying the stage of the glaucoma [22], [23].

Artificial Neural Networks owe their name to the parallelism in structure and function with the biological nervous system [14], [17], [24]. It consists of a group of neurones. Each neurone simultaneously receives various inputs from other neurones and adds them in accordance with the weights associated to each link, producing a response which depends on the level of inputs received and the weights associated to the links (see Figure 1).

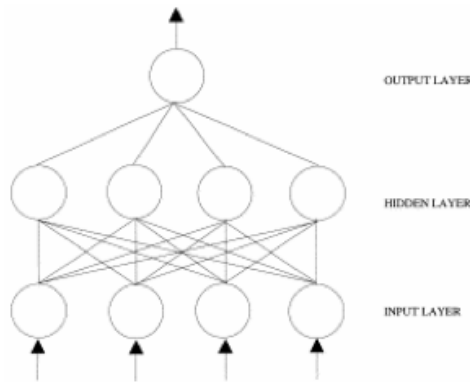


Fig. 1. ANN Model of a multilayer perceptron with one hidden layer

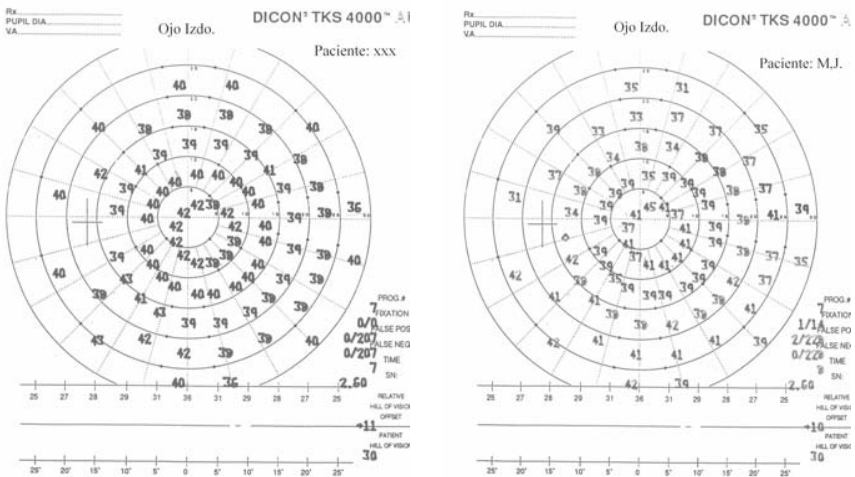


Fig. 2. Perimetry of a left eye of a normal patient (left image). Perimetry of a right eye of a patient with terminal glaucoma (right image).

For the diagnosis of glaucoma, we propose a system of Artificial Intelligence that employs Artificial Neural Networks and integrates, jointly, the analysis of the nerve fibres of the retina from the study with scanning laser polarimetry (NFAII;GDx), perimetry and clinical data.

2 Patients and Methods

The present work shows an analysis of 106 eyes of 53 patients, in accordance with the stage of glaucomatous illness in which each eye was found. The groups defined include stage 0, which corresponds to normal eyes; stage 1, for ocular hypertension; 2, for early glaucoma; 3, for established glaucoma; 4, for advanced glaucoma and 5, for terminal glaucoma. The developed ANN is a 16-30-1 multilayer perceptron provided with the Levenberg-Marquardt method [13], [17], [24].

To classify the eyes in groups, besides studying IOP (Goldmann tonometry) and the ophthalmoscopic study of the optic disc by means of biometry using Volk aspheric lens [14], [19]. Dicon TKS 4000 autoperimetry (Figure 2) has been used for the analysis of the visual field and laser polarimetry for the measurement of the thickness of the layer of retinal nerve fibres using the NFA-II, GDx fibre analyser (Figures 3 and 4).

The used ANN is a multilayer perceptron with backpropagation with a hidden layer. The input layer consists of 16 neurones, the hidden layer has 30 neurones and the output layer is a single neurone. The input neurones receive the values of 16 input variables and the output neurone obtains the value of the output variable that corresponds with the stage of the glaucoma for each eye. The definition of the 16 variables of input of the neuronal network consists of:

- **AGE.** Age of the patient.
- **CHAMBER.** Depth of the anterior chamber of the ocular globe.
- **IOP.** Intraocular pressure—expressed in millimetres of mercury.
- **OPTIC DISC.** Cup-to-disc ratio. If the result was less than 0.4, 0 was assigned; if between 0.4-0.5, 1 was assigned; if between 0.5-0.6, 2 was assigned and if between 0.7-0.9, 3 was assigned.
- **FIXATION.** Fixation losses by the patient from all those performed during the autoperimetry testing of visual field examination.
- **NS, TS, NI, TI.** Average of the values of the visual field in the superior nasal, superior temporal, inferior nasal and inferior temporal quadrant, respectively.
- **MEAN.** Mean of all the values of the visual field.
- **NORMAL DEVIATION SUPERIOR, INFERIOR, TEMPORAL, NASAL.** Difference of the thickness of the nerve fibre layer employing the GDx programme in the superior, inferior, temporal and nasal quadrant, respectively, for our patient compared with the normal patient of the same race and age.
- **NUMBER.** Experimental number extracted from all the values on acquiring an image employing NFA-II, GDx.
- **MEAN THICKNESS.** Mean of the thickness of all the pixels of the image; utilising the 65,536 points in an image considered valid.

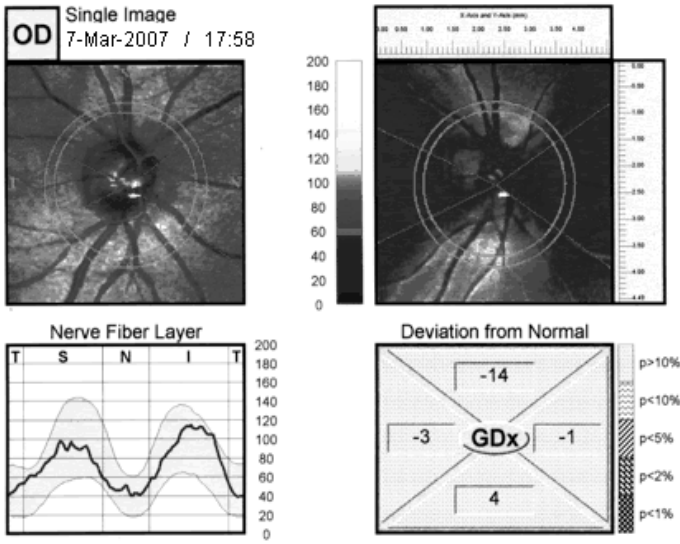


Fig. 3. Analysis with laser polarimetry for the measurement of the thickness of the layer of retinal nerve fibres using the NFA-II, GDX fibres analyser for a normal patient

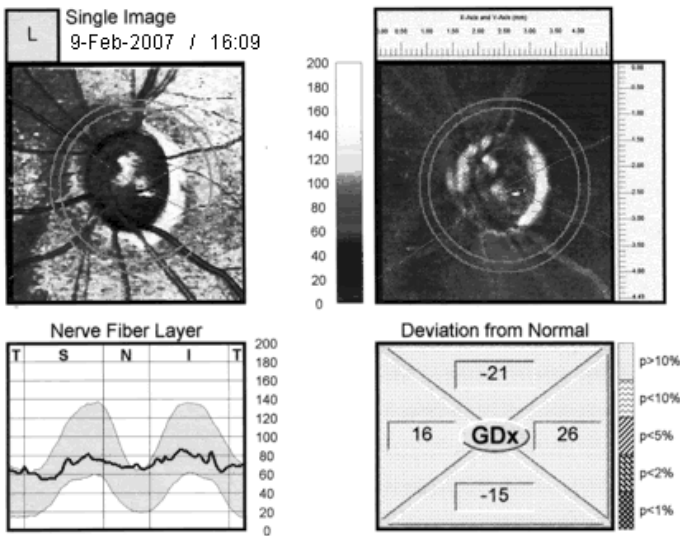


Fig. 4. Analysis with laser polarimetry for the measurement of the thickness of the layer of retinal nerve fibres using the NFA-II, GDX fibres analyser for a glaucomatous patient with stage 5 glaucoma

3 Results

The implementation of the neural network model has been carried out by means of the scientific computation platform *Matlab*, using the toolbox of *Neural Networks*. Once the model had been defined, half the data were randomly employed to train the ANN. The learning was carried out using half of the data from the study, in accordance with the diagnosis of an ophthalmologist, expert in glaucoma. The evolution of the process of learning is shown in Figure 5.

Finally, with the model of the ANN trained, the other half of the data were used to evaluate the yield of the model utilised. The learning was carried out with half of the data and with the training function of gradient descent w/momentum backpropagation and was checked by the diagnosis of an ophthalmologist, expert in glaucoma.

The model of neuronal network has been evaluated from the other half of the data. A 100% correct classification of each eye in the corresponding stage of glaucoma has been achieved. Therefore, the specificity and sensitivity are 100%.

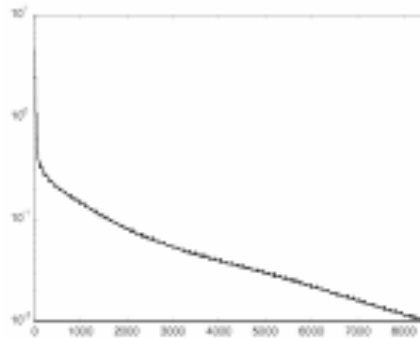


Fig. 5. Training evolution of our proposed neural network model

4 Discussion

Neuronal networks have been applied to evaluation of visual fields for the diagnosis of glaucoma and have compared favourably with the clinical evaluation made by clinical specialists [2], [6], [7]. The design of these networks is similar to that employed in this study [4], [5]. Nevertheless, the recent use of laser polarimetry in clinical exploration to the study of retinal nerve fibre layer thickness has increased the parameters available for performing a diagnosis and assigning a degree of affectedness of glaucoma.

In our study we have integrated both data obtained with polarimetry and algorithms originating from the exploration of the visual field by autoperimetry. The approach of this study differs basically from previous works in the inputs. The data utilised do not originate from only the parameters of visual field analyser. The neuronal network elaborated in this study is set up to process 16 input variables, which include those data extracted from the clinical exploration itself, such as optic disc appearance, from autoperimetry and laser polarimetry. This neuronal network therefore analyses the

degree of affectedness, both structural and functional, achieving the assignment of each case to the group corresponding to the stage of the glaucomatous illness. The high specificity and sensitivity obtained make this network superior to the expert systems and to other neuronal networks with similar architecture but with algorithms originating only from visual field examination [16-18].

In the future the definition of more exact and specific indices for incipient glaucomas or the prediction of the possibility that an ocular hypertension could evolve towards glaucoma will make the neuronal networks with this model an essential tool in clinical practice for early diagnosis and treatment of glaucoma.

This method provides an efficient and accurate tool for the diagnosis of glaucoma in the stage of glaucomatous illness by means of AI techniques. Further implementation for this model will include more algorithms and quantitative data obtained from new objective functional and structural tests.

Acknowledgments. We thank Prof. JoséLuis López and Prof. Narciso MartíOliet for helpful comments and discussions and Mr. G.H. Jenkins for his help with the English version of the ms. Work partially supported by Ministerio de Ciencia y Tecnología I+D+I Projects TIN2006-15660-15660-C02-01 and TIN2006-26882-E.

References

1. Antú A, Pastor JC: La Inteligencia Artificial en el Diagnóstico Precoz del Glaucoma. *Diagnóstico Precoz del Glaucoma. Sociedad Española de Oftalmología* (2005) 373-395.
2. Antú A, Zangwill L, Emdadi A, Weinreb RN: Nerve Fiber Layer Measurements with Scanning Laser Polarimetry in Ocular Hypertension. *Arch Ophthalmol* (1997) 115(3): 331-4.
3. Aramendá E: Electroretinograma con Estímulo Estructurado y Glaucoma. Universidad del País Vasco, E.H.U. Tesis doctoral (1992).
4. Bengtsson B, Bizios D, Heijl A: Effects of Input Data on the Performance of a Neural Network in Distinguishing Normal and Glaucomatous Visual Fields. *Invest Ophthalmol Vis Sci* (2005) 46:3730-6.
5. Bizios D, Heijl A, Bengtsson B: Trained Artificial Neural Network for Glaucoma Diagnosis Using Visual Field Data: a Comparison with Conventional Algorithms. *J Glaucoma* (2007) 16:20-28.
6. Brigatti L, Hoffman D, Caprioli J: Neural Networks to Identify Glaucoma with Structural and Functional Measurements. *Am J Ophthalmol* (1996) 121(5): 511-21.
7. Brigatti L, Nouri Mahdavi K, Weitzman K, Weitzman M, Caprioli J: Automatic detection of glaucomatous visual field progression with neural networks. *Arch Ophthalmol* (1997) 115(6): 725-8.
8. Burgansky-Eliash Z, Wollstein G, Chu T, Ramsey JD, Glymour C, Noecker RJ, Ishikawa H, Schuman JS: Optical Coherence Tomography Machine Learning Classifiers for Glaucoma Detection: a Preliminary Study. *Invest Ophthalmol Vis Sci* (2005) 46:4147-4152.
9. Caprioli J, Ortiz-Colberg R, Miller JM, Tressler C: Measurements of peripapillary nerve fiber layer contour in glaucoma. *Am J Ophthalmol* (1989) 108: 404.

10. Chi Q, Tomita G, Kitazawa Y: Reproducibility of Retinal Nerve Fiber Layer Thickness Measurements with a Scanning Laser Polarimeter. *Folia Ophthalmology Japan* (1995) 46: 387-391.
11. De Souza Lima M, Zangwill L, Weinreb RN: Scanning Laser Polarimetry to Assess the Nerve Fiber Layer. In Schuman J: *Imaging in Glaucoma*. Thorofare, NJ, Slack (1996) 83-92.
12. Franco Suárez-Bárcena, I, Franco Sánchez A, Hernández Galilea E, Cabezas R: Utilidad de la Asociación de la Autoperimetría al Análisis Cuantitativo del Espesor de Fibras Nerviosas Mediante Polarimetría Láser. Alicante, 74 Congreso de la Sociedad Española de Oftalmología (1998).
13. Franco Suárez-Bárcena I, Santos-García G, Hernández Galilea E, Franco Sánchez A: Aplicación de Redes Neuronales a la Clasificación Clínica en Estadíos de Glaucoma. Madrid, 76 Congreso de la Sociedad Española de Oftalmología (2000).
14. García Sánchez J, García Feijoo J, Arias A: Correlation Between the Nerve Fiber Layer Thickness and Delphi Perimetry in Glaucoma Patients. *Invest Ophthalmol Vis Sci* (1997) 38: 835.
15. Giangiacomo A, Garway-Heath D, Caprioli J: Diagnosing Glaucoma Progression: Current Practice and Promising Technologies. *Curr Opin Ophthalmol* (2006) 17:153-162.
16. Huang ML, Chen HY: Development and Comparison of Automated Classifiers for Glaucoma Diagnosis Using Stratus Optical Coherence Tomography. *Invest Ophthalmol Vis Sci* (2005) 46:4121-9.
17. Lippmann RP: An Introduction to Computing with Neural Nets. *IEEE ASSP Mag* (1987) 4-22.
18. Pan F, Swanson WH, Dul MW: Evaluation of a Two-Stage Neural Model of Glaucomatous Defect: an Approach to Reduce Test-Retest Variability. *Optom Vis Sci* (2006) 83:499-511.
19. Quigley HA, Enger C, Katz J *et al.*: Risk Factors for the Development of Glaucomatous Visual Field Loss in Ocular Hypertension. *Arch Ophthalmol* (1994) 112: 644-9.
20. Santos-García G: The Hopfield and Hamming Networks Applied to the Automatic Speech Recognition of the Five Spanish Vowels. In Albrecht RF, Reeves CR, Steele NC (eds.): *Artificial Neural Nets and Genetic Algorithms*. Viena, Springer-Verlag (1993) 235-242.
21. Tjon-Fo-Sang MJ, de Vries J, Lemij HG: Measurement by Nerve Fiber Analyzer of Retinal Nerve Fiber Layer Thickness in Normal Subjects and Patients with Ocular Hypertension. *Am J Ophthalmol* (1996) 122: 220-7.
22. Weinreb RN, Shakiba S, Sample PA *et al.*: Association between Quantitative Nerve Fiber Layer Measurement and Visual Field Loss in Glaucoma. *Am J Ophthalmol* (1995) 120: 732-8.
23. Weinreb RN, Shakiba S, Zangwill L: Scanning Laser Polarimetry to Measure the Nerve Fiber Layer of Normal and Glaucomatous Eyes. *Am J Ophthalmol* (1995) 119: 627-636.
24. Widrow B, Lehr MA: 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. *Proceedings of the IEEE* (1990) 78(9): 1415-1442.
25. Yu J, Abidi SS, Artes P, McIntyre A, Heywood M: Diagnostic Support for Glaucoma Using Retinal Images: a Hybrid Image Analysis and Data Mining Approach. *Stud Health Technol Inform* (2005) 116:187-192.
26. Zhou Q: Retinal Scanning Laser Polarimetry and Methods to Compensate for Corneal Birefringence. *Bull Soc Belge Ophtalmol* (2006) 89-106.

Dimensional Reduction in the Protein Secondary Structure Prediction – Nonlinear Method Improvements

Gisele M. Simas¹, Sflvia S.C. Botelho ¹, Neusa Grando², and Rafael G. Colares¹

¹ Fundação Universidade Federal do Rio Grande do Sul (FURG)

Av. Itália Km 8 – 96.200-090 – Rio Grande – RS – Brazil

² Universidade Tecnológica Federal do Paraná (UTFPR)

Av. Sete de Setembro – 80.230-901 – Curitiba – PR – Brazil

gisele_simas@yahoo.com.br, silviacb@furg.br,

neusa@cpgei.cefetpr.br, mugrah@gmail.com

Abstract. This paper investigates the use of the method of dimensional reduction Cascaded Nonlinear Components Analysis (C-NLPCA) in the protein secondary structure prediction problem. The use of the C-NLPCA is justified by the fact that this method manage to obtain a dimensional reduction that considers the nonlinearity of the data. In order to prove the effectiveness of the C-NLPCA, this paper presents comparisons of methods of components extraction, as well as, of existing predictors. The C-NLPCA revealed to be efficient, propelling a new field of research.

Keywords: C-NLPCA, Dimensional Reduction, Protein Secondary Structure Prediction.

1 Introduction

This paper investigates the use of artificial Neural Networks (NNs) as pre-processing stage in the prediction and classification of protein secondary structure, from Blast profiles [1] gotten through their sequence of amino acids. This classification serves to reduce searching space in other methods of prediction of protein tertiary structure that, in turn, is strictly related with the protein functionality [9].

Considering the large dimension of data to be treated, a pre-processing stage can increase the predictor performance. [10, 11, 12] use the traditional techniques of Principal Components Analysis (PCA) and Independent Component Analysis (ICA) for dimensional reduction, showing that such methods can contribute enough for the improvement of the predictors.

This paper suggests the use of a pre-processing stage, presenting a comparison between gotten results through practical experiences and the linear dimensional reduction methods used by [10, 11, 12]. In previous studies, originally, we propose the use of NNs through the Cascaded Nonlinear Components Analysis (C-NLPCA) [2] method in the secondary structures prediction problem. Such proposal is justified by the possibility of exploitation of the NNs nonlinear potential, combined with the reduction, acquires useful information for the classification phase. Moreover, the use of the C-NLPCA prevents the occurrence of an undesired simplification in the

variability of data, which could be caused by the use of a linear method, such as the PCA and the ICA.

This paper has 5 sections. In section 2, we present an overview of prediction of protein secondary structure problem. The third section is dedicated to the applied methodology, our pre-processing method for data reduction is related. Moreover, makes explanation of the classifiers. Our implementation, tests and results are analyzed in the fifth section, finally we present the conclusion.

2 The Prediction of Secondary Structure

The protein secondary structure prediction problem consists of, from its primary structure (sequence of amino acids), classifying each of its constituent amino acids in one of the recurrent substructures in the three-dimensional conformation can be grouped in: α -helixes, β -sheets and coils. The algorithms that simulate the folding of the sequences, developed until the moment, can not accurately simulate the laws that control this process. Therefore, NNs show as a promising method and come frequent being used, demonstrating good results [3, 4, 10, 13, 14].

Among the predictors found in literature, the predictor CONSENSUS [3] combines four others: DSC [5], PHD [8], NSSP [15], and PREDATOR [3], without making data reductions. In turn, taking into account the high dimension of the data to be treat, [10, 11, 12] proposed the application of linear dimensional reduction techniques, PCA and ICA, under the data to be used by the predictors. In Section 5, an analysis of the gotten results by this work in comparison with other predictors is presented.

3 Methodology

The proposal of this study is to evaluate the improvements that the use of a nonlinear method for dimensional reduction of data can bring in the prediction of secondary structure. Therefore, a similar procedure to those of [10, 11, 12], which used a linear method, was carried through. The Figure 1 shows our propose to treat the problem. From a NLPCA reduction method, a large Blast profiles input are reduced. After a NN predictors used to classify the proteins in α -helixes, β -sheets and coils.

3.1 Using Blast Profiles in the Prediction

The 396 sequences of proteins present in the CB396 [16] were submitted to Position Specific Iterated – Basic Local Alignment Search Tool (PSI_Blast) [1]. This tool searches for similar proteins in the selected data base. The system returns as output PSI_Blast profiles which are composites for values that represent the occurrences of each amino acid in each sequence position. The searched protein receives a score that is stored in the Specific Score Matrix Position (PSSM) [10]. Such a matrix has the dimension $p \times n$, where p represents 20 amino acids and n the number amino acid that composes protein sequence. This PSSM matrix is re-passed to the reduction stage, which will be described in the next section.

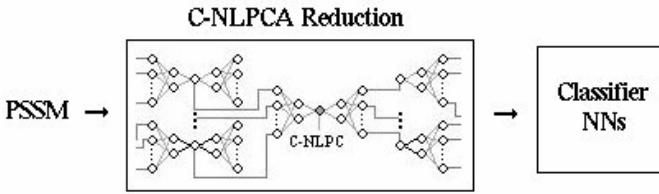


Fig. 1. Proposed solution

3.2 Dimensional Reduction

The PSSM score matrix of each protein is covered by overlapped windows of size (Figure 2) w , displaced until reaching the original size of the protein. The dimension of each window is $p=20 \times w$ values (representing the frequencies of 20 existing amino acids $\times w$ amino acids belonging to the window). Each set of p values is reduced from p to q . Later, each set of q components is forwarded to the classification stage, which has the task to predict the corresponding secondary structure to the central amino acid of the window.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
(1) R	-1	5	-1	-1	-4	2	0	-2	-1	-3	-3	3	-2	-3	-2	-1	-1	-3	-2	-3
(2) T	0	-2	-1	-2	5	-1	-1	-2	-2	-2	-2	-1	-2	-3	-2	3	4	-3	-2	-1
(3) D	-1	-3	3	0	-4	-2	-2	6	-2	-5	-5	-2	-4	-4	-3	-1	-2	-4	-4	-4
C	-2	-1	1	-3	9	-1	-3	-3	-3	-2	-3	-3	-2	0	-4	-1	1	-3	-3	-2
Y	-3	-3	-4	-5	2	-3	-4	-5	0	-3	-3	-3	-2	2	-5	-3	-3	1	9	-3
G	1	-3	-2	-3	-4	-3	0	6	-3	-5	-5	-3	-4	-5	-3	-1	0	-4	-4	-4
N	-3	-2	4	5	4	0	1	-3	1	-4	-4	-1	2	-4	-3	-1	-2	-5	-4	-3
V	-2	-4	-4	-5	-3	-4	-4	-5	-5	6	0	-4	0	-2	-4	-3	1	-4	-3	3
N	1	1	2	-2	-3	0	1	-3	-2	-2	-2	-2	6	-3	-3	2	1	3	-3	-2
R	0	1	1	0	2	-2	0	-3	-3	1	-2	2	-2	-4	-3	2	3	-4	-3	-2
I	0	-4	-5	-5	-3	-4	-4	-5	-4	4	3	-4	4	-2	-4	-3	-2	-4	-3	3
(w-1) D	-2	-2	-1	4	-4	3	4	-2	1	-4	-4	1	-3	-5	-1	2	0	-5	-3	-4
(w) T	1	-3	-2	-3	-2	-2	-2	-3	-3	-2	-3	-2	-2	-4	-3	1	7	-4	-3	-2
T	-1	-2	-2	-2	-3	-2	-2	-3	-3	-2	-3	-1	-2	-1	4	3	4	-4	-3	0
G	-1	-3	-2	-2	-3	-2	-3	6	-3	-4	-4	0	-3	-4	-3	-1	1	-4	-4	-4
A	5	-3	-3	-3	4	-2	-3	-2	-3	0	-3	-2	-2	-3	-2	0	-2	-4	-3	-1
S	1	-2	1	-2	-2	-2	-2	-2	-3	-4	-2	-3	-4	1	5	2	-4	-3	-3	-3
C	1	-3	-2	1	8	-1	1	0	-3	-3	-3	-2	-3	-3	-1	-2	-2	2	-3	-3

} window
[1..w]

Fig. 2. One window of the PSSM score matrix

Principal Components Analysis (PCA). The PCA is a technique that can be used to supply a statistic analysis of the data set. This analysis is concerned with the extraction of the factors that better represents the structure of interdependence between variables of large dimensions. Therefore, all the variables are analyzed simultaneously, each one in relation to all the others, aiming at determining factors (principal components) that maximize the explanation of variability existing in the data. However, the PCA is indicated for the analysis of variables that have linear relations [7]. In this paper, we used the Expectation Maximization (EM) [9] algorithm in order to calculate the PCA.

Independent Components Analysis (ICA). The ICA is a method of linear data transformation, which goal is to find a data representation that minimizes the statistic dependency of the components represented. This way, to obtain the components of a vector x , the ICA tries to find a linear transformation $s=Wx$, in which, the s_i components are as independent as possible [11]. This is accomplished through the maximization of a function $F(s_1, \dots, s_m)$ capable of measuring the independency of the components. Both PCA and ICA are projection techniques on there own space. The principal difference lies in the fact that the PCA generates uncorrelated components, while the ICA generates independent components [11].

Cascaded Nonlinear Components Analysis (C-NLPCA). In this paper, it is suggested the use of C-NLPCA (see details in [2]) to provide a nonlinear mapping of data, since linear methods, such as the PCA and ICA, may introduce undesirable simplifications in the analysis of variables with nonlinear relations. The C-NLPCA method is based on the cascading in layers of simple NLPCAs NNs [8], aiming the nonlinear treatment of high dimension data.

The NLPCA Analysis. NNs for the analysis of nonlinear principal components, called NLPCAs, are Multi-Layers-Perceptron (MLPs) NNs composed by reduction and expansion stages. In the reduction, NNs with five layers are used: input (p neurons), hidden of codification (m neurons), bottleneck (r neurons), hidden of decoding (m neurons) and output (p neurons). Neurons of the codification and decoding layers use nonlinear functions, while those of input, bottleneck and output use linear functions of activation. These characteristics make the NNs model a set of functions. The inputs in a p -dimensional space, when forwarded to the bottleneck layer, are mapped by the r -dimensional space. The activation values of the bottleneck layer neuron supply the nonlinear principal components. Each NLPCA NN is trained to get a mapping between input and output, which minimize the function:

$$\min \sum_{i=1}^n |\bar{X}_i - \bar{X}'_i| \quad (1)$$

where \bar{X}'_i is the output of the NN for each \bar{X}_i input.

The output produced by the expansion will contain the cumulative error of samples, so the $\bar{X}_i - \bar{X}'_i$ residue value can be used for the attainment of the second principal component and, thus, successively.

From NLPCAs Sets to C-NLPCA System. Due to the intrinsic problems of saturation of the NNs, the applicability of the NLPCA is restricted to the cases where $p \ll n$ [2]. In order to avoid such limitation, we have proposed an architecture where NLPCAs are grouped in layers, see [2] for more details. In the reduction stage, data of p initial dimensions are grouped in a series of small NLPCAs NNs of p' input neurons. Each NLPCA reduces its respective inputs of p' for 1 dimension. The reduced data (principal locals) again are grouped and reduced successively in subsequent layers (reduction layers). Next, the bottleneck neuron of the last NLPCA NN will supply the first global principal component (C-NLPC).

The Expansion Stage. After of the reduction layers, a set of MLP NNs composes the expansion stage. The output values gotten by bottleneck NLPCA are used as input for MLP NNs with 1 input neuron and p' output neurons. These NNs, designated as expansion NNs, go being disposed successively, layer to layer, until the reproduction of output sets whose final dimension is equal to the presented to the system: p . During the expansion, there is not a training of NNs, only the value of the principal component is propagated.

Since every principal local are combined successively, in the stage of reduction, the C-NLPCA considers all the relations of neighborhood between the variables. As well as the relations between all the windows that compose PSSM matrix of a given protein are also analyzed in their use as samples in the training.

3.3 Classification: Using NN Predictor

The proposed classifier associates three MLPs NNs with distinct topologies, aiming the choice of a better local minimum. The distinction of the NNs is associated with the amount of hidden layer neurons. Each output value is associated according to the established norms in the stage of combination of rules. The training of NNs is made by epoch, based on the Resilient Propagation (RPROP) [7].

The combination of the three NNs of different topologies intends to improve the prediction of the secondary structures [3]. The NNs outputs are combined by rules: Voting, Average, Product, Maximum and Minimum [10]. After the combination of NNs, the results are evaluated using a reduced variation of the Jack-Knife process [10]. The exactness of prediction is measured by the value of $Q3$, obtained through the equation:

$$Q3 = \frac{\text{correct_number_a min oacids}}{\text{total_number_a min oacids}} \times 100 \quad (2)$$

4 Tests and Results

A tool in C++ was developed aiming at to implement the preprocessing stage in CNLPCA and the classifier NNs. Profiles of window of size $w=13$ were analyzed, resulting data with the dimension of 260 elements (13×20 amino acid) to be processed and reduced. The reduced data were placed in predictor NNs with 30, 35 and 40 neurons in the hidden layer.

From every profile, the 80 first principal components were extracted. This way, a reduction from $p=260$ to $q=80$ was accomplished. The data of dimension $p=260$ are divided in series of NLPCAs with $p'=10$ entrance neurons, see Figure 3.

The dimensional reduction from 260 to $r=1$ was gotten 80 times, being that, in each repetition, the residue is normalized and injected as input for the calculation of the next component. Table 1 shows the accuracy percentage of the predictors related in Section 2 and obtained in this paper.

GMC predictor was developed by [4] with the same solution method adopted in this paper, but not using any reduction data, and, for this reason, making difficult the classifiers training. The predictor developed by Guimarães [10, 11, 12], does a dimensional reduction 80 and 180 principal components, through the PCA and ICA.

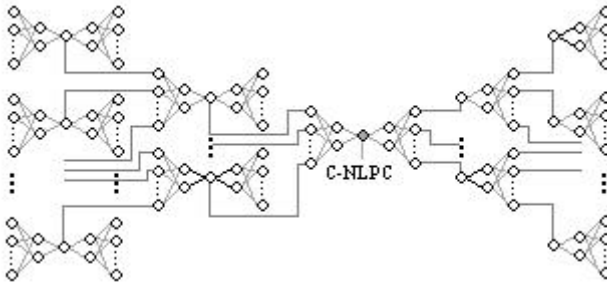


Fig. 3. C-NLPCA system

The predictor adopted in this paper, using C-NLPCA reduction, presented better results than the GMC, and this is due to the fact that the dimensional reduction took into account the data nonlinearity, besides providing a more effective training of the classificatory NNs.

The C-NLPCA obtained a better performance than the predictors, who used linear dimensional reduction, PCA and ICA, with a larger number of principal components. This proves the importance of considering the data nonlinearity.

Comparing the C-NLPCA with the PCA and ICA is much more extensive process, but even though, the results justify there use. In order to reduce the time of processing to obtain the principal components, Both the NNs of the architecture C-NLPCA and the ones used in the classify stage, were trained through the RPROP [8] method. As parameters, we used $\Delta_{\min}=1e-6$, $\Delta_{\max}=50$, $\eta^+=1,2$ e $\eta^-=0,5$ (check [8] for details).

Table 1. The best results of our C-NLPCA approach

Method	Q3 (%)	Method	Q3 (%)
PHD [14]	71.9	PCA [10] – 80 PCs	73.8
DSC [7]	68.4	ICA [11] – 80 PCs	73.9
PREDATOR [3]	68.6	PCA [12] – 180 PCs	74.5
NNSSP [15]	71.4	ICA [12] – 180 PCs	74.9
CONSENSUS [3]	72.9	C-NLPCA – 80 PCs	76.1
GMC [4]	75.9		

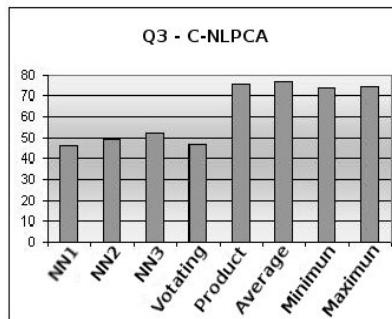


Fig. 4. The values of the 80 PCs

Regarding the classifier stage, the combination rules had contributed in the increase of precision of the results. Figure 4 shows that the application of the “average rule” has gotten the best results. The neural network with the most neurons in the hidden layer (NN3 in Figure 4) presents a better performance in comparison to the others.

5 Conclusions

This paper presented a study on computational methods for the classification of protein secondary structures in: α -helices, β -sheets and coils. It was investigate the use of the C-NLPCA method as dimensional reduction stage of the input data of the classifiers.

The protein primary structures contained in CB396 bank were submitted to the software PSI_Blast that located distant homologous proteins. From the similar proteins, the PSI_Blast generated a PSSM scores matrix. Such a matrix was passed to a reduction stage and the reduced data were passed through classifier NNs with different topologies, whose outputs were combined.

The obtained results with C-NLPCA confirm the effectiveness in the dimensional reduction of data for the prediction of secondary structures. It stimulates researches on the application of this method. Studies can be done concerning the topology of the NNs used in C-NLPCA.

Acknowledgments. This work was supported by Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

References

1. Altschul, S., Madden, T., Shaffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.: Gapped Blast and PSI-Blast: A New Generation of Protein Database Search Programs. *Nucleic Acids Research* (1997).
2. Botelho, S.S.C., Bem, R.A., Almeida, I.L., Mata, M.M.: C-NLPCA: Extracting Non-Linear Principal Components of Image Datasets. *Simpósio Brasileiro de Sensoriamento Remoto* (2005).
3. Cuff, A.J., Barton, J.G.: Evaluation and Improvement of Multiple Sequence Methods for Protein Secondary Structure Prediction. *Proteins Structure, Function and Genetics*, 34 (1999) 508–519.
4. Guimaraes, K.S., Melo, J.C.B., Cavalcanti, G.D.C.: Combining few Neural Networks for Effective Secondary Structure Prediction. *Proceedings of the Third IEEE Symposium on Bioinformatics and Bioengineering* (2003) 415–420.
5. Jones, D.T.: Protein Secondary Structure Prediction Based on Position-Specific Scoring Matrices. *J. Mol. Biol.* 292 (1999) 195–202.
6. Khattree, R., Naik, D.N.: *Multivariate Data Reduction and Discrimination with SAS Software*. Cary, NC: SAS Institute Inc. (2000).

7. King, R., Sternberg, M.: Identification and Application of the Concepts Important for Accurate and Reliable Protein Secondary Structure Prediction. *Proteins Science* 5 (1996) 2298–2310.
8. Kramer, M.A.: Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE Journal* 37 (1991) 233–243.
9. Lehninger, A.L.: *Principles of Biochemist*. Editora Sarvier, São Paulo (1984).
10. Melo, J.C.B., Cavalcanti, G.D.C., Guimaraes, K.S.: PCA Feature Extraction for Protein Structure Prediction. *International Joint Conference on Neural Networks* (2003a) 2952–2957.
11. Melo, J.C.B., Cavalcanti, G.D.C., Guimaraes, K.S.: Protein Secondary Structure Prediction with ICA Feature Extraction. *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing – Special Session on Bioinformatics* (2003b).
12. Melo, J.C.B., Cavalcanti, G.D.C., Guimaraes, K.S.: Protein Secondary Structure Prediction: Efficient Neural Network and Feature Extraction. *IEE Electronics Letters*, Vol. 40, n. 21 (2004) 1358-1359.
13. Qian, N., Setnowski, T.J.: *Predicting the Secondary Structure of Globular Proteins Using Neural Network Models*. Baltimore (1988).
14. Rost, B., Sander, C.: Combining Evolutionary Information and Neural Network to Predict Secondary Structure. *Proteins* 19 (1994) 55–72.
15. Salamov, A., Solovyev, M.: Prediction of Protein Secondary Structure by Combining Nearest-Neighbor Algorithm and Multiple Sequence Alignments. *Journal of Molecular Biology* 247 (1995) 11–15.
16. University of Dundee and The Barton Group. Cb396. <http://www.compbio.dundee.ac.uk/>, 20 jul 2005.

Focused Crawling for Retrieving Chemical Information

Zhaojie Xia, Li Guo, Chunyang Liang, Xiaoxia Li, and Zhangyuan Yang

State Key Laboratory of Multiphase Reactions, Institute of Process Engineering,
Chinese Academy of Sciences, Beijing, 100080
{zjxia, lguo, cyliang, xxia, zzyang}@home.ipe.ac.cn

Abstract. The exponential growth of resources available in the Web has made it important to develop instruments to perform search efficiently. This paper proposes an approach for chemical information discovery by using focused crawling. The comparison of combination using various feature representations and classifier algorithms to implement focused crawlers was carried out. Latent Semantic Indexing (LSI) and Mutual Information (MI) were used to extract features from documents, while Naive Bayes (NB) and Support Vector Machines (SVM) were the selected algorithms to compute content relevance score. It was found that the combination of LSI and SVM provided the best solution.

1 Introduction

The advent of the Web has created a new medium for publishing and disseminating various sorts of information. As of its exponential growth, the Web has now become the biggest human knowledge base, including chemical information. Some techniques have been proposed to facilitate the chemical information seeking process in the Web. Chemistry web directories, such as ChemDex [1], Links for Chemists [2] and ChIN [3], face the problem of high cost of maintenance and limitation in coverage due to labor-intensive process and ongoing human effort. Another key technique is focused crawling which searches and retrieves only the subset of the Web that pertains to a specific topic of relevance. A focused crawler aims to collect as many relevant pages with respective to some predefined topics and as few irrelevant ones as possible. Conceptually, we can treat a focused crawler as a general crawler which traverses the Web graph according to a topic-dependent ordering instead of breadth-first ordering. What's important during the progress is the ranking algorithm for identifying the next most appropriate link to follow.

Early works in focused crawling include FishSearch [4], SharkSearch [5] and intelligent crawling [6]. However it seems that the comparison of combination using various feature representations and classifiers to implement focused crawlers has not been tried before, especially for the topic of chemistry. In this study, we combine two feature representations (LSI and MI) and two classification algorithms (SVM and NB) to implement chemistry focused crawler.

2 Methodologies

The documents in document collection are represented by a term-document matrix, which contains the weight values of index terms occurring in each document.

Generally the matrix is very sparse because there are so many terms in the document collection. Feature Selection and extraction are special optimization techniques, which will help to remove noisy terms and reduce the dimensionality. In this paper, we use Mutual Information (MI) and Latent Semantic Indexing (LSI) for feature representations of documents.

2.1 Feature Selection

Mutual Information is a specialized case of the notions of cross-entropy, known as one of the most effective methods for select topic-specific features [7]. The MI weight of the X_i in the topic V_j is defined as:

$$MI(X_i, V_j) = P[X_i \wedge V_j] \log \frac{P[X_i \wedge V_j]}{P[X_i] \wedge P[V_j]} \quad (1)$$

Mutual Information can be regard as measure of how much the joint distribution of feature X_i and topic V_j deviate from a hypothetical distribution.

Latent Semantic Indexing is an approach that maps documents as wells as terms to a representation in the so-called latent semantic space [8]. It is concept-based automatic indexing method that suggests additional relevant terms and to reveal the “hidden” concepts of a given document collection while eliminating high order noise. The attractive point of LSI is that it captures the higher order latent structure of word usage across documents rather than just surface level term choice. LSI uses a truncated Singular Value Decomposition (SVD) of the term-document matrix for dimensionality reduction:

$$M = TSD \quad (2)$$

The original term-document matrix M is decomposed into a reduced rank term matrix T , a diagonal matrix of singular values S and a document matrix D . The row vector of T and column vector of D are the projections of word vector and document vectors into singular value space.

2.2 Corpus

ChIN is a comprehensive chemistry web directory created in 1996 and has been well maintained since then, and now it has more than 10,000 carefully selected chemistry resources on Internet. A random sample of 1,000 documents directly linked by ChIN was used as the positive training set of the chemistry focused crawlers. Another 1 000 random sample from a variety of other topics such as arts, sports and so on taken from the DMOZ directory are used for negative set. All pages are parsed and tokenized, the stop-words are removed and the remaining words are stemmed. Feature selection is done by using MI and LSI. TF-IDF weighting is used to represent the pages.

2.3 Classification

After feature selection and document representation, a training phase of classification is carried out for building a mathematical decision model base on pre-built term-document matrix. Our focused crawler using SVM and NB as topic-specific classifier.

SVM is a relatively new machine learning techniques first introduced by Vapnik in 1995 to solve two-class pattern recognition problem based on the structural risk minimization principle from the computational learning theory [9]. SVM seeks a decision surface to separate the training data points into two classes and makes decisions based on the support vectors that are selected as the only effective elements in the training set. In this paper we use the open-source implementation SVM provided by Chang et al [10]. Linear form of SVM where training amount to finding a hyperplane in the m -dimensional feature vector space.

A Naive Bayes classifier is a simple probabilistic classifier. Bayes theorem is the basis of this classifier that the features are assumed to be independent [11].

$$P(c_j|d_i) = P(c_j) \prod_{k=1}^M P(w_{ik}|c_j) \quad (3)$$

Where w_{ik} is the feature that occurs in the k th position in document d_i . $P(c_j|d_i)$ is the probability that document d_i belongs to class c_j . $P(c_j)$ is the probability of class c_j , and $P(w_{ik}|c_j)$ is the probability that class c_j belongs to feature w_{ik} .

2.4 Focused Crawling

In order to achieve the goal of efficiently retrieving topic-relevant resource, crawler should be forced to stay focused on this topic while collecting web pages. The key challenge during the progress is identify the next most appropriate link to follow from its crawling frontier. Our approach utilizes a classifier to implement an evaluation function and predict relevance of a document to a specific topic for priority control. The classifier computes and assigns relevance scores of hyperlinks embedded in

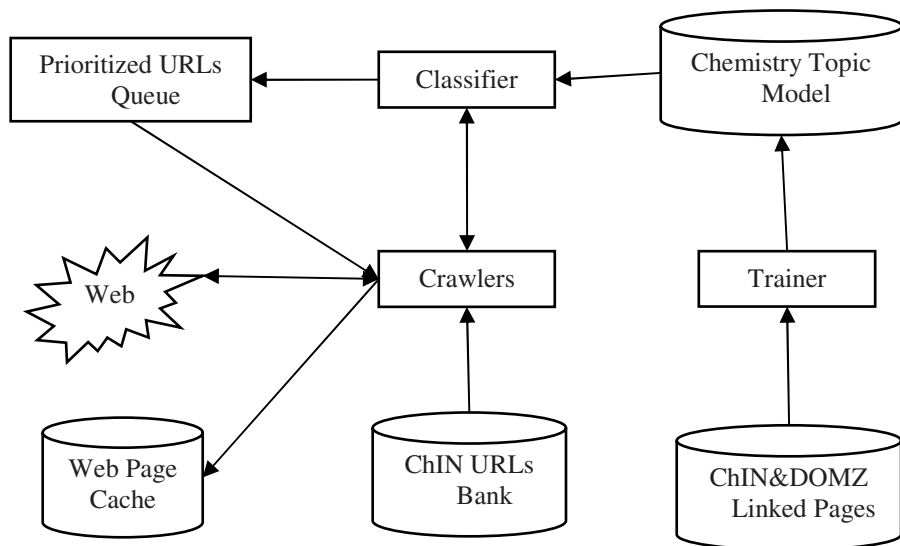


Fig. 1. The workflow of the chemistry focused crawler

pages. The assigned hyperlinks are then added to a priority queue (frontier) based on these scores. In the next iteration, crawler picks the hyperlink with the highest score in the queue to crawl. The workflow of chemistry focused crawler is shown in Fig. 1.

3 Results

There are many indicators of the performance for crawlers. Precision (harvest rate) and recall are two frequently used of them. In our experiments, we will only measure precision of crawlers for the total number of relevant pages is unknown which is needed for the calculation of recall. Precision can be defined as the percentage of the predicted pages that are classified correctly for all crawled pages.

Four different focused crawlers, namely MI_SVM crawler, LSI_SVM crawler, MI_NB crawler, LSI_NB crawler, are evaluated to determine crawling precision in following experiments. Breadth-First (BF) crawling widely used by general search engines has the simplest strategy for crawling which uses a FIFO (First In First Out) queue and fetches links in the order of with least hops first. In this study, BF acted as baseline for crawling experiments.

For the MI feature representation, n keywords with the highest mutual information were selected for the final feature representations. For LSI approach, indices corresponding to the largest n singular values were used for the final feature representations. In both cases, n was 100, 200, 300, 400, 500, 600, 700 and 800.

After documents from ChIN and DMOZ were represented by the MI and LSI features, SVM and NB algorithms were subsequently applied to train and direct the focused crawling. For each crawler, we use the same seed URL set from ChIN URLs' database, and we ran the crawler until it had visited 100,000 for evaluation. As expected, BF displayed the worst precision and provided us with a baseline for all measures; it dropped quickly with the number of crawled pages increasing. After crawling 100,000 pages, the precision of BF dropped near to 10%. Summarized results for four focused crawling are shown in Table 1.

Table 1. Summarized results for four types of chemistry focused crawling

Number of features	Precision			
	LSI_SVM	LSI_NB	MI_SVM	MI_NB
100	76.742%	69.583%	69.267%	66.168%
200	77.965%	70.268%	70.164%	66.939%
300	76.824%	68.945%	71.363%	68.257%
400	77.124%	69.632%	71.773%	68.835%
500	76.653%	69.344%	72.195%	69.265%
600	76.027%	68.164%	72.856%	68.567%
700	76.369%	68.697%	72.134%	68.953%
800	75.222%	68.023%	71.152%	68.122%

The study shows that the four focused crawlers kept relative high precision and clearly outperformed BF in gathering web pages on the topic of chemistry. One also can see the combination of LSI and SVM (LSI_SVM) provided the best solution. The LSI_SVM focused crawling performed best and reached precision of 77.965% with 200 LSI indices. MI_SVM was second best combination when the number of features was equal to 600. The performance LSI_NB was slightly better than MI_NB.

4 Discussions

This study analyzed four focused crawling for retrieving chemical information. These focused crawlers were formed by combining two feature representations (LSI and MI) and two classification algorithms (SVM and NB) from machine learning. The study shows that the four focused crawling can keep a high precision to collect chemistry-relevant pages. It was also found that the combination of SVM and LSI provided the best performance in gathering web pages on the topic of chemistry.

In the future, we will consider to implement a focused crawler with document segmentation to evaluate relevance score of unvisited URLs respectively that may possible further improve crawling precision. More extensive training of classification, as well as new classification methods, such as adaboost, will be further investigated. The approach proposed in this paper can be used to retrieve for more specific topics in chemistry or other disciplines.

Acknowledgments. This research is supported by a grant of Natural Science Foundation of China (project No. 20273076).

References

1. University of Sheffield. ChemDex. <http://www.chemdex.org/>.
2. University of Liverpool. Links for Chemists. <http://www.liv.ac.uk/Chemistry/Links/links.html>
3. Institute of Process Engineering, Chinese Academy of Sciences. Chemistry Portal, Chinese National Science Digital Library, <http://chin.csdl.ac.cn/>
4. Bra D.P., Houben G., Kornatzky Y., Post R.: Information Retrieval in Distributed Hypertexts. In Proceedings of the 4th RIAO Conference (1994) 481-491.
5. Hersovici M., Heydon A., Mitzenmacher M.: The Sharksearch Algorithm-An Application: Tailored Web Site Mapping. Proceedings of the 7th International World Wide Web Conference (1998) 213-225.
6. Aggarwal C., Al-Garawi F., Yu P.: Intelligent Crawling on the World Wide Web with Arbitrary Predicates. Proceedings of the 10th International World Wide Web Conference (2001) 96-105.
7. Yang Y., Pedersen O.: A Comparative Study on Feature Selection in Text Categorization. Proceeding of the 14th International Conference on Machine Learning (1997) 412-420.

8. Berry M., Dumais S., Letsche T.: Computation Methods for Intelligent Information Access. Proceedings of the 1995 ACM/IEEE Supercomputing Conference 1995.
9. Cortes C., Vapnik V.: Support Vector Networks. Machine Learning, 20(1995) 273-297.
10. Chang C., Lin C.: LIBSVM: A Library for Support Vector Machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
11. Lewis D.D.: Naive Bayes at Forty: The Independence Assumption in Information Retrieval. Proceedings of ECML-98, 10th European Conference on Machine Learning (1998) 4-15.

Optimal Portfolio Selection with Threshold in Stochastic Market

Shuzhi Wei¹, Zhongxing Ye¹, and Genke Yang²

¹ Department of Mathematics, Shanghai Jiao Tong University
800 Dong Chuan Road, 200240 Shanghai, China
{szwei, zxye}@sjtu.edu.cn

² Department of Automation, Shanghai Jiao Tong University
800 Dong Chuan Road, 200240 Shanghai, China
gkyang@sjtu.edu.cn

Abstract. This paper considers a continuous-time optimal portfolio problem, where the price processes of assets depend on the state of the stochastic market, which is assumed to follow a diffusion process. And trading in the risky asset is stopped if the market state hits a predefined threshold. The problem is formulated as a utility maximization with random horizon. Using the techniques of dynamic programming and Feynman-Kac representation theorem, we obtain a stochastic representation of optimal portfolio. Furthermore, in some special case, the closed-form of optimal portfolio is derived. Finally, we present computational results that show the differentiation between this proposed model and classical Merton model.

Keywords: Optimal Portfolio, utility maximization, threshold, HJB equation, Feynman-Kac representation.

1 Introduction

Since Merton's models [1, 2], the continuous-time optimal portfolio models has been heavily researched, for example Cox and Huang [3], Cox et al. [4], Duffie and Zame [5], He and Pearson [6], Karatzas et al. [7]. Recently, Portfolio problems, where the price processes of assets are affected by random market environment, have drawn more attentions. An investment-consumption model with regime switching was studied in Zariphopoulou [8]; an optimal stock selling for a Markov-modulated Black-Scholes model was derived in Zhang [9]; a stochastic approximation approach for the liquidation problem could be found in Yin and Liu and [10]; a continuous-time version of Markowitz's mean-variance portfolio selection with regime switching was studied by Zhou [11].

This paper considers a continuous-time portfolio selection problem for utility maximization of terminal wealth. Its special feature is that we assume the risky asset price is correlated with the market state. Investor predefines a threshold according to the market state. Once the market state hits the threshold, the investor stops investment in the risky asset. Actually, the proposed model is quite realistic. With the

experience of recent failure of large financial institutions such as the Barings Bank, sufficient risk control measures are clearly essential. Therefore, it is an approach for some investors predefines a threshold so that to avoid catastrophic consequence. The problem is formulated as optimization problem with random horizon. Research problems in this version have been done in various contexts see Richard [12], Karatzas [13], Bouchard [14], Kraft [15]. Following the method of Kraft [15], we obtain the optimal portfolio for the proposed model.

This paper is structured as following: The model is described in section 2. Section 3 shows stochastic representation result of the optimal portfolio. In the section 4, the closed-form of the optimal portfolio is given for a special case. Finally, we present computational results In section 5.

2 The Portfolio Model

Let $(\Omega, F, P, \{F_t\}_{\geq 0})$ be a filtered probability space. $W_0(t)$ and $W(t)$ are independent Brownian motions, and the filtration F_t is generated by $W_0(t)$ and $W(t)$.

We assume that the opportunities of an investor include a locally risk-less asset and a risky asset, and the price of risky asset is influenced by the market state. The objective of investor is to maximize the utility of terminal wealth. We want to emphasize that the terminal time is an F- stopping time τ . More precisely, the investment horizon depends on the market state. The investor predefines a threshold which consists of upper boundary H and lower boundary L of the market state.

We assume that the market state $Y(t)$ follows the dynamics:

$$\begin{cases} dY(t) = \mu(Y(t), t)dt + \beta(Y(t), t)dW_0(t) \\ Y(0) = y_0 \end{cases} \tag{1}$$

The investment horizon of the investor is given by a stopping time τ triggered by the market state process Y in the following sense $\tau = T \wedge \tilde{\tau}^0$, here T is a deterministic time and $\tilde{\tau}^0$ is the first hitting time of Y at the given threshold, i.e., $\tilde{\tau}^0 = T_H^0 \wedge T_L^0$, $T_H^0 = \inf\{s, s \geq 0, Y(s) = H\}$, $T_L^0 = \inf\{s, s \geq 0, Y(s) = L\}$;

The price of locally risk-less asset is subject to the dynamics process

$$\begin{cases} dP_0(t) = r(Y(t), t)P_0(t)dt \\ P_0(0) = 1 \end{cases} \tag{2}$$

where $r(y, t)$ is the short-term risk-less interest rate.

The price process of the risky asset satisfies the stochastic differential equation (SDE)

$$\begin{cases} dP(t) = P(t) \left\{ b(Y(t), t) dt + \rho \sigma(Y(t), t) dW_0(t) + \sqrt{1 - \rho^2} \sigma(Y(t), t) dW(t) \right\} \\ P(0) = p > 0 \end{cases} \tag{3}$$

where $b(y, t)$ is the appreciation rate; $\sigma(y, t)$ is the volatility of the risky asset; the constant $|\rho| \leq 1$ is the correlation coefficient between the price of risky asset and market state process.

We assume the coefficients of the above equations satisfy the following conditions:

(A1) $\mu(y, t)$, $\beta(y, t)$, $r(y, t)$, $b(y, t)$, $\sigma(y, t)$ are uniformly Lipschitz continuous in $(y, t) \in R \times [0, T]$

(A2) There exist two positive constants β_0, σ_0 such that $\beta(y, t) \geq \beta_0$, and $\sigma(y, t) \geq \sigma_0$.

Consider an investor whose total wealth at time t is $X(t)$. We denote $\pi(t, y)$ the proportion of wealth invested in the risky asset at time t and the market state $Y(t) = y$. Then, the wealth equation of the investor's portfolio is as follows

$$\begin{cases} dX(t) = X(t) \left[(r + (b - r)\pi)dt + \pi\sigma\rho dW_0(t) + \pi\sigma\sqrt{1 - \rho^2} dW(t) \right] \\ X(0) = x_0 > 0 \end{cases} \tag{4}$$

For short we drop the independent variables of all coefficients in the above formula.

The objective of the investor is to maximize the expected discounted utility. Namely, the optimal portfolio problem can be stated as the following optimization problem:

$$\begin{cases} \max_{\pi} E^{0, y_0, x_0} \left[\frac{1}{\gamma} \eta(\tau, Y(\tau))(X(\tau))^\gamma \right] \\ s.t. \end{cases} \tag{5}$$

where $0 < \gamma < 1$, $\eta(t, y)$ is nonnegative, continuous, and deterministic; the superscripts to emphasize the initial time, the corresponding market state and the wealth value of conditional expectation.

3 Optimal Portfolio

To solve the optimization problem (5) proposed in previous section, we use the dynamic programming approach. The corresponding value function is given by

$$V(t, x, y) = \max_{\pi} E^{t, x, y} \left[\frac{1}{\gamma} \eta(\tau, Y(\tau))(X(\tau))^\gamma \right] \tag{6}$$

The Hamilton-Jacobi-Bellman equation (HJB) is as follows:

$$\sup_{\pi} \left[V_t + \frac{1}{2} (x^2 \pi^2 \sigma^2 V_{xx} + 2x\pi\sigma\beta_0\rho V_{xy} + \beta_0^2 V_{yy}) + x(r + (b - r)\pi)V_x + \mu_0 V_y \right] = 0 \tag{7}$$

with the initial-boundary conditions:

$$V(T, x, y) = \frac{1}{\gamma} \eta(T, y)x^\gamma, \quad V(t, x, H) = \frac{1}{\gamma} \eta(t, H)x^\gamma, \quad V(t, x, L) = \frac{1}{\gamma} \eta(t, L)x^\gamma$$

To solve the problem we proceed to solve the HJB following the usual procedure. First, assuming $V_{xx} < 0$ the candidate for the optimal control is given by

$$\pi^* = -\frac{(b-r)V_x}{\sigma^2 x V_{xx}} - \frac{\beta_0 \rho V_{xy}}{\sigma x V_{xx}} \tag{8}$$

Second, inserting π^* in the HJB leads to the partial differential equation for V

$$V_t - \frac{1}{2} \frac{(b-r)^2}{\sigma^2} \frac{V_x^2}{V_{xx}} - \frac{1}{2} \beta_0^2 \rho^2 \frac{V_{xy}^2}{V_{xx}} - \frac{(b-r)\beta_0 \rho V_{xy} V_x}{\sigma V_{xx}} + \frac{1}{2} \beta_0^2 V_{yy} + xrV_x + \mu_0 V_y = 0 \tag{9}$$

We apply the separation, and set

$$V(t, x, y) = \frac{1}{\gamma} x^\gamma \cdot [f(t, y)]^k, \quad \text{with } k = \frac{1-\gamma}{1-\gamma(1-\rho^2)},$$

It leads to the following partial differential equations (PDE) initial-boundary problem for f

$$\begin{cases} f_t + \frac{\gamma}{k} \left(r + \frac{1}{2} \frac{1}{1-\gamma} \frac{(b-r)^2}{\sigma^2} \right) f + \left(\mu_0 + \frac{\gamma}{1-\gamma} \frac{(b-r)\beta_0 \rho}{\sigma} \right) f_y + \frac{1}{2} \beta_0^2 f_{yy} = 0 \\ s.t. f(T, y) = \eta^{\frac{1}{k}}(T, y), \quad f(t, H) = \eta^{\frac{1}{k}}(t, H), \quad f(t, L) = \eta^{\frac{1}{k}}(t, L) \end{cases} \tag{10}$$

For the notation simplicity, set

$$\hat{r}(y, t) = -\frac{\gamma}{k} \left(r(y, t) + \frac{1}{2} \frac{1}{1-\gamma} \frac{(b(y, t) - r(y, t))^2}{\sigma(y, t)^2} \right) \tag{11}$$

$$\hat{\mu}(y, t) = \mu(y, t) + \frac{\gamma}{1-\gamma} \frac{(b(y, t) - r(y, t))\beta(y, t)\rho}{\sigma(y, t)} \tag{12}$$

$$\hat{\beta}(y, t) = \beta(y, t), \quad \xi(y, t) = \eta^{\frac{1}{k}}(y, t) \tag{13}$$

Propositions 1. Consider the initial-boundary value problem [16]

$$\begin{cases} f_t - \hat{r}(y, t) f + \hat{\mu}(y, t) f_y + \frac{1}{2} \hat{\beta}^2(y, t) f_{yy} = 0 \\ f(y, T) = \xi(y, T), \quad f(H, t) = \xi(H, t), \quad f(L, t) = \xi(L, t) \end{cases} \tag{14}$$

let $L < Y(t) = y < H$, $T_H^t = \inf\{s, s \geq t, Y(s) = H\}$,

and $T_L^t = \inf\{s, s \geq t, Y(s) = L\}$, $\tilde{\tau}^t = T_H^t \wedge T_L^t$, $\tau^t = \tilde{\tau}^t \wedge T$, then

$$f(y, t) = \hat{E}^{t,y} \left[e^{-\int_t^T \hat{r}(Y(s), s) ds} \xi(Y(\tau^t), \tau^t) \right] \tag{15}$$

is unique solution for the problem.

Where \hat{E} denotes expectation under the measure \hat{P} defined by

$$\frac{d\hat{P}}{dP} = \exp \left(-\frac{1}{2} \int_0^t \left(\frac{\gamma}{1-\gamma} \frac{b(Y(s), s) - r(Y(s), s)}{\sigma(Y(s), s)} \rho \right)^2 ds + \int_0^t \frac{\gamma}{1-\gamma} \frac{b(Y(s), s) - r(Y(s), s)}{\sigma(Y(s), s)} \rho dW_0(s) \right) \tag{16}$$

such that $\hat{W}_0(t) = W_0(t) - \frac{\gamma}{1-\gamma} \int_0^t \frac{b(Y(s), s) - r(Y(s), s)}{\sigma(Y(s), s)} \rho ds$ is a Brownian motion under \hat{P} , and

$$dY(t) = \hat{\mu}(Y(t), t) dt + \hat{\beta}(Y(t), t) d\hat{W}_0(t) \tag{17}$$

Theorem 2. For the optimization problem (5), the value function is

$$V(t, x, y) = \frac{1}{\gamma} x^\gamma \cdot [f(t, y)]^k \tag{18}$$

and the optimal portfolio strategy is given by

$$\pi^* = -\frac{(b-r)V_x}{\sigma^2 x V_{xx}} - \frac{\beta_0 \rho V_{xy}}{\sigma x V_{xx}} = \frac{(b-r)}{\sigma^2 (1-\gamma)} + \frac{\beta_0 \rho}{\sigma} \frac{1}{1-\gamma(1-\rho^2)} \frac{f_y}{f} \tag{19}$$

It is easily seen that the optimal portfolio strategy is not closed-form since f is only an expectation of a stochastic functions. In the following section, we will give closed-form of optimal portfolio strategy for a special case that all coefficients are constants.

4 Special Case

In this section, we assume that the all coefficients of equations are constants in the previous section; and $\eta(\tau, Y(\tau)) = e^{r(T-\tau)}$. Then the market state process follows a Brownian motion with drift $Y(t) = y_0 + \hat{\mu}t + \hat{\beta}\hat{W}_0(t)$ under the measure \hat{P} . In order to obtain the closed-form of optimal portfolio, we first give the following proposition.

Proposition 3. Consider $Y(t) = y_0 + \hat{\mu}t + \hat{\beta}\hat{W}_0(t)$ under the measure \hat{P}

For $s \geq t$, and $L \leq Y(t) = y \leq H$, we have

$$\begin{aligned}
 \hat{P}^{t,y}(T_H^t \wedge T_L^t \in ds) &= \varphi_{t,y}(s) ds \\
 &= \exp\left(-\frac{1}{2}\left(\frac{\mu}{\beta}\right)^2(s-t)\right) \frac{1}{\sqrt{2\pi(s-t)^3}} \sum_{-\infty}^{+\infty} (2n\delta_H + \delta_L) \exp\left(-\frac{(2n\delta_H + \delta_L)^2}{2(s-t)}\right) ds \\
 &\quad + \exp\left(\frac{\mu}{\beta}\delta_H - \frac{1}{2}\left(\frac{\mu}{\beta}\right)^2(s-t)\right) \frac{1}{\sqrt{2\pi(s-t)^3}} \sum_{-\infty}^{+\infty} \left\{((2n+1)\delta_H - \delta_L)\right. \\
 &\quad \cdot \left.\exp\left(-\frac{((2n+1)\delta_H - \delta_L)^2}{2(s-t)}\right)\right\} ds
 \end{aligned} \tag{20}$$

Where $\delta_H = \frac{H-L}{\beta}$, $\delta_L = \frac{y-L}{\beta}$, the proof can be obtained by (8.11 exercise p99 of [17]).

From the above proposition, the closed-form of function $f(t, y)$ is given as follows:

$$f(t, y) = e^{\frac{\hat{r}t + \frac{r}{k}T}{k}} \int_0^T e^{-\left(\frac{\hat{r} + \frac{r}{k}\right)s} \varphi_{t,y}(s) ds + e^{-\hat{r}(T-t)} \int_0^{+\infty} \varphi_{t,y}(s) ds \tag{21}$$

Substituting the above $f(t, y)$ back to (19) yields the closed-form of optimal portfolio.

5 Simulation Results

To illustrate our results we compute the optimal portfolio for different correlation ρ of risky asset and the market state (Figure 1), and the optimal portfolio for different thresholds (Figure 2). The remaining parameters of the model are fixed. Meanwhile, we compare the results with the optimal portfolio (denote π_M^*) of Merton model.

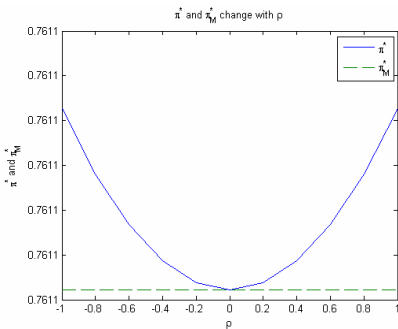


Fig. 1. Optimal portfolio strategy with different ρ for $t = 0, T = 1, \gamma = 0.2, \mu_0 = 0, r = 0.0252, b = 0.08, \sigma = \beta_0 = 0.3, y = 7.6, H = y + \delta, L = y - \delta, \delta = 0.4$

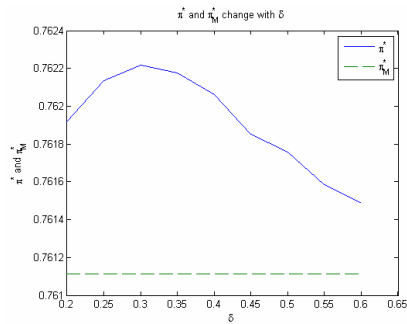


Fig. 2. Optimal portfolio strategy with different δ for $t = 0, T = 1, \gamma = 0.2, \mu_0 = 0, r = 0.0252, b = 0.08, \sigma = \beta_0 = 0.3, \rho = 1, y = 7.6, H = y + \delta, L = y - \delta$

From the simulation results, we get some intuitive interruption. Firstly, the more correlated between the risky asset and the market state (the absolute value of the correlation is greater), the greater different between the optimal portfolio and Merton's. Secondly, the more relax threshold constrained (δ is greater), the smaller different between the optimal portfolio and Merton's.

Acknowledgements. This work was supported by China National Nature Science Foundation under Grant 70671069 and 60574063, and by National Basic Research Program of China (973 Program) under Grant 2007CB814903.

References

1. Merton, R.C.: Lifetime portfolio selection under uncertainty: the continuous-time case. *The Review of Economics and Statistics* LI, 247--257 (1969)
2. Merton, R.C.: Optimal consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory* 3, 373--413(1971)
3. Cox, J.C., and Huang, C.F.: Optimal consumption and portfolio policies when asset prices follow a diffusion process. *J. Economic Theory* 49, 33 --83 (1989)
4. Cox, J.C., Ingersoll, J.E., Ross, S.A.: An intertemporal general equilibrium model of asset prices. *Econometrica* 53, 363 --384 (1985)
5. Duffie, D., Zame, W.: The consumption-based capital asset pricing model. *Econometrica* 57, 1279 --1297 (1989)
6. He, H., Pearson, N.: Consumption and portfolio policies with incomplete markets: the infinite dimensional case. *J. Economic Theory* 54, 259--305 (1991)
7. Karatzas, I., Lakner, P., Lehoczky, J.P., Shreve, S.E.: Equilibrium in a simplified dynamic, stochastic economy with heterogeneous agents. In: Mayer-Wolf, E., Merzbach, E., Schwartz, A. (Eds.), *Stochastic Analysis*. pp. 245--272. Academic Press, New York (1991)
8. Zariphopoulou T.: investment-consumption models with transaction costs and Markov-chain parameters. *SIAM J. Control Optim.* 30, 613--636 (1992)
9. Q. Zhang, Stock trading: An optimal selling rule. *SIAM J. Control Optim.* 40: 64--87(2001)
10. Yin G., Liuand R.H., Q. Zhang: Recursive algorithms for stock Liquidation: A stochastic optimization approach. *SIAM J. Optim.* 13, 240--263 (2002)
11. X. Y. Zhou and Yin G.: Markowitz's mean-variance portfolio selection with regime switching: A continuous-time model. *SIAM J. Control Optim.* 42, 1466--1482 (2003)
12. Richard S.: Optimal consumption, portfolio and life insurance rules for an uncertain lived individual in a continuous time model. *J. Financial Econ.* 1(2), 187--203 (1975)
13. Karatzas I., H. Wang: Utility maximization with discretionary stopping. *SIAM J. Control Optim.* 39, 306--329 (2000)
14. Bouchard B., Pham H.: Wealth-path dependent utility maximization in incomplete markets. *Finance Stoch.* 8, 579--603 (2004)
15. Kraft H., Steffensen M.: Portfolio problems stopping at first hitting time with application to default risk. *Math. Meth. Oper. Res.* 63, 123--150 (2006)
16. Friedman A.: *Stochastic differential equations and applications*. vol 1. Academic press, New York (1975)
17. Karatzas I., Shreve S.E.: *Brownian motion and stochastic calculus*. Springer-Verlag, New York (1991)

Classification Based on Association Rules for Adaptive Web Systems

Saddys Segrera and Mara N. Moreno

University of Salamanca, Department of Computer Science and Automatics, Plaza de la Merced s/n, 37008 Salamanca, Spain
{saddys,mmg}@usal.es
<http://www.usal.es>

Abstract. The main objective of this work is to apply more effective methods than the traditional supervised techniques in the implementation of personalized recommender systems, which improve the accuracy of the predictions in classification tasks. Different model-based classification algorithms based on association rules and others that combine the induction of decision trees with this type of rule were studied. Data from the MovieLens recommender system was used in the analysis and comparison of the different algorithms.

Keywords: Classification, association rules, recommender system.

1 Introduction

Recommender systems are directly related to the personalization of web sites and electronic commerce development. The objective of a recommender system is to offer users, according to their profile, the products or services that could interest them, that is, it anticipates the user's necessities in each new visit to the web site, which makes it an adaptive system.

In this field, our efforts are focused on the construction of efficient classification models in personalized recommender systems. A possible solution for increasing the value of the accuracy is the combination of classifiers. Furthermore, they obtain good results with few training data, even in presence of noise.

In spite of the advantages offered by the multiclassifiers, the works that describe their implementation in the recommender systems are scarce. For this reason, in previous research [16] a comparative study was carried out about multiclassifiers versus traditional classification techniques for application in recommender systems. As a result, we found that multiclassifiers increase the accuracy of the predictions with respect to the individual classifiers, but in turn they increase the computational cost, which is not so important for algorithms that induce the models off-line (model-based), but it is for models that are induced on-line. The execution time of this second type of algorithm directly affects the user response time in recommender systems, and therefore its use is not advisable.

For this reason, memory-based algorithms that are induced on-line have been excluded from this study. The aim of this study was to build different models-based

on association rules to solve a classification problem, so the attributes of the users are related with those corresponding to movies with the objective of emitting recommendations according to the valuations that are made of them. This information is collected starting from the application web MovieLens [11].

Unsupervised techniques such as association rules are difficult to evaluate. However, recent works show that the unsupervised algorithms can be successfully used to solve classification problems [10], [8], [17]. Nowadays new works deal with the integration of supervised and unsupervised techniques to solve some of the problems presented by the classification algorithms [7].

The improvement of the algorithms for generating association rules has been the object of many research works; however the studies about its use in classification problems are few. The association rules algorithms discover patterns in the way $X \Rightarrow Y$. If the consequent part (Y) of the rule is a class, the obtained patterns can be used to predict the class of unclassified records.

The process of rules generation is quite simple [4], the problem is the great number of generated rules, many of which do not have any utility, so it is necessary to evaluate their validity. Most of the methods consider support and confidence to quantify the statistical strength of a pattern. Other methods consider different measures such as unexpectedness, understandability and interestingness [5], [3].

This document has been organized in the following way: in section 2 a comparative analysis between different classification methods based on rules is carried out. The section 3 contains a brief introduction of the algorithms based on association rules that have been used for classification tasks in this study. Section 4 is dedicated to the analysis of the experimental results obtained in the treatment of data from the recommender system with different methods applying the associative classification. Finally, the conclusions are presented in section 5.

2 Rule-Based Classification Methods

A general comparative study of the different classification methods based on rules was done in [7]. According to this research, rule discovery methods are classified into these categories:

- Methods based on covering algorithms. A characteristic method in this category is CPAR [17].
- Methods based on decision tree. An example of this type is C4.5 algorithm [15].
- Association based methods: the association rules were proposed with the objective of solving market basket problems in transactional databases [1], [2]. Some classification methods based on associations are CBA [10] and CMAR [8].
- Association based optimal methods. An algorithm of this type is the mining of the set of optimal class association rules proposed in [9].

The first two types of methods usually yield small sets of rules because the overlapping of sets of satisfied examples is minimized. In contrast, the last two types usually generate large sets of rules because the set of examples satisfied by the rules are highly overlapped. An optimal set of class association rules is significantly smaller than one of association rules, but it is still large. It is not strange for a set of

association rules to be 100 times bigger than an optimal set of rules of class association while an optimal set of rules is 100 times bigger than a rule set based on C4.5.

3 Methods Based Association

Classification based on association, also called associative classification, consists of the application of association rules to classification problems. It is based on restricting the association rules to those containing just the label attribute in the consequent part and other attributes in the antecedent part. This type of rule is called a class association rule (CAR), and they are used in algorithms such as CBA (Classification Based on Associations) [10] for building precise classifiers.

There are two types of models for building association rule based classifiers [7]: Classifiers based on ordered rules and Classifiers based on unordered rules. The first model is simple and effective considering that the second model is not even prudent. The first model carries out a prediction based on the maximum probability. This is because usually the rules with high accuracy precede the rules with the lowest accuracy and the accuracy approaches conditional probability when the data set is large. CMAR uses all the class association rules. However, the problem of the independence of the rules is not addressed in a set of association rules where most of the rules are correlated and their set of satisfied examples is much overlapped.

However, the classifiers based on ordered rules are relatively effective and stable. They do have other problems, however. The small classifiers do not tolerate the inadvertently lost values in the test data and therefore, they are not robust. All the classifiers based on ordered rules use small sets of rules.

An association rule model was proposed in [14] to predict software size. Additionally, an iterative process to refine association rules, which uses the best attributes for classifying, was carried out in order to find suitable rules for prediction.

Association rules have also been applied in knowledge discovery in web mining with data from recommender systems. In a research work on the environment of web usage mining [13], a model was implemented to assist the new clients whose user's profile did not yet form part of the recommender system. The authors applied an algorithm of association rules for finding the initial model. The rating and the user information were used in the rules generation procedure.

The next section describes the methods used in this study for building classification models that could be applied to an adaptive web system for recommending movies.

3.1 Classification Based on Associations (CBA)

The algorithm called CBA [10] (Classification Based on Associations) consists of two parts, a rule generator, which is based on the Apriori algorithm for finding association rules, and a classifier builder.

The candidate rule generator is similar to the Apriori one. The difference is that CBA updates the support value in each step while the Apriori algorithm only updates this value once. This allows us to compute the confidence of the ruleitem.

Next, a classifier is built from CAR. To produce the best classifier out of the whole set of rules would involve evaluating all the possible subsets of it on the training data and selecting the subset with the right rule sequence that gives the least number of errors. There are 2^m such subsets, where m is the number of rules. It is a heuristic algorithm. However, the classifier it builds is better compared to the one built by C4.5.

Given two rules, $r_i (X_i \Rightarrow c_i)$ and $r_j (X_j \Rightarrow c_j)$, r_i precedes r_j , if the number of elements that form the X_i subset is lower than X_j ; if the number of elements is the same in the two subsets, the confidence of r_i is greater than that of r_j , if the confidence values coincide, then the support of r_i is greater than that of r_j ; but also if the support values are the same, in this case, to verify that r_i is generated earlier than r_j . On the other hand, the rule $X_i \Rightarrow c$ is redundant if $Y_i \subset X_i$ and $Y_i \Rightarrow c$.

If R is a set of generated rules and D the training data, the basic idea of the algorithm is to choose a set of high precedence rules in R to cover D . The classifier follows this format: $\langle r_1, r_2, \dots, r_m, \text{default_class} \rangle$, where $r_i \in R$. In classifying an unseen case, the first rule that satisfies the case will classify it. If there is no rule that applies to the case, it takes on the default class as in C4.5.

3.2 Classification Based on Multiple Class-Association Rules - CMAR

As with most algorithms that use association rules in classification tasks, CMAR [8] is divided into two phases, the initial one for the rule generation and a subsequent one for classification. In the first phase, CMAR finds the rules that will participate in the classification step. At the beginning, it mines the training data set to find the complete set of rules passing certain confidence and support thresholds. CMAR adopts a variant of FP-growth [6].

CMAR uses FP-growth instead of Apriori. FP-growth is a frequent pattern mining algorithm. In some cases, FP-growth is faster than Apriori, when there are large data sets, a low support threshold, and/or long patterns [8]. Apriori visits each transaction when generating a new candidate set whereas FP-growth does not work in this way. It can use data structures to reduce the transaction list. In addition, it traces the set of concurrent items whereas Apriori generates candidate sets. As a drawback, FP-growth uses more complex data structures and mining techniques than Apriori.

CMAR scans the training data set T once in order to find the set of attribute values which appear at least as often as the value of minimal definite support. Then, set F is created, which is a list of those attributes that satisfy the previous condition. This set is called a frequent item set. The rest of the attribute values that do not appear in F are deleted and not considered. Afterwards, CMAR shortens attribute values in F in support descending order. The algorithm scans the training data set again to build a frequent pattern tree (FP-tree).

The classification phase based on multiple rules is carried out in this way: if the rules are not consistent in class labels, CMAR divides the rules into groups according to class labels. All the rules in a group share the same class label and each group has a distinct label. CMAR compares the effects of the groups by the combined effect measure of each group. Intuitively, if the rules in a group are highly positively correlated and also have high support values, the group should have a strong combined effect.

There are different ways to measure the combined effect of a group. The best alternative is to integrate either correlation information or popularity. After empirical verification, CMAR uses a weighted χ^2 measure to predict the classes.

3.3 Classification Based on Predictive Association Rules - CPAR

The method of classification based on Predictive Association Rules, known as CPAR [17], is based on the FOIL (First Order Inductive Learner) algorithm proposed in [15]. FOIL learns rules to distinguish positive examples starting from negative examples. Its essence resides in looking for, in a repetitive way, the positive examples of the best rule in course and then all the positive examples that satisfy this rule are eliminated. In a multi-class problem, for each class their examples are positive and those belonging to another class are negative examples. Then the rules of all the classes are joined to form the set of resultant rules.

The authors proposed Predictive Rule Mining (PRM). This after an example is correctly covered by a rule, instead of remove; its weight is decreased by multiplying a factor.

A literal p is an attribute-value pair, taking the form of (A_i, v) , where A_i is an attribute and v is a value. CPAR uses PRM, but it keeps all close-to-the-best literals during the rule building process. Therefore, CPAR can select more than one literal at the same time and build several rules simultaneously.

At a certain step in the process of building a rule, after finding the best literal p , another literal q that has a gain similar to that of p is found. Besides continuing building the rule by appending p to r , q is also appended to the current rule r to create a new rule r' , which is pushed into the queue. Each time a new rule is to be built, the queue is checked first. If it is not empty, a rule is extracted from it and is taken as the current rule.

The classification is made in the following way: given a rule set containing rules for each class, the best k rules of each class for prediction are used with the following procedure. First, select all the rules whose bodies are satisfied by the example; second, from the rules selected previously, choose the best k rules for each class; and third, compare the average expected accuracy to the best k rules of each class and select the class with the highest value as the predicted class.

4 Experimental Results

MovieLens is a recommender system of movies, accessible through the Internet. This data set has already been used by the authors of the present work [16] in the search for effective classification models for recommender systems.

The available data set is characterized as very large, which is a property of the recommender systems. More than a million valuations of movies for 6040 users about 3593 films are registered. It possesses 21 attributes, 19 of them only possess 2 categories (the attribute corresponding to the sex of the users and 18 attributes related to the genre of the movies). The two possible values that can take the 18 attributes of the genre of movies are 1 for describing the ownership of a certain gender and 0 otherwise. On the other hand, the attribute "occupation" possesses 21 categories and the age of the users has been discretized into 7 ranges. Also, the label of the problem,

“ratings”, has 5 possible values that we have summarized into 2 in this work (whether the movie is recommended or not). Nevertheless, a logical way of reducing the possible values of “occupation” and “age” for classification methods by means of association rules without valuable information lost does not exist.

The behaviour of this data set was analyzed for the methods CBA, CMAR, FOIL and CPAR against a classifier of decision trees based on C4.5. In all the cases cross validation with 10 folders was used. The experiments were carried out in a 1.50GHz Intel Centrino PC with 1GB of memory.

The comparative study was carried out with a subset of the database (1000 records) owing to the high amount of available examples. It became necessary to reduce their number to make it possible to execute this type of algorithm, because, as we mentioned previously, the generation of a large amount of rules requires a large amount of memory, especially with the existence of many attributes, two of them with more than three different categories.

For the classification methods by means of association rules the values of minimum support of 20% and minimum confidence of 80% were settled. In the case of CBA the number of attributes in the antecedent part of the rules is high for the MovieLens data; its value is greater than 6 elements. In CMAR the number of frequent sets increased in an excessive way with these data, the figure being 711 697. On the other hand, CPAR is affected by an excessive number of generated rules. The results of the execution of these algorithms are shown in the Table 1.

The first three methods of associative classification (CBA, CMAR and FOIL) diminished the errors with respect to the decision tree classifier, so the accuracies of the predictions were improved. Nevertheless, the execution time significantly increased. FOIL was the only method that reduced the computational cost with respect to the decision tree, with a difference of 6.58 seconds and at the same time, the value of the accuracy increased by 12.45%. CMAR obtained a significant improvement in the prediction accuracy for the valuations of the movies (8.94%), but its execution time, longer than 25 minutes, was the worst of all the methods in the study. The CBA method, although executed in a shorter time than CMAR, doubles the time consumed by the decision tree. The CPAR algorithm was the only method obtaining the highest value in error of all the methods for this data set, including the decision tree; therefore its use was ruled out, since although its time of execution did not reach the second, its capacity for correctly predicting the valuations of the movies was less than 50%.

Better results were expected for CPAR. However, in this study, CPAR showed behaviour contrary to that found with 26 data sets from the UCI Machine Learning Repository in a research work developed by [17], which gave evidence of its superiority over the rest of the associative classification methods.

Table 1. Results of the execution of the decision tree, CBA, CMAR and CPAR

Algorithms	Accuracy (%)	Execution time (seconds)
Decision tree J48	72.34	8.00
CBA	85.94	17.02
CMAR	91.06	1 519.09
FOIL	84.79	1.42
CPAR	49.85	0.18

These methods constitute model-based algorithms, therefore the execution time does not directly influence the user response time in a recommendation system. These models are suitable when the preferences of the users do not change very fast over time, so the built models are applicable during an acceptable time period. For this reason, all can be good candidates for classification tasks in this recommendation system, except CPAR.

5 Conclusions

The application of the classification methods based on association rules for large volumes of data and attributes requires, in general, many computational resources. Only the FOIL method among those studied reduced the computational cost in comparison with the inductor of decision trees for the data set from the MovieLens recommender system and at the same time it increased the value of the accuracy.

CPAR was the fastest, but it presented the lowest percentage in examples correctly evaluated for this study case.

The CBA, CMAR and FOIL methods studied achieved high values of accuracy in this study and they overcame the decision tree classifier. Any one of these methods could be proposed for the construction of classification models in a recommender system.

In future works, we will consider a new algorithm that use classifiers based on ordered rules and interestingness measures different to support and confidence, which allow to reduce the number of candidate rules.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. Proceedings of the ACM SIGMOD, Conference on Management of Data, Washington, D.C., May (1993) 207-216
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Databases (VLDB94), Santiago de Chile (1994) 487-489
3. Berzal, F., Cubero, J.C., Marín, N., Serrano, J.M., Sánchez, D., Vila, A.: Association rule evaluation for classification purposes. I Congreso Español de Minería de Datos (CEDI'05), Actas del III Taller de Minería de Datos y Aprendizaje (TAMIDA '05), Granada. Thomson (2005) 135-144
4. Cabena, P., Hadjinian, P., Stadler, R. Verhees, J. and Zanasi, A.: Discovering Data Mining from concept to implementation. Prentice Hall (1998)
5. Ghosh, A., Nath, B.: Multi-objective rule mining using genetic algorithms. Information Sciences, Vol. 163 (2004) 123-133
6. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. Proceedings of the ACM SIGMOD International Conference on Management of Data SIGMOD'2000, Paper ID: 196 (2000)
7. Hu, H., Li, J.: Using Association Rules to Make Rule-based Classifiers Robust. Proceedings of the Sixteenth Australasian Database Conference, ADC 2005, Newcastle, Australia, January 31st - February 3rd (2005) 47-54

8. Li, W., Han, J., Pei, J.: CMAR: accurate and efficient classification based on multiple class-association rules. Proceedings of the IEEE International Conference on Data Mining, (ICDM '01), California (2001) 369-376
9. Li, J., Shen, H., Topor, R.: Mining the optimal class association rule set. Knowledge-Based System, Vol. 15, 7 (2002) 399-405
10. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. Proceedings of the 4th International Conference Knowledge Discovery and Data Mining (KDD-98). AAAI Press (1998) 80-86
11. Miller, B., Riedl, J., and Konstan, J.: Experiences with GroupLens: Making Usenet useful again. Proceedings of the 1997 Usenix Winter Technical Conference, January (1997)
12. Molina, L.C., Bñar, J.: Integraci3n de reglas de asociaci3n y de clasificaci3n. Reporte T3cnico, Departamento de Lenguajes y Sistemas Inteligentes, Universidad Polit3cnica de Cataluã, noviembre (1999)
13. Moreno, M.N., Garc3a, F.J., Polo, M.J., L3pez, V.: Using Association Analysis of Web Data in Recommender Systems. Lectures Notes in Computer Science, LNCS 3182 (2004) 11-20
14. Moreno, M.N., Miguel, L.A., Garc3a, F.J., Polo, M.J.: Building Knowledge Discovery-Driven Models for Decision Support in Project Management. Decision Support Systems, 38 (2004) 305-317
15. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA (1993)
16. Segrera, S., Moreno, M.N.: Application of Multiclassifiers in Web Mining for a Recommender System. WSEAS Transactions on Information Science and Applications, Vol. 3, 12, ISSN 1790-0832, December (2006) 2471-2476
17. Yin, X., Han, J.: CPAR: Classification based on Predictive Association Rules. Proceedings of the 2003 SIAM International Conference on Data Mining (SDM'03), San Francisco, CA, May (2003)

Statistical Selection of Relevant Features to Classify Random, Scale Free and Exponential Networks*

Laura Cruz Reyes¹, Eustorgio Meza Conde², Tania Turrubiates López^{1,3},
Claudia Guadalupe Gómez Santillán^{1,2}, and Rogelio Ortega Izaguirre²

¹ Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada (CICATA). Carretera Tampico-Puerto Industrial Altamira, Km. 14.5. Altamira, Tamaulipas, México. Teléfono: 01 833 2600124

² Instituto Tecnológico de Ciudad Madero (ITCM). 1ro. de Mayo y Sor Juana I. de la Cruz s/n CP. 89440, Tamaulipas, México. Teléfono: 01 833 3574820 Ext. 3024

³ Instituto Tecnológico Superior de Álamo Temapache (ITSAT). Km. 6.5 Carreteta Potrero de Llano – Tuxpan, C.P 92750. Alamo, Veracruz, México. Teléfono: 01 7568440038
lauracruzreyes@hotmail.com, emezac@ipn.mx, tania_251179@acm.org, cgs71@hotmail.com, rortegai@ipn.mx

Abstract. In this paper a statistical selection of relevant features is presented. An experiment was designed to select relevant and not redundant features or characterization functions, which allow quantitatively discriminating among different types of complex networks. As well there exist researchers given to the task of classifying some networks of the real world through characterization functions inside a type of complex network, they do not give enough evidences of detailed analysis of the functions that allow to determine if all are necessary to carry out an efficient discrimination or which are better functions for discriminating. Our results show that with a reduced number of characterization functions such as the shortest path length, standard deviation of the degree, and local efficiency of the network can discriminate efficiently among the types of complex networks treated here.

Keywords: Complex Networks, Internet Modeling, Classification, Variable Selection, Experimental Design.

1 Introduction

Any natural or artificial complex system of the real world as Internet, can be modeled as a complex network, where the nodes are elements of the system and the edges interactions between elements [1]. The term complex network refers to a graph with a non trivial topologic structure, this has motivated the study of topological characteristics of real networks [2] to identify characteristics that allow the discrimination among different types of complex networks and in this way optimize the performance of processes carried out in this networks as: search of distributed resources [3], traffic management and design of routing algorithms.

* This research was supported in part by CONACYT and DGEST.

Up to now the best way to identify the type of complex network has been observing the graphic of the degree distribution [2]. This work presents a statistical selection of a set of characterization functions that allow quantitatively identified the type of complex network.

1.1 Characterization Functions of Complex Networks

The characterization functions provide information about the topological characteristics of a complex network, when they are applied on a network, a vector of characteristic is obtained [4]. Through this vector, the behavior of the complex networks is characterized and analyzed. The characterization functions used in this work were: the average degree of the network, standard deviation of the degree, the shortest path length, diameter, clustering coefficient, global and local efficiency, this characterization functions are described and detailed in [4].

One characterization function widely used to identify the type of complex network is the degree distribution which provides graphic information about the connectivity behavior of the nodes; through this function different types of networks as random networks, scale-free networks and exponential networks can be identified.

1.2 Three Types of Complex Networks

Toward late 50's and early 60's Erdős and Rényi focused in studying statistical properties in graphs where the connection among the nodes is established randomly. These graphs are called random networks and they are characterized for approaching a degree distribution binomial when the number of nodes n is small; when $n \rightarrow \infty$, the degree distribution approaches a Poisson distribution [5].

In the decade of the 90's diverse researchers [6], [7], discovered networks of the real world as Internet exhibits a degree distribution following a power law distribution. These networks are called scale-free, because independently of the scale, number of nodes, the main characteristic of the network does not change, a reduced set of nodes have a very high degree and the rest of the nodes have a small degree [8].

Some natural networks exhibit an exponential distribution [9], [10], where the majority of the nodes have a degree closer to the average degree and other nodes with a high degree can be observed [6], these networks are called exponential networks. To analyze the characteristics of these networks and to understand the phenomena carried out in them, generation models of complex networks have been created.

1.3 Generation Models of Complex Networks

The generation models of networks are an important tool to reproduce graphs sharing topological characteristics of networks of the real world making possible its study [8]. Some generation models are the Erdős-Rényi (ER) model [5] that reproduces random networks, the Barabási-Albert (BA) model [11] that reproduces scale free networks and the Liu model [12] that reproduces scale free and exponential networks.

2 Related Works

A recent application in the field of complex networks, is used information obtained through the characterization functions with the objective to discriminate among different types of complex networks, this application is related with the area of pattern recognition also known as classification. The classification of natural and artificial structures modeled as complex networks implicate an important question: what characterization functions to select in order to discriminate among different types of complex networks [4].

In the classification area the selection of features (in this case, characterization functions) has great benefits: to improve the performance of classification procedure, and to construct simple and comprehensible classification models, these are achieved leaving aside, irrelevant and redundant characteristics that can introduce noise [15], [16].

The work reported in [4], used a classification procedure to identify the type of a network with unknown nature, the results show that the type of network assigned to the networks, vary according to the characterization functions selected and an excessive number of characterization functions can compromise the quality of the classification.

An experimentation to evaluate the stability and separability of different types of networks is presented in [17], taking into account 6 topologies of networks to discriminate, 47 characterization functions are used as inputs to the classifiers. This work does not present evidences of a selection of relevant and not redundant characterization functions.

In [18], the objective is to determine from a set of models those describing more adequately networks that represent biological systems, a technique described in [19] was utilized for extracting characteristics serving as inputs to the classifier permitting to relate an instance generated by a model with a real instance. Though the results of classification obtained are good, they do not show if the characteristics extracted by the technique improve the performance of the classifier regarding the information provided by characterization functions extensively used in the field of the physics and the social sciences. In [20] networks in 3 types of topology are classified by means of a neural network; the eigenvalues of the adjacency matrix are utilized as inputs of the neural network.

3 Experimentation

Following a general scheme of procedures described in [13] an experiment was designed, to determinate, given a set of characterization functions, the functions permitting quantitatively discriminate among three types of complex networks: random networks, scale free networks and exponential networks.

In other words, three different populations are defined for each type of networks, from which topological characteristics are extracted, if the averages of those characteristics are significantly different and the populations do not overlap then those characteristics can help to distinguish networks of those populations. Three possible results of classifying networks, can be observed:

- Case 1: There are significant differences among the 3 types of networks according to a set of characteristics, in this way through these characteristics a new network can be classified inside a type of network.
- Case 2: There exist significant differences among some of the types of networks according to a set of characteristics.
- Case 3: There does not exist significant differences among the 3 types of networks according to a set of characteristics, these two last cases lead to wrong classifications of new networks that need to be identified.

The methodology to select the characterization functions was:

- Step 1: Determining statistically which characterization functions differ significantly according to the type of network..
- Step 2: Detecting from which type of networks differ the characterization functions in order to identify the cases above mentioned.
- Step 3: Confirming the observations carried out in the Step 2, performing a characterization functions clustering.
- Step 4: Selecting from the Case 1 and Case 2 the characterization functions not correlated.

Instances of complex networks were generated to carry out the experimentation, through the models of E-R (random networks), BA (scale free networks) and Liu (exponential networks), with 200, 512 and 1024 nodes by each type of network; subsequently topologic characteristics were extracted such as: average degree (Avg), standard deviation of the degree (Std), clustering coefficient (CG), global efficiency (EG), local efficiency (EL), the shortest path length (L), diameter (D). The statistical packages MINITAB and SAS were utilized to study these characteristics.

In the related works, the networks have the same number of nodes and edges, so the average degree is equal for each type of network; in this experimentation the number of edges with the purpose of introducing variability to the experiment and carry to correct conclusions is not determined. In [17] it was observed that determining the number of edges aid to understand phenomena of interest but introduces dangers in statistical tests.

In this way, two factors that can influence in the process of characterization were identified; these factors can be controlled without affecting data normality, the type of network and the number of nodes. The factor, type of network, is composed of 3 levels, represented by the 3 types of networks being analyzed. The factor, number of nodes, is also composed of 3 levels, 200, 512 and 1024 nodes. Giving the characteristic of the problem a two-factor factorial design was chosen. The significance level α was set in 0.05.

The operation characteristics curves was used to determine the number of instances of networks n , appropriate for detecting significant differences. Through this procedure it was determined that with $n = 4$ instances a probability of 94% is obtained to detect differences up to 0.01 this procedure can be found in [13]. Due that this experiment was extended to the classification, $n = 4$ was taken as the minimum number of instances to generate; $n = 30$ instances was set for the experimental design.

Once the networks were generated and characterized, a Principal Components Analysis (PCA) was performed to verify the normality of the data. According to the

fixed effects model describing to a two-factor factorial design [13] a hypothesis were formulated to detect significant differences. A Multivariate Analysis of Variance General (GML) was carried out to obtain results permitting to reject or to accept the hypothesis, these results are discussed subsequently.

Table 1. F_0 -values calculated by GLM for each variable

	Avg	Std	CG	EG	EL	L	D
Type of network	178.47	124.39	220.18	507.82	55.21	566.56	523.72
Number of nodes	32.80	50.45	1.70	10.50	1.07	39.11	21.07
Interaction	32.70	20.87	0.56	3.57	0.18	13.90	10.01

The values showed in Table 1, were compared with the criteria for rejection associate to the hypothesis formulated, it can be observed that according to the type of network, the characterization functions satisfy the criteria for rejection, therefore, it can be concluded that all the characterization functions differ significantly according to the type of network.

For the effect of the number of nodes, and the effect of the interaction between the type of network and number of nodes, the criteria for rejection are satisfied for Avg, Std, EG, L and D, therefore, it can be concluded that these characterization functions differ significantly according to the number of nodes presented in the network, and the interaction between the type of network and the amount of nodes.

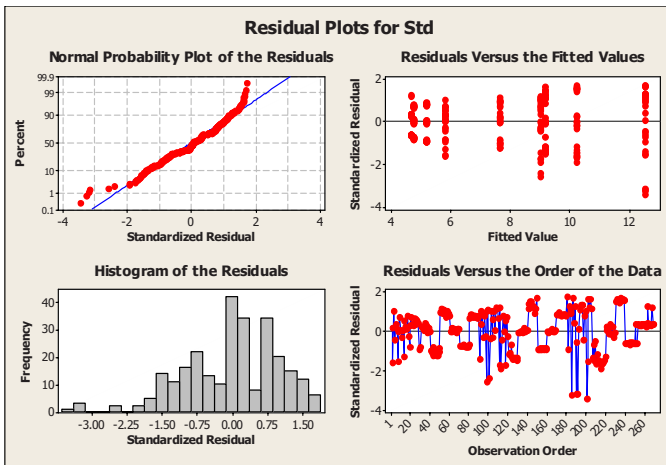


Fig. 1. Plot of residuals for Std

The analysis of the residuals of Avg and CG, showed abnormalities, this is because in spite of the fact that the number of edges was not determined, Avg and CG for scale free and exponential networks, were defined by initial parameters employed by the models of generation, on the other hand the plots of Std, EL, EG, L, D showed normality. In Fig. 1 the plot of Std is shown as an example of normality.

MANOVA tests shown in Table 2, for the type of network, the quantity of nodes and the interaction, are statistically significant, by which the significant differences detected in the analysis of a variable at one time are reaffirmed, being real and not false positive. Due that the values of F_0 are greater for factor of type of network, it is concluded that this factor has greater influence in the values obtained by the characterization functions.

Table 2. F_0 values calculated for the MANOVA test and values of F distribution

MANOVA Test	Type of Network		Number of nodes		Interaction	
	F_0	F_{α, v_1, v_2}	F_0	F_{α, v_1, v_2}	F_0	F_{α, v_1, v_2}
Wilks'	515.91	1.23	88.62	1.23	32.97	1.17
Pilliai's	391.57	1.23	34.52	1.23	17.04	1.17
Lawley – Hotelling	676.26	1.23	183.88	1.23	66.26	1.17
Roy's	1137.97	1.29	367.89	1.29	250.28	1.29

Once it was detected that the characterization functions differ significantly according to the type of network and this factor has greater effect, using the test of Tukey [13], multiple comparisons were carried out, with the objective of detecting which characterization functions differ significantly in each level of the factor of type of network.

Tukey's tests carried out showed the means of the characterization functions Avg, EL, EG and CG are significantly equal for scale free and exponential networks, and for the random networks the mean is significantly greater. For the function Std, the means of the scale free and random networks are significantly equal and for the exponential networks it is significantly smaller. The characterizations functions where significant differences in the means of the random, scale free and exponential networks were observed are L and D.

From the information obtained through the cluster variables, it can be observed (Fig. 2) that the characterization functions form 3 groups, the first group (Avg, EL, EG, CG) differentiates the random networks from scale free and exponential networks, the second (Std) differentiates the exponential networks from scale free and random networks, these two groups are representative of the Case 2, the third (L, D) manages to differentiate among the 3 types of networks and represents the Case 1.

The characterization functions selected to discriminate among different types of networks, were L, Std and EL, in this way a characterization function of each group representing the cases 1 and 2 were taken into account. The Avg and CG were not taken into account because of the abnormalities presented in the plots of the residuals, thus D and EG also were excluded because of being highly correlated with L, which indicates redundancy.

To obtain an efficient classifier of networks, a quadratic discriminant analysis [14] was carried out with different combinations of the characterization functions selected, using in both cases the cross-validation procedure. In Table 3, the percentages of networks classified correctly inside each type of network, are shown and the total percentage of correct classifications.

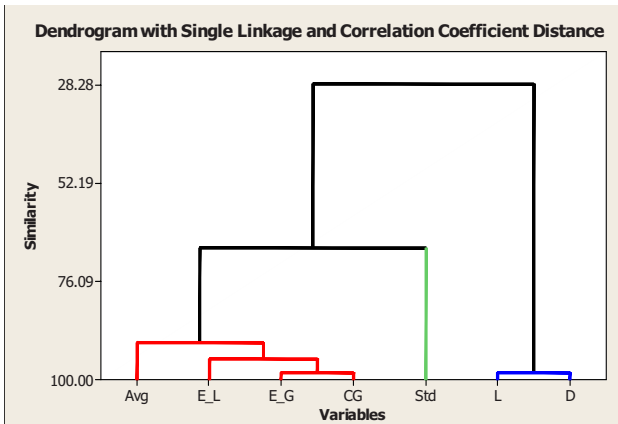


Fig. 2. Dendrogram showing the groups formed by the characterization functions

Table 3. Results of the Quadratic Discriminant Analysis

Quadratic Discriminant Analysis				
Combination of variables	Random Networks	Scale Free Networks	Exponential Networks	Total
L	97.8% (88 / 90)	78.9% (71 / 90)	50.0% (45 / 90)	75.6%
L, Std	97.8% (88 / 90)	88.9% (79 / 90)	95.6% (84 / 90)	94.1%
L, EL	97.8% (88 / 90)	72.2% (63 / 90)	66.7% (60 / 90)	78.9%
L, Std, EL	98.9% (89 / 90)	100% (90 / 90)	100% (90 / 90)	99.6%

4 Conclusions and Future Works

An experimental design was presented for the purpose of selecting statistically characterization functions of complex networks relevant and not redundant, the results obtained in the experimentation show that the shortest path length, the standard deviation and the local efficiency, permit in a quantitative way to discriminate among random networks, scale free networks and exponential networks. By means of the quadratic discriminant analysis an accuracy of classification of the 99.6% was obtained, the size of the sample was justified statistically. As it can be appreciated in this work the number of characterization functions is very small in relation to the 47 functions utilized in [17], this reduces dramatically the time of computation required for the classification of complex networks.

In future works it is recommended to extend this experimental design for networks with greater amount of nodes and other types of networks, besides it is recommended to work with procedures and algorithms for selection of characteristics utilized in the area of machine learning in order to compare the results obtained with this experimental design.

References

1. Barabási, A.L. Albert R. Jeong, H. "Mean-Field theory for scale-free random networks". *Physica A*, vol. 272, pp 173-189, 1999.
2. Albert, R. Barabási, A.L. "Statistical Mechanics of Complex Networks". *Reviews of Modern Physics*, vol. 74, pp 47-97, 2002.
3. Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: "Search in power law network". *Physical Review E*, vol. 64, pp 046135-1 - 046135-8, 2001.
4. Costa, L., Rodrigues, F.A., Travieso, G., Villas, P.R. "Characterization of Complex Networks: A survey of measurements." http://arxiv.org/PS_cache/cond-mat/pdf/0505/0505185v5.pdf (2007)
5. Bollobás, B. Riordan, O.M.: *Mathematical results on scale-free random graphs*. *Handbook of Graphs and Networks*. Wiley-VCH. Berlin. (2002). 1-32.
6. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. *Nature*. Vol. 506. (2000). 5234-5237.
7. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationship on the internet topology. *ACM SIGCOMM*. Vol. 29, No. 4. (1999). 251-262.
8. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review*. Vol. 45, No. 2. (2003). 167-256.
9. Amaral, L.A.N., Scala, A., Barthelemy, M., Stanley, H.E.: Classes of small world networks. *PNAS*. Vol. 97, No. 21. (2000) 11149-11152.
10. Sen, P., Dasgupta, S., Chatterjee, A., Sreeram, P.A., Mukherjee, G., Manna, S. S.: Small-world properties of the Indian Railway network. *Physical Review E*. Vol. 67. (2003).
11. Barabási, A.L. Albert, R.: Emergence of Scaling in Random Networks. *Science*. (1999). 509-512
12. Liu, Z., Lai, Y., Ye, N., Dasgupta, P.: Connectivity distribution and attack tolerance of general networks with both preferential and random attachments. *Physics Letters A*. Vol. 303. (2003). 337- 344.
13. Montgomery, D.C.: *Diseño y Análisis de Experimentos*. Limusa Wiley. (2004)
14. Jonhson, D.E.: *Métodos multivariados aplicados al análisis de datos*. International Thomson Editores. (2000)
15. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*. Vol. 3. (2003) 1157 - 1182.
16. Singhi, S.K., Liu, H.: Feature Subset Selection Bias for Classification Learning. *Proceedings of the 23rd ICML*. *ACM ICPS*. Vol. 148. (2006). 849 – 856.
17. Airoidi, E.M., Carley, K.M.: Sampling algorithms for pure network topologies: a study on the stability and the separability of metric embeddings. *ACM SIGKDD Explorations Newsletter*. Vol. 7, No. 2. (2005). 13 - 22.
18. Middendorf, M., Ziv, E., Adams, C., Hom, J., Koytcheff, R., Levovitz, C., Woods G., Chen, L., Wiggins, C.: Discriminative Topological Features Reveal Biological Network Mechanisms. *BMC Bioinformatics* 2004. Vol 5, No. 181 (2004).
19. Ziv, E., Koytcheff, R., Middendorf, M., Wiggins, C.: Systematic Identification of statistically significant network measures. http://arxiv.org/PS_cache/cond-mat/pdf/0306/0306610v3.pdf (2005)
20. Ali, W., Mondragón, R.J., Alavi, F.: Extraction of topological features from communication network topological patterns using self-organizing feature maps.

Open Partner Grid Service Architecture in eBusiness

Hao Gui and Hao Fan

International School of Software, Wuhan University, China
issgui@gmail.com, hfan@iss.whu.edu.cn

Abstract. With the changing demands of the markets and the developments of grid technology, grid computing is beginning to spill out over the boundaries of individual organizations. Partner grids create a collaboration environment to share computing resources, data and applications. OPGSA is a grid-based implementation of ebusiness solutions built upon commercial enterprise grid systems, and it makes use of Globus Toolkits 4 and DRMAA open standards. With this architecture, it is possible for the firms participating in a corporate supply chain to incorporate greater management capabilities and gain greater control over complex business processes.

Keywords: Open Partner Grid Service Architecture, e-business, grid computing, partner grid, Globus, DRMAA.

1 Introduction

Grid computing is to make distributed resources into a virtual supercomputer. In fact, it is a set of technologies enabling resource virtualization and bringing the world towards the ubiquitous market of e-services, which consists of a broad range of electronic services and applications that have the potential of interacting, including Web services, telecom services, services on all kinds of mobile devices, and so on.

To achieve sustainable business growth, companies are constantly expecting their information technology systems to deliver business process sooner and make them available to a wider audience. At the same time, they also want to employ emerging technologies such as provisioning, orchestration and virtualization and new standards for interoperability to improve their systems' resilience and make more efficient use of human, technology and capital resources.

Grid computing started in the scientific community to solve long-running, complex computationally intensive tasks. Such grid implementation is known as Computational Grid. Along with the maturity and development of related technologies and standardization around it, grid computing holds more and more attentions even in some industrial and business scenarios. When it comes to the participants and beneficiaries of a given grid implementation, there is another kind of grid, Enterprise Grid, these systems are created in order to solve certain business applications within a given enterprise[1].

With the standardization and open-source development efforts in almost every aspect, grid computing will eventually form the cornerstone of ebusiness infrastructures and allow smooth transactional communication at manageable and reduced cost. Typically these business tasks do not have a long-running nature (opposite to the Computational Grid tasks) but rather are used to shorten the execution time of a given task within acceptable levels and make better utilization of existing I/T resources[2].

When implementing an Enterprise Grid in eBusiness scenarios, it is possible to carry out computations and profitable businesses that are previously unavailable to the enterprises, just because they could not be solved using the existing hardware and infrastructure. By making use of collaborative execution environments and employing distributed grid-based storage architectures, i.e., interconnecting several associated Enterprise Grids together, it is possible to provide reasonable solutions for optimizing end-to-end order management processes for multifaceted, trade-oriented businesses.

We will briefly look at the different forms that today's grids have taken, and illustrate the characteristics of the partner grids, which are considered as the next wave of grid adoption. We will introduce our Open Partner Grid Service Architecture (OPGSA) in detail, which is a grid-based implementation of ebusiness solutions built upon commercial enterprise grid systems. And the collaborative interaction process and tasks scheduling model will be discussed as well.

2 Partner Grids in ebusiness

From centralized mainframes to clusters which are composed of a group of machines all serving an identical purpose, eventually evolving into open, scalable grid systems to address the increasing need for flexibility. Scaling out methodology makes it possible for enterprises to achieve more with less. As other things, grid technologies never stop evolving.

2.1 Business Perspective

Grid solutions exist to satisfy business needs, as shown in Fig 1. Typical requirements that can be satisfied by different types of Grid solutions include[3]:

- More compute power to run business applications.
- Affordable analyses performed more accurately and more often.
- Results returned more quickly—when they are most valuable and relevant.
- Fast access to all necessary information, irrespective of where it is in an organization.
- A better way for analysts to use an existing IT infrastructure.
- Compute power available just when it is needed and users charged for their fair shares.
- No boundaries between various computers and information.

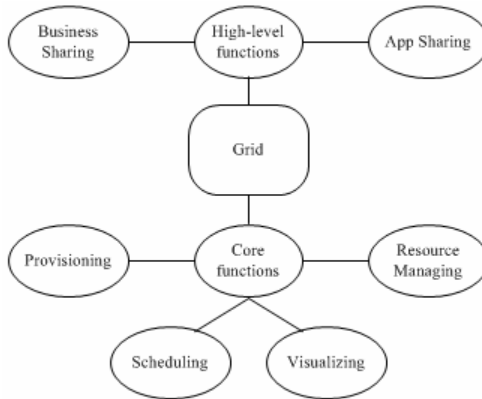


Fig. 1. Grid Solution

2.2 Moving to Partner Grids

Enterprise Grid can improve application service levels, better manage enterprise systems, and reduce the cost of its IT systems. Available resources within a company are better exploited and the administrative overhead is minimized by using Grid technology. Currently, Enterprise grids are the primary application of grid technology[4]. And this will remain the case for quite a few years, but the trend toward partner grids and service grids (also referred to as Utility Grid) is already developing. And the latter two are considered as the next wave of grid technology which built upon Enterprise Grid. These two kinds of grid framework distinguish themselves from each other in architectures and functionalities.

Grid computing offers access to significant computing and data resources on demand. Scientists plan to make services provided by grids into a public utility, just like power and water. With the emergence of the utility grid computing paradigm, it has become increasingly desirable for organizations worldwide to improve efficiency by outsourcing their business processes.

When talking about ebusiness, the efficient proliferation of information, the ever-increasing scale of operations and the collaborative business processing are required[5]. As a result, more trading partners, demands a more joint and efficient platform for getting the job done, for B2B and even B2B2B operations. Partner Grid shares resources among multiple organizations. It can be used to link associated organizations and supply chain participants. For instance, in the automotive manufacturing chain, the suppliers to a motor manufacturer have their own downstream vendors, and they need to collaborate with both partners and vice versa. So they will build application-centric collaboration networks with an underlying Grid infrastructure to facilitate seamless interoperation between different sites and computing environments. These organizations can extend their Enterprise Grids to trusted partners, and form a larger pool of resources.

Partner Grid is a part of the computing paradigm shift, which also includes Web services, on-demand, virtualization. It represents a significant business opportunity

for ebusiness, and it makes an enterprise whose business processes -- integrated end-to-end across the company and with key partners, suppliers, and customers -- can respond with speed to any customer demand, market opportunity, or external threat. Partner Grids allow participants to pass workload through an extended resource-sharing environment, boosting high performance processing capabilities beyond what any one partner would be able to achieve individually. After extending the Enterprise Grids into a Partner Grid, there will be seamless integration between a producer and its suppliers. This not only makes supply chains easier to manage, but also creates an opportunity to establish standards across firms. It improves the quality of products across the supply chain because a manufacturer can help its upstream and downstream suppliers to streamline the interactions and maintain a high level of performance.

The adoption of a partner grid based architecture provides businesses with the ability to rapidly deploy new applications and easily integrate with other component applications both inside and outside the organization. This kind of decentralized application environment can enhance fair competition, lower prices of services, facilitate the supply chain management dramatically.

2.3 Considerations in the Partner Grid Solutions

Partner grid is a natural extension for Enterprise Grid, from intranet-working for private resource sharing within a single enterprise to extranet-working to enable resource sharing among selected enterprise partners. Partner grids always connect these resources beyond the boundaries of either enterprise grid, crossing corporate firewalls. In the meanwhile, some inevitable obstacles will be posed in the implementations of partner grids.

In principle, an enterprise grid is a collection of interconnected grid components under the control of a grid management entity. Task scheduling and resources provisioning only involve one organization. On the other hand, partner grid infrastructures aim to provide large-scale, secure and reliable sharing of resources among partner organizations. It will definitely cross multiple administration domains, as a result, we can not deploy of centralized task schedulers to totally control over client requests and resource status. Therefore, task scheduling infrastructures have to be redesign, and the existing organization level task schedulers should provide support for multiple intra-organization users.

Another key requirement for the success of the services like this is safeguarding the security of data on these grids. Because partner grid is just like a business community, it may bring several firms with rivalry together in order to make profitable commercial decisions. There will be some information which cannot be shared with others because of conflicting commercial interests, maybe minimum prices in a bidding documents or something else. People are not going to share their data unless they have absolute confidence in security and privacy of the environment.

There are other requirements as well, such as rapid provisioning and automated transaction processing. In order to maintain favorable business relationship, it has to be provisioned nearly immediately. In addition, there are some other issues beyond actual technologies, such as legal affairs.

3 Open Partner Grid Service Architecture

Our partner grid architecture is built upon Globus Toolkit 4 (WSRF) and Sun Grid Engine 6. WSRF[6] defines conventions for managing state so that applications can reliably share changing information with greater interoperability between heterogeneous organization infrastructure. Sun Grid Engine characterizes itself with cross-platform supports, including Windows and others OS. This kind of combination provides important services for Partner Grids, and the architecture is a natural fit for the market that involves extensive business collaboration and sharing.

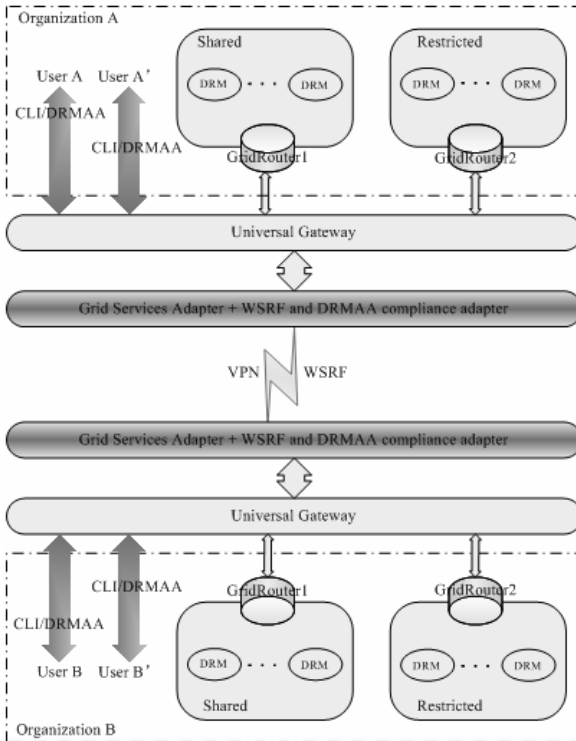


Fig. 2. Open Partner Grid Service Architecture

Fig 2 shows the arrangement aiming to let two or more organizations handle demands for processing power and collaborative businesses while still keeping their computers in-house.

In this architecture, the local resources of individual organization are managed by Distributed Resource Manager (DRM) that queues, dispatches, and controls jobs in the grid. But it only supports command line interface which will hamper the adoption of grid technology to industries. DRMAA has been employed, and it is a Global Grid

Forum (GGF) API specification for the submission and control of jobs to one or more Distributed Resource Management (DRM) systems. It offers a standardized API for application integration with C, Java, and Perl bindings, and it is one of the first specifications to enable the development and deployment of high performance grid applications. Providing "write-once" capabilities to any DRM system that supports DRMAA, the specification makes it possible for new applications -- both enterprise and technical -- to be used in a grid environment, thereby dramatically expanding the reach of grid computing.

In each organization, the resources have been partitioned into Shared and Restricted area, as shown in the figure. Access to these resources is controlled by internal Grid Routers. For example, the Shared and Restricted resources are separated from each other and both separated by an additional security wall, Universal Gateway. Universal Gateway has played two roles in the architecture described above. On the one hand, it belongs to a publish/subscribe realm with the internal Grid Routers, the changes to the resources in the specific area need to be propagated to the others. Universal Gateway has the full knowledge of the organization level resources infrastructure, and it will direct the grid computing tasks from internal to the appropriate area. On the other hand, it is a security guard for the whole organization, responsible for forwarding all incoming messages to the internal Grid Routers. Messages that without proper destinations and security credentials are rejected here.

Grid Services Adapter is the portal for both sides of the enterprise grid. It has a WS-MDS information registry to provide the information about organization level resources collected from internal Grid Engine. This information can be used by other partners in a variety of ways, e.g. by DRMAA interface. It also has a interface that translates WS-GRAM API calls from partners to Grid engine commands invocations and DRMAA C API. So it is a key component that links the rest of the partner community and the organization itself. Actually, WSRF and DRMAA compliance adapters are grid middlewares, allowing the simultaneously use of Globus and Grid Engine services. These are the core standards from GGF ensuring interoperability with current and future Grid offerings from other vendors.

Although there are no commercial partner grids in operation, the Open Partner Grid Service Architecture presented here provides an efficient way for grid services for both inside and outside requests to access organization level systems and applications. And these benefits allow a gradual migration from Enterprise grids to Partner grids, and even, the long-term coexistence of both. This architecture gives us a framework for the collaborative processes involving many partner firms with common goals in business-to-business scenarios.

4 Task Scheduling in Open Partner Grid Service Architecture

To demonstrate the process of how a grid tasks involving different partners is executed in Open Partner Grid Service Architecture more clearly, let's decompose the task scheduling process and look inside the interaction that takes place between these organizations.

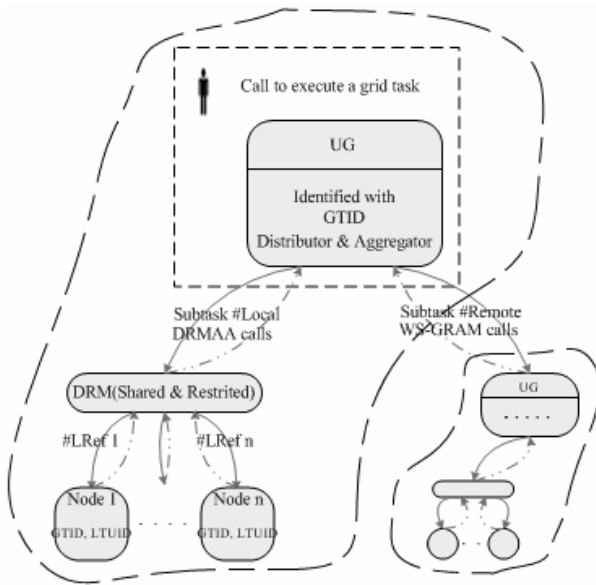


Fig. 3. Task scheduling in OPGSA

Each grid task submitted by an end user is identified with a Global Task ID (GTID), which is unique across the partner grid, and this ID is used to split task into subtasks and aggregate them back. As show in Fig 3, each grid task has an corresponding tree structure of task units. Leaf nodes in the tree structure are task units, each represents a unit of work that can be distributed through out the partner grid for execution.

At the start point of the execution cycle, it is up to the Universal Gateway to determine if it needs to be split and processed in sub units, or needs to be forwarded to the partners in the community. This process repeats itself recursively on all involved partners until each part of the global task are allocated. The WS-MDS information registry in Grid Services Adapter can help the organization to route the subtasks to appropriate service provider in the community. When the final result is fully aggregated back to original grid node and returned.

Grid Services Adapter in this architecture plays an important role in the task scheduling. It holds the information about organization level resources collected from internal Grid Engine and the information published by other partner’s GSA used to describe their services. The main feature of this task scheduling model is the use of the WS-GRAM calls to recursively interface to the services available in another Globus based grid. WS-GRAM calls provide the standard functionality required to implement a gateway to a federated grid. Combined with DRMAA, this model allows the required virtualization technology to be created in order to provide a powerful abstraction of the underlying grid resource management services. With this model, it is possible to make use of virtualization of application resources involves publishing application components as services for use by multiple consumers.

In addition, web services and the semantic web can be used as links between grids, they are likely to add greater intelligence and sophistication to computations that rely upon grids.

5 Conclusion

Grid computing has the potential to revolutionize the world of information technology, much in the same way as the Internet completely transformed the way people and businesses communicate and share information.

Partner Grid is an extension of Enterprise Grid that enable transparent, secure and coordinated resource sharing and collaboration among partner organizations in business. According to the analysts, about half of the firms in autos, computers, semiconductors and other industries would have Partner Grids by 2009, linking them with their closely suppliers and crossing corporate communities. This demonstrates a real commercial commitment to grid technologies. With the awareness of the characteristics summarized in Part 2, there still are some partner grid specific issues need to be addressed before the day comes.

References

1. Foster I, Kesselman C, Tuecke S. The anatomy of the grid: Enabling scalable virtual organizations. *Int'l Journal of High Performance Computing Applications*, 2001, 15: 200-222.
2. Pawel Plaszczak and Richard Weller. *Grid computing: The savvy manager's guide*. Morgan Kaufmann, 2005.
3. Kaizar Amin, Gregor von Laszewski, Armin R. Mikler. *Grid Computing for the Masses: An Overview*. *GCC (2) 2003*: 464-473
4. L. Pouchard, L. Cinquini, B. Drach, et al., "An Ontology for Scientific Information in a Grid Environment: the Earth System Grid," *CCGrid 2003 (Symposium on Cluster Computing and the Grid)*, Tokyo, Japan, May 12-15, 2003.
5. L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, "A Community Authorization Service for Group Collaboration," in *IEEE 3rd International Workshop*
6. The GlobusAlliance Globus Toolkit. <http://www.globus.org>. 2005.

An Architecture to Support Programming Algorithm Learning by Problem Solving

Francisco Jurado, Miguel A. Redondo, and Manuel Ortega

Computer Science and Engineering Faculty
University of Castilla-La Mancha
Paseo de la Universidad, 4 – 13071 Ciudad Real – Spain
{Francisco.Jurado,Miguel.Redondo,Manuel.Ortega}@uclm.es

Abstract. Programming learning is an important subject for the students of computer science. These students must acquire knowledge and abilities which will deal with their future programming work for solving problems. In this sense, the discipline of programming constitutes a framework where Problem Based Learning (PBL) is the base used for acquiring the knowledge and abilities needed. Computer programming is a good research field where students should be assisted by an Intelligent Tutoring System (ITS) that guides them in their learning process. Furthermore, the complexity of these eLearning environments makes indispensable the necessity of the reuse and interoperability principles among eLearning tools. In this paper we will present an architectural approach that enables PBL for programming learning, merging several techniques: from Artificial Intelligence (AI) disciplines such as Bayesian Networks (BN) and Fuzzy Logic (FL); and from eLearning standards such as IMS Learning Design (IMS-LD).

1 Introduction

Programming learning implies that students of computer science should acquire and develop several abilities and aptitudes. Therefore, it is a suitable subject for the application of the Problem Based Learning (PBL) [8] and Computer Supported Collaborative Learning (CSCL) [11]. In PBL, students must solve problems presented strategically by the teacher applying the notion of *trial and error*. Solving several kinds of problems will make students acquire the abilities and aptitudes they need to solve the real problems in their future labour life.

Students of programming use computers for developing the learning activities the teacher has specified. This makes it an ideal environment for Computer Assisted Learning (CAL), where students are assisted by an Intelligent Tutoring System (ITS) that guides them in their learning process, helping them to improve and to acquire the abilities and aptitudes they should acquire and develop, leaving the slow *trial and error* process.

ITS allows adapting the learning process to each student. For this, ITS has its base on determining which the student cognitive model is, knowing which parts of the domain he/she knows. With this, ITS can determinate the next learning activity for

each specific student. ITS are usually used together with Adaptive Hypermedia Systems (AHS) for providing “intelligent” navigation through the educative material and learning activities. These systems that merge ITS and AHS are the so-called Adaptive Intelligent Educational Systems (AIES). Several examples of systems that integrate AHS and ITS for programming learning can be found in EML-ART [5], Interbook [4], KBS-Hyperbook [13] or AHA! [7]

On the other hand, the growth of Web-Based Education (WBE) environments has made groups such as IEEE LTSC, IMS Global Learning Consortium or ADL work for providing a set of standards allowing reusability and interoperability into the eLearning industry, setting the standard base where engineers and developers must work for getting eLearning systems integration.

One of those specifications is IMS Learning Design (IMS-LD) [10]. IMS-LD has the aim of centring on cognitive characteristics and in the learning process, allowing isolating the learning design process from the learning object development.

In this paper we will outline an architectural approach for developing a system that allows PBL as a learning/teaching strategy in an environment to be used for programming learning in Computer Science. This architecture merges: by one side standard eLearning specifications, allowing communication, integration and interoperability with other eLearning systems; on the other hand Artificial Intelligent techniques for enabling ITS to lead students to achieve the abilities and aptitudes they need for their future work.

The paper is structured as follows: first an overview of the architecture will be shown (section 2); next the different parts of architecture will be described in detail, that is, the student model and the support for adaptation (section 3), and the evaluation of the solution (section 4); finally some concluding remarks and future works will be commented.

2 Architecture Overview

There are some interesting proposals of eLearning architectures that point out a service-oriented view such as the one provided by Brusilovsky [2], where the need to create a distributed service-based AHS is shown, isolating the *student model server* from the *activity server* and from other *value-adding services* [2]. We propose an approach taking in mind the idea of services integration. The architecture of this proposal is shown in figure 1. The top of the figure illustrates the following models:

- *Student cognitive model*: specifying what parts of the domain the student know and in what degree. Due to the nature of this kind of model, it must allow managing uncertainty for *estimating* what knowledge the student has.
- *Instructional model*: specifying how the learning/teaching process is specified, that is, it must allow specifying which the learning design is.

These models can constitute to the main body of a generic ITS. With the *instructional model* the teaching strategy can be set. Merging it with the *student cognitive model*, a learning flow adaptation can be performed.

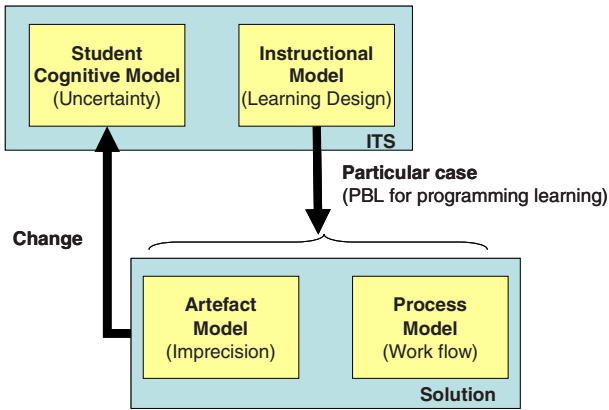


Fig. 1. Architectural approach

However, in our case of study, a concrete learning design must be implemented, that is, PBL for programming algorithm learning. Then, for supporting this, models that support managing the solution must be considered. These models are shown at the bottom of figure 1:

- *Artefact model*: it allows supporting processing and analysing artefacts developed as solution for a proposed problem. This process must provide a mechanism for managing the imprecision and vagueness in which both teacher and student specify and the solution.
- *Process model*: it allows tracking all the steps students have made until reaching the final solution, that is, the evolution of the developed artefact from the beginning till the delivered solution. In other words, the work flow followed by the student in the solution design.

The artefact and process models that analyse the solution interact with the student cognitive model for updating it, reflecting the evidences of knowledge that have given shape to the solution developed by the student. In this way, in function of the student's work, the instructional adaptation can be achieved, deciding the next learning activity to be proposed. Thus, the learning activities are shown as a consequence of how the student solves problems. In the next sections we will show each of these models in depth.

3 Student Model and Instructional Adaptation

In the previous section, we have pointed out as a requirement for the student model that it must process uncertainty. This means that the system will have evidence about the degree of knowledge of some concepts and will be able to estimate the degree of knowledge about other concepts.

An artificial intelligence technique that allows the process of uncertainty is Bayesian Networks (BN) [14]. BN is suitable in diagnostic situations, that is, it allows that given an evidence (known values for a set of variables), the subsequent

probability for the non observed variables can be calculated. This is known as evidence propagation.

Our aim is to provide an approximation that allows creating ITS considering student cognitive model and the instructional strategy needed to teach the lesson. In this sense, it is necessary to use techniques from AHS, summarized in Brusilovsky's taxonomy [3]: a) *A user model based in concepts*: this consists in a set of concepts with attributes such as the degree of knowledge; b) *Adaptive link hiding*: this means that a set of Boolean expressions can be defined in function of values from the user model, with this, the showing and hiding of a link can be evaluated; c) *Conditional inclusion of fragments*: this introduces a set of conditional blocks that allows the appearance or not of text fragments.

In instructional design, we must provide learning action sequencing. Then Brusilovsky's taxonomy will be adopted in our proposal in the following way:

- *Student cognitive model*: it models the cognitive stage for the student in every moment. This matches the user model based on the concept taken from Brusilovsky's taxonomy.
- *Instructional model*: it allows specifying the instructional strategy to be applied. This matches the adaptive link hiding and conditional inclusion fragments from Brusilovsky's taxonomy. Thus, learning activities substitute links and fragments, for example reading a text, designing quizzes, multimedia simulation, chats, etc.

We propose the use of BN for the student cognitive model. Since our aim is to develop an ITS that allows applying instructional strategies according to the subject to learn/teach, we propose the use of IMS-LD [10] for specifying the method that must be used in the teaching/learning process, that is, for specifying the instructional adaptation model.

IMS-LD represents instructional strategies using the theatre metaphor. Hence, an instructional *method* is divided in *play* elements that contain several *acts* where different *roles* bring *activities* in specific *environments* in their *role-parts*. These activities can be classified into three kinds: 1) *learning-activities* that lead the learner to get the knowledge; 2) *support-activities* that do not contribute to the learning process itself, but are needed for the success of learning activities; 3) *structure-activities* that allow structuring learning activities, support-activities or other structure-activities in sequence or selection order. All these activities can be brought into specific *environments*. These environments have *learning objects* and *services* that the different roles can use in the activities.

IMS-LD can be used for developing adaptive learning (AL) [17]. LD can be enriched with a collection of variables from the student profile. These variables allow specifying conditions to determine if a learning activity or a branch of the learning flow (a set of learning activities) is shown or hidden to a specific student. This can be done in the following way: each *method* has a section for defining *conditions* that points how it must adapt itself to specific situations through rules. An example rule can be: if the student knowledge about a concept is less than 5, then the activity A1 is hidden and the activity A2 is shown, in the opposite case, activity A1 is showed and activity A2 is hid.

In our architecture, the variables used for defining the adaptation rules in the condition section of an instructional method, are obtained from the student model

represented with BN. Moreover, in our case of study (programming learning), the evidence nodes must obtain its value from the artefact (algorithm) developed by the student. In the next section we explain how to evaluate the algorithms that the students design as a result of the learning activities and how their cognitive model is updated.

4 Evaluating the Student Algorithm

For this service, our objective is having a tool that allows comparing the artefact (algorithm) developed by the student with that specified by an expert. Therefore it is necessary to have a way for representing the *approximate ideal algorithm* that the expert (the teacher) estimates for solving a certain problem. Next, the algorithm that the student has written will be compared with that *approximate ideal representation*. This situation leads to thinking in techniques of code similarities analysis: the algorithm that the student has written is better whatever nearer to the *approximate ideal representation* for the solution of the problem.

At this point, the code similarity is introduced. This is a field of study of a branch of software engineering called Software Forensic (SF) or authorship analysis [15][9]. This discipline uses scientific methods for solving plagiarism practices and finding who makes them, trying to determine the authorship of source code analyzing the degree of similarity between two code fragments. Thus, this discipline is based on software metrics analysis of subjective elements such as writing style (indentation, noted of variables ...), programming style, etc. In [15] it is remarked that: “*Software science metrics and those based on control-flow of programs produce metric values that are clearly program-specific*”. From this, we can deduce that the similarity of metrics in different codes is related with the existence of a similar functionality in both codes. This will be the premise which leads this part of our work.

By other hand, the knowledge acquisition from the expert is a hard and complex task. The expert knows how to do her/his labour, however usually he/she is not able to transmit that knowledge with enough precision. Then our artefact model must take into account imprecision and vagueness. In the case of programming, we can consider that the expert (teacher) knows what the *ideal algorithm* that solves a specific problem is. Based on this *ideal algorithm* the teacher will make the correction of that developed by the students. How to represent that *ideal algorithm* that the teacher has in her/his mind for having a computer processed specification.

The techniques used in SF are typically statistical [15] [9]. However, fuzzy logic [18] has been used in the field of the SF [12] for the manipulation of subjective elements of styles of code writing, as well as in other analyses of software engineering metrics using fuzzy decision trees [16].

Our proposal is the one shown in figure 2. In this figure the teacher writes an implementation for the *ideal approximate algorithm* that solves a problem (at the left down of the figure). Next, several software metrics that shapes its functionality will be calculated. With this, we obtain an *instance of the ideal approximated algorithm*. After that, the fuzzy set for each metric will be established in the next way: initially, each fuzzy set will be a default trapezoidal function around the metric value from the *approximate algorithm*; the teacher can evaluate first students' algorithms indicating

whether an algorithm is not correct, little correct or correct; thus, the each fuzzy set will be adapting and deforming itself. With this, we obtain a collection of fuzzy sets that characterize the algorithm. Thus, we get a *fuzzy representation* of that *ideal approximated algorithm*, that is, we obtain an *ideal approximated algorithm fuzzy representation* that solves a concrete problem (at the top of the figure 2).

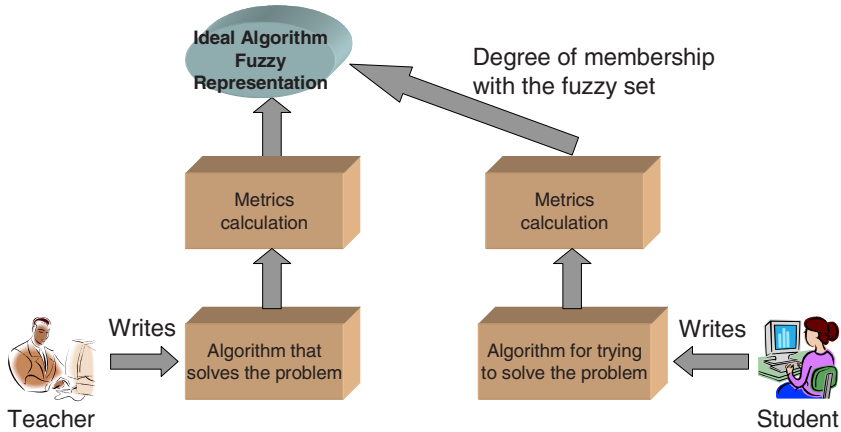


Fig. 2. Evaluating the student algorithm

Algorithms that students have written (on the right of the figure) will be correct if they are *instances* of that *ideal algorithm fuzzy representation*. Knowing the degree of membership for each software metric obtained from the algorithm wrote by students in the correspondent fuzzy set for the *ideal approximated algorithm fuzzy representation*, will give us an idea of how good is the algorithm that the students have developed.

Then, with this method, we have an artefact model that manages imprecision and vagueness; furthermore it is based on solid engineering practice (software engineering).

5 Evaluating the Student Process

Evaluating the process will give the teacher a *log* about the steps the student has made till reaching the final solution. This *log* will contain information about the changes that the student has made to the code that implements the algorithm (from a version of the algorithm to the next one). Close together with these changes, a set of software metrics and a list of errors and warnings returned by the compilation process will be added to the log.

For automatic analysis of the *log*, several methods can be used for acquiring knowledge from information, that is, automatic machine learning techniques such as data mining, fuzzy logic rules extraction, etc. In this sense, we can point the work of Barros [1], where interactions in terms of conversational structures in CSCL environments are analyzed using fuzzy logic. In this way, the system will be able to

give the teacher some *explanation* about the process followed in the activity the student has done till achieving the final solution. In this sense, we propose the use of fuzzy rules obtained with a machine learning algorithm such as the one showedn in [6], where a method for obtaining the maximal structure rules in a fuzzy logic expert system is explained.

With this, the teacher will have as feedback the algorithm developed by the student, the evaluation about this that the system has done with the method shown in section 4.1, and a process *explanation* generated by fuzzy rules. All this can be used by the teacher for *re-writing* or adapting the learning design and the *ideal approximated algorithm fuzzy representation* for improving the system.

6 Concluding Remarks and Future Works

Along this paper we have described our architectural approach in order to develop systems that enable Problem Based Learning programming learning in computer sciences. First we have begun outlining the services that the architecture must provide. Next we have introduced how each service will be implemented. In the implementation we have taken into account several eLearning standard specifications and artificial intelligent techniques. With this, we try to enable a system that leads students to achieve the abilities and aptitudes they need for their future work.

Now we are working on the evaluation of the different parts of our architecture: modelling the Bayesian Network that allows representing the student model for the programming algorithm subject; evaluating the quality of our fuzzy representation of the solution; and analysing the interaction and integration among the user model, the IMS-LD specifications and the fuzzy representation of the solution.

As future works, we want to extend our architecture for its application in Computer Supported Collaborative Learning (CSCL), taking in mind social and group models. With this we hope to provide a richer Problem Based Learning environment that allows improving constructivism for programming algorithm learning.

Acknowledgments. This research work has been partially supported by the Junta de Comunidades of Castilla-La Mancha, Spain through the “AULA_IE: Educative standards integration to the AULA e-Learning system” project (PIB-05-006), and the Ministerio de Educaci3n y Cultura, Spain through the “E-CLUB-II: Integrating collaborative and ubicuitous systems in ADAPTAPLAN” project (TIN2005-08945-C06-04).

References

1. Barros, B. and Verdejo, M.: Analysing students interaction process for improving collaboration. the DEGREE approach. In: JAIED, Vol. 11. pp. 221-241 (2000)
2. Brusilovsky, P.: KnowledgeTree: A distributed architecture for adaptive e-learning. In: WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters. , New York, NY, USA104-113 (2004)
3. Brusilovsky, P.: User Modeling and User-Adapted Interaction. In: Adaptative Hypermedia Vol. 11, nr. 1-2, pp. 87-110, Kluwer academic Publisher (2001)

4. Brusilovsky, P., Eklund, J., and Schwarz, E.: Web-Based education for all: A tool for developing adaptative courseware. In: Proceedings of Seventh International World Wide Web Conference, pp. 291-300 (1998)
5. Brusilovsky, P., Schwarz, E.W. and Weber, G.: ELM-ART: An Intelligent Tutoring System on World Wide Web. In: Intelligent Tutoring Systems, pp. 261-269 (1996)
6. Castro, J., Castro-Sanchez, J. and Zurita, J.: Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. In: Fuzzy sets and systems, Vol. 101. pp. 331-342 (1999)
7. De Bra, P., Ad Aerts, Bart Berden, Barend de Lange, Brendan Rousseau: AHA! The Adaptative Hypermedia Architecture. In Proceeding of HT'03 (2003)
8. Dewey, J.: How We Think, A Restatement of the Relation of Reflective Thinking to the Educative Process, In: D.C. Heath and company, New York (1922)
9. Gray, A., Sallis, P. and MacDonell, S.: Software Forensics: Extending Authorship Analysis Techniques to Computer Programs. In: Proceedings of the 3rd Biannual conference of the international association of forensic linguists (IAFL), International Association of Forensic Linguists (IAFL), Durham NC, USA (1997)
10. IMS-LD: IMS Learning Design. Information Model, Best Practice and Implementation Guide, XML Binding, Schemas. Version 1.0 Final Specification, Technical report, IMS Global Learning Consortium Inc, Online, <http://www.imsglobal.org/learningdesign/index.cfm> (2003)
11. Koschmann, T.: Paradigm shifts and instructional technology: an introduction. Hillsdale, NJ: Lawrence Erlbaum, pp. 1-24 (1996)
12. Kilgour, R., Gray, A., Sallis, P. & MacDonell, S.: A fuzzy logic approach to computer software source code authorsip analysis. In: Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems., Dunedin, New Zealand, pp. 865-868 (1997)
13. Nejdil, W. and Wolper, M.: KBS Hyperbook—A Data Driven Information System on the Web. In: WWW8 Conference, Toronto (1999)
14. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (1988)
15. Sallis, P.; Aakjaer, A. and MacDonell, S.: Software Forensics: old methods for a new science. In: Proceedings of the 1996 International Conference on Software Engineering: Education and Practice (SE: E&P'96), IEEE, pp. 481-485 (1996)
16. Sahraoui, H.A.; Boukadoum, M.A. & Lounis, H.: Using Fuzzy Threshold Values for Predicting Class Libraries Interface Evolution. In: Proceedings of the 4th International ECOOP workshop on Quantitative Approaches in Object-Oriented Software Engineering (2000)
17. Towel, B. & Halm, M.: Learning design: A handbook on modelling and delivering networked education and training. Springer-Verlag, chapter 12 - Designing Adaptive Learning Environments with Learning Design, pp. 215-226 (2005)
18. Zadeh, L.: Fuzzy sets. In: Information and Control, Vol. 8, pp. 338-358 (1965)

Explain a Weblog Community

Alberto Ochoa¹, Antonio Zamarró¹, SaíGonzález², Arnulfo Castro²,
and Nahitt Padilla²

¹ Maestría en Sistemas Inteligentes, Technological Institute of Leó, Méico

² Instituto de Ingeniería y Tecnología (Departamento de Ingeniería Eléctrica y Computaci3n);
UACJ

megamax8@hotmail.com

Abstract. The weblog medium while fundamentally an innovation in personal publishing has also come to engender a new form of social interaction on the web: a massively distributed but completely connected conversation covering every imaginable topic of interest. A by product of this ongoing communication is the set of hyperlinks made between weblogs in the exchange of dialog, a form of social acknowledgement on the part of authors. This paper seeks to understand the social implications of linking in the community, drawing from the hyperlink citations collected by the Blogdex project over the past three years. Social network analysis is employed to describe the resulting social structure, and two measures of authority are explored: popularity, as measured by webloggers' public affiliations and influence measured by citation of each others writing. These metrics are evaluated with respect to each other and with the authority conferred by references in the popular press.

Keywords: Web Log, Social Nets, Social Network Analysis, and Permalink.

1 Introduction

The medium of weblogging differs very little from other forms of online publishing which have constituted the web since its beginnings. During its infancy, only a handful of authors were writing dialy to websites identified as weblogs, but undoubtedly there were many thousands of others who updated their personal homepages nearly as frequently and in a similar writing style. What distinguishes weblogging from previous web media is the extent to which it is social, and one can say that the medium came into existence when the set of web journal writers recognized themselves as a community.

In the early days, there were only a handful of individuals who practiced the form, but with the addition of simple, personal publishing tools the community began an exponential growth that persists today. What was once a small family has matured into a burgeoning nation of millions including immense sub-communities around tools such as LiveJournal and DiaryLand. While some of these webloggers identify with the progenitors of the medium, others feel that their practice is distinct from that form. Regardless of affiliation, the nation of weblogging exists as such because every individual who takes part is connected to all others through the social ties of readership [8].

Every informal social system has its own order, constituted by the attribution of friendship, trust, and admiration between members. These various forms of social association give rise to higher-level organization, wherein individuals take on informal roles, such as opinion leadership, gatekeeper or maven. Within the weblog community, these positions are sought after by many authors, as they convey a sense of authority that increases readership and ties with other webloggers.

This paper is an exploration of the concept of authority as it is manifested in the community of webloggers. The Blogdex aggregator has been collecting data on the referential information contained within webloggers for the past three years, namely the hypertext links contained within webloggers' writing.

Background

Social network analysis (SNA) is a discipline of social science that seeks to explain social phenomena through a structural interpretation of human interaction both as a theory and a methodology [14]. SNA assumes a basic graph representation where individuals (actors) are characterized by nodes, and the relationships (ties) they form with each other are edges between these nodes. This graph may be undirected, assuming that all social relationships are reciprocal, or directed, where each interaction describes a one-way association between two people. The degree of any node is defined as the number of associates that node has; in the case of undirected graphs, the degree is separated into in-degree (links in) and out-degree (links out).

Social scientists have characterized power as an actor's ability to control resources and information within the network, typically by exerting some type of structural advantage over other actors. Katz and Lazarsfeld made the observation that influence is controlled by a two-step flow of communication wherein opinion trickles up to opinion leaders and then back down to the rest of the population [6]. Social network showing that innovations, rumors, and beliefs tend to move from those marginal in a network, to the central figures, and back to the rest of the population [10, 11, 12, 13].

Opinion leaders are typically observed by their centrality to a given network, or by their ability to exercise large portions of the population in question by controlling the flow of information [4,5]. Freeman has described centrality in three different measures: degree centrality, or the total number of ties an actor has, betweenness centrality, or the probability that an individual lies on a path between any two nodes in the network, and closeness centrality, the extent to which an actor is close to all other actors [3]. Since betweenness and closeness centrality require a complete description of the network and considerable computational resources for large data sets, degree centrality is typically used as simple and efficient means of calculating authority.

Network analysis is well suited for the study of weblogs as many of the social relationships between weblog authors are explicitly stated in the form of hypertext links. Webloggers have posited their own interpretation of popularity and influence based on the number of links a weblog has in various link aggregation systems. Many webloggers use the total number of links to their site to evaluate the effectiveness of their writing.

A recent debate that has raised quite a bit of attention among webloggers is related to the distribution of these links within the community, Clay Shirky wrote a piece documenting the fact that a small group of webloggers had an enormous number of links to their site while the great majority only had a few [9]. This distribution, he

claimed, followed a power law distribution, a widely observed phenomenon popularized recently in [1]. Shirky assumed this model to claim that within the weblog ecosystem, the “rich get richer”, and that the longer one has been an author; the more central they will be.

Defining Weblog Social Ties

Weblogs are a massively decentralized conversation where millions of authors write for their own audience; the conversation arises as webloggers read each other and are influenced by each others’ thoughts. A number of distinct subtypes of links have emerged within the medium, each one conveying a slightly different kind of social information:

Blogrolls

Nearly every weblog contains a list of other weblogs that the author reads regularly. Termed the blogroll. This form evolved early in the development of the medium both as a type of social acknowledgement and as a navigational tool for readers to find other authors with similar interests. In some hosted services, such as LiveJournal and Xanga, the blogroll is a core part of the interaction, allowing users to be notified when their friends make a post or even to create a group dialog represented by the sum of the group’s individual weblogs.

Permalinks

Weblogs are comprised of many individual entries, each covering a different interest or line of thinking. During the development of the first weblogging systems, it became apparent that it would be necessary to refer to specific posts instead of an entire weblog [2]; this feature allowed authors to have a sort of distributed conversation, where one post can respond to another on an entirely different weblog. These entry reference points are called permalinks and they are a core element of nearly every weblog today.

Comments

The most basic form of weblog social interaction is the comment, a reader-contributed reply to a specific post within the site. Comment systems are usually implemented as a chronologically ordered set of responses, much like web bulletin board systems.

Trackbacks

A recent feature of weblog tools is the trackback, an automatic communication that occurs when one weblog references another. If both weblogs are enabled with trackback functionality, a reference from a post on weblog A to another post on weblog B will update the post on B to contain a back-reference to the post on A. This automated referencing system gives authors and readers an awareness of who is discussing their content outside the comments on their site.

Design and Methodology

The Blogdex project was launched in 2001 as an effort to track the diffusion of information through the weblog community. The system currently tracks over 30,000 weblogs, updating its index when weblogs are changed, and keeping a record of each link made on a weblog along with the time the citation occurred. These links are aggregated into an index of the most rapidly diffusing content at any given point in time. These data are made available publicly on the project’s homepage [7].

To extract the social network from the database, we first normalize the URLs of known weblogs to deal with potential duplicates, removing any leading “www” string and any trailing file name:

`http://www.myweblog.com/index.html` → `myweblog.com`

The resulting string is termed the weblog ego as it represents a unique key to all links that come from a particular site. These strings are then queried as substrings of links in the database; when a match is found, if the normalized form of the resulting URL is the same as the weblog ego, we assume this is a blogroll link. When the URL points to content other than the front page, we presume the link is a permalink. For example, if the ego in question is “myweblog.com”, we would identify the following blogroll and permalinks as such:

`http://www.myweblog.com/` → blogroll

`http://myweblog.com/archives/001385.html` → permalink

The social network is then represented by recording the weblog the link occurred on, the weblog the link pointed to, and the type of link (blogroll or permalink). Since one weblog can link to many permalinks on another given weblog, we associate a weight score with these links, the value of which is simply the number of permalinks from one weblog to another.

Blogrolls and permalinks represent two different types of social reference. A link made on a blogroll is made explicitly as a statement of social affiliation. By placing a link to another weblog, one assumes that the author either endorses that weblog, wishes to promote it, or claims to read it on regular basis. To achieve this comparison, we will examine the top 1,000 weblogs by degree for both sets of network data, qualitatively analyzing the top weblogs for each ranking and looking quantitatively at the distribution of authority across these sites.

2 Results

The network data collected by Blogdex contains 27,976 weblogs that have at least one inbound or outbound tie. The blogroll network data consists of 116,234 ties between these weblogs while the permalink data contains 285,970 ties. The higher density of permalink ties can be attributed to the fact that these relations accrue over time, and during the process of writing a weblog, while blogroll links require an explicit effort.

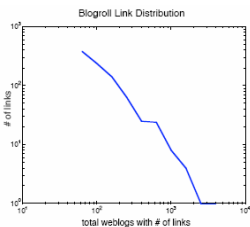


Fig. 1. Blogroll distribution

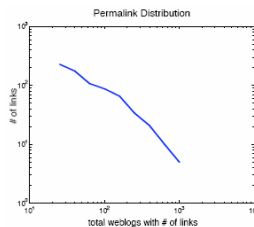


Fig. 2. Permalink distribution

In ranking these sites, the first observation is that most of the top sites are standard weblogs per se, they are weblog-like tools supported by communities of authors. Metafilter, Slashdot, Plastic, Fark and others depend upon tens of thousands of people for their content, while the rest of the list consists of sites operated by one or a small group of people. We can think of these systems as playing a role that a single human cannot, i.e. maintaining social ties with thousands of individuals. These sites play a crucial role in connecting large parts of the weblog network, resonating the important information that is diffusing through the community. Figures 1 and 2 show the distribution of rank across the weblog social network for both blogroll links and permalinks. The first observation from these plots that the slope of the blogroll distribution is slightly steeper than the permalinks, suggesting that the falloff for authority in blogrolls is quicker, leaving a bigger separation between those at the tail of the curve.

Table 1. Top authoritative sites by Blogroll and Permalink degree

Rank	Blogroll Degree Rank		Permalink Degree Rank	
1	2581	metafilter.com	1322	boineboing.net
2	2434	slashdot.org	1270	diveintomark.org
3	2145	boingboing.net	1096	metafilter.com
4	1825	kottke.org	1073	slashdot.org
5	1604	instapundit.com	982	kottke.org
6	1527	scripting.com	976	weblog.siliconvalley.com/column/dangillmor
7	1307	evhead.com	956	instapundit.com
8	1220	andrewsullivan.com	828	andrewsullivan.com
9	1062	memepool.com	827	themorningnews.org
10	1007	doc.weblogs.com	826	rathergood.com
11	977	megnut.com	819	textism.com
12	961	littlegreenfootballs.com/weblog	683	denbeste.nu
13	899	diveintomark.org	626	doc.weblogs.com
14	880	littleyellowdifferent.com	625	asmallvictory.net
15	848	textism.com	582	rightwingnews.com
16	846	rebaccablood.net	577	microcontentnews.com
17	758	plasticbag.org	568	joi.ito.com
18	737	ashes.com/anil	560	buzzmachine.com
19	719	ftrain.com	553	waxy.org
20	714	plastic.com	522	a.wholelottanothing.org

Table 1 shows the top 20 sites by degree for both the blogroll and permalink network data sets. While some sites, such as Kottke, Boingboing, Andrewsullivan and Instapundit maintain high rank in both lists, as the list continues it becomes increasingly divergent. Many of the earlier “A-List” weblogs, such as Rebecca Blood, Scripting News, Megnut and LittleYellowDifferent do not place in the top 20 for permalinks, suggesting that while their names are recognized and placed on many blogrolls, they are not writing content as widely influential as those with high permalink rank. The power law distribution observed there contains many authors whose weblogs are half as old as those at the top of the blogrolls.

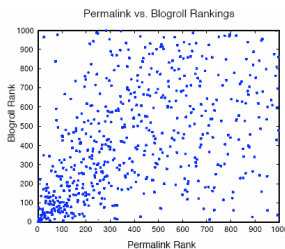


Fig. 3. Permalink and blogroll rank differential

Figure 3 demonstrates the differential between these data sets by plotting the rank in permalink versus the rank in blogroll. While the highest ranked data points tend to cluster around similar ranks, as soon as the rank passes 100 the correlation becomes much less apparent. This is further evidence that age is not the only factor in determining rank; otherwise these two data sets would be more tightly clustered around the line with slope 1. The fact that these two measures are not closely related implies that authority as measured by popularity cannot be interpreted as authority of influence.

Our queries made to Lexis Nexis returned 4,728 articles from both magazines and newspapers, of which 310 contained at least one known weblog URL. These documents yielded 545 total weblog URLs representing 212 unique sites. The twenty most cited weblogs are listed in Table 2.

Table 2. News citation rank

Rank	News citations	Blogroll Rank	Permalink Rank	Site
1	24	9	8	andrewsullivan.com
2	21	5	7	instapundit.com
3	14	6	102	scripting.com
4	12	19	39	rebeccablood.net
5	11	1	3	metafilter.com
6	11	41	144	robotwisdom.com
7	10	7	46	evhead.com
8	7	11	708	memepool.com
9	6	14	66	megnut.com
10	6	147	166	bgbg.blogspot.com
11	5	22	23	plasticbag.org
12	5	42	18	buzzmachine.com
13	5	61	54	benhammersley.com
14	5	117	617	danbricklin.com/log
15	5	184	223	links.net
16	5	962	564	voxpolitics.com
17	4	28	151	camworld.com
18	4	38	2376	obscurestore.com
19	4	72	1052	loobylu.com
20	4	86	110	ntk.net

3 Discussion

The websites collected by Blogdex were originally culled from lists of weblogs available at the time of its creation, but has since become an opt-in service for any weblogs who wish to participate. One caveat of the system is that the data set includes a selection bias based on the individuals who choose to participate. Newer weblog aggregators such as Technorati operate on an opt-out policy that creates a much more comprehensive set. Blogdex is currently transitioning towards a similar model, but the results contained within this paper are constructed from data collected under the opt-in system.

In the process of migrating the system from opt-in to opt-out, the addition of new sites to the system has been halted, resulting in a data set that is missing new weblogs over the past 5 months. However, since we are focusing only on the top 1,000 ranked weblog in each category, missing these newer sites should not drastically affect our

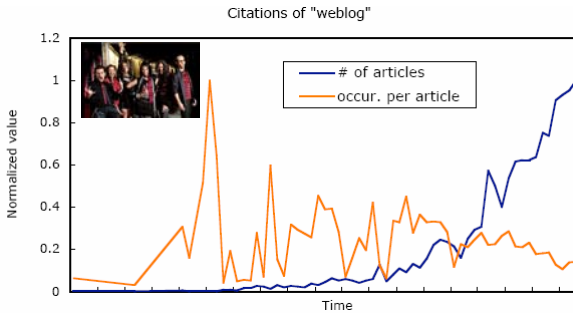


Fig. 4. Articles containing the term “weblo”, “blog”, or “web log”, and average number of usages per article related to *Timbiriche Band*

results. When a more complete data set is collected, unless we have missed entire sub-networks of authors, we expect our distributions to simply be shifted up by the factor of increase. Another probable explanation for the news citation rankings is that the role of weblogs in journalism is changing. Figure 4 shows the number of articles containing a weblog term versus the average number of times these terms are used per article. While the number of articles about weblogs shows exponential growth, the number of times the term is used per article has started to wane. This is a sign that the concept of the weblog has become part of our vernacular, and as such articles about weblogging alone are probably on the decline. More recent articles are likely to be influenced more by weblogs, and less about the medium itself.

4 Future Work

This work is by no means complete, and could be benefited from a number of future research directions. First, this work completely ignores the dynamic element of social networks; the suspicion that blogrolls reflect a “rich get richer” scenario more than permalinks could be easily validated by examining the growth of degree for network members over time [9].

Second, not all permalinks are created equal as some weblog posts receive many orders of magnitude more traffic than others. In some cases, one post can define a weblog’s permalink rank entirely, even though this attention is quickly lost. Looking at the distribution of permalink citations for each individual weblog may allow us to renormalize the data to avoid this phenomenon. Finally, the data collected by Blogdex, while useful for the task of measuring authority, is not complete. Moving into an opt-out system and crawling weblogs at large will present a much more accurate picture of authority and influence.

5 Conclusions

The initial excitement over the weblog power law made many webloggers uncomfortable. How can a person get excited about a medium where attention is garnered by the number of weeks one has participated? Looking only at popularity by

blogroll rank, it does appear that the “rich get richer”, but another assessment of authority, permalinks, might be an equally good proxy to authority and a better measure of influence. Barabási has noted that the growth of scale free networks is not only determined by the age of nodes, but also by the *node strength*, an undefined property related to a node’s ability to acquire links. Permalink rank might be an accurate way of measuring node strength, and a better proxy to authority and influence at a given point in time.

References

1. Barabási, Albert-László(2002) “Linked: The new science of networks” Cambridge, MA: Perseus Publishing.
2. Dash, Anil. (2004) Interview with Paul Bausch 2003 [cited March 24, 2004]. Available from http://www.sixapart.com/log/2003/09/interview_with_.shtml.
3. Freeman, Linton C. (1978). “Centrality in social networks conceptual clarification”. *Social Networks* (3): 215-239.
4. Granovetter, Mark. (1973). “The Strength of Weak Ties”. *The American Journal of Sociology* 78 (6): 1360-1380.
5. Granovetter, Mark (1983). “The Strength of Weak Ties: A Network Theory Revisited”. *Sociological Theory* 1:201-233.
6. Katz, Elihu and Paul F. Lazarsfeld (1955). *Personal Influence*. Glencoe, IL: Free Press.
7. Marlow, Cameron (2004). *Blogdex 2001* [cited May 2004]. Available from <http://blogdex.net/>.
8. Marlow, Cameron (2002). “Getting the Scoop: Social Networks for News Dissemination”. Paper read at Sunbelt Social Network Conference XXII, at New Orleans, LA.
9. Ochoa A. et al. (2007) “Discover behavior of Turkish people in Orkut” 17th International Conference on Electronics, Communications and Computers; Puebla, México.
10. Rogers, Everett M. (2003). “Diffusion of innovations”. 5th ed. New York: Free Press.
11. Shirky, Clay. (2004) “Power laws, weblogs and inequality 2003” [cited May 15, 2004]. Available from http://www.shirky.com/writings/powerlaw_weblog.html.
12. Valente, Thomas W. (1995). “Network models of the diffusion of innovations, Quantitative methods in communication”. Cresskill, N.J.: Hampton Press.
13. Weimann, Gabriel. (1982). “On the importance of marginality: One more step in the two-step flow of communication”. *American Sociological Review* 47 (6): 764-773.
14. Wellman, Barry. (1997) “Structural analysis: From method and metaphor to theory and substance”. In *Social structures: A network approach*, edited by B. Wellman and S.D. Berkowitz. Greenwich, CT: JAI Press.

Implementing Data Mining to Improve a Game Board Based on Cultural Algorithms

Alberto Ochoa¹, Saúl González², Arnulfo Castro², Nahitt Padilla²,
and Rosario Baltazar¹

¹ Maestría en Sistemas Inteligentes, Technological Institute of León, México

² Instituto de Ingeniería y Tecnología (Departamento de Ingeniería Eléctrica y Computación); UACJ
megamax8@hotmail.com

Abstract. Evolutionary computation is a generic term used to make reference to the solution of computational problems planned and implemented based on models of an evolutionary process. Most of the evolutionary algorithms propose biological paradigms, and the concepts of natural selection, mutation and reproduction. However, other paradigms that can be adopted in the creation of evolutionary algorithms exist. Many problems involve not structured environments that can be considered from the perspective of cultural paradigms; the cultural paradigms offer a wide range of categorized models that ignore the possible solutions to the problem, -a common situation in the real life-. The purpose of the present work is to apply the computational properties of the cultural technology; on this case, to corroborate them by means of data mining to propose the solution to a specific problem. The above mentioned, carrying out an adaptation from the perspective of the societies modeling. An environment to carry out tests of this type was developed to allow the learning on the not very conventional characteristics of a cultural technology. This environment is called Baharastar.

Keywords: Cultural Algorithms, Data Mining, Modeling of Societies.

1 Introduction

The most of computational problems found at real World, do not have a definitive (final) solution [1]. Cultural Algorithms uses the culture like a vehicle to store accessible relevant information to the population's members during many generations, were developed to model the evolution of the cultural component over the time and to demonstrate how this learns and acquires knowledge [10]. A cultural algorithm can be described with the next pseudo code.

Initially, a population of individuals that represents the solution space, which is represented as a group of solutions inside the search space, is randomly generated to create the first generation. In our example, the solution space contains a list of the attributes that can be used in the classification procedure. The space of beliefs is empty. For each generation, the Cultural Algorithm will be able to involve a population of individuals using the “frame” Vote-Inherit-Promote (VIP). During the

```

Begin
  t=0;
  Initialize POP(t); /* Initialization of population */
  Initialize BLF(t); /* Initialization of believing space */
  Repeat
    Evaluate POP(t);
    Vote (BLF (t), Accept (POP(t)));
    Adjust (BLF (t));
    Evolve(POP(t), Influence(BLF(t)));
    t = t +1;
    Select POP(t) from POP(t-1);
  Until (Term condition is reached)
End
    
```

Fig. 1. Pseudo code of Cultural Algorithms

phase of Vote of this process, the population's members are evaluated to identify their contribution to the space of beliefs using the acceptance function. These beliefs allow contributing in most of the solution of the problem and they are selected or put on voting to contribute to the current space of beliefs. The beliefs space is modified when the inherited beliefs are combined with the beliefs that have been added by the current generation, this is carried out using a reasoning process that allows updating the space of beliefs.

Next, the updated space of beliefs is used to influence in the population's evolution. The belief space is used to influence the path on which the population of variable combinations is modified. During the last phase a new population is reproduced using a basic set of evolutionary operators. This new population could be evaluated and the cycle continues successively. The VIP Cycle finishes when a condition of termination is introduced. This condition is usually reached when only a small or none change is detected in the population through several generations or when certain knowledge has emerged in the space of beliefs, as can be appreciated in the Figure 2.

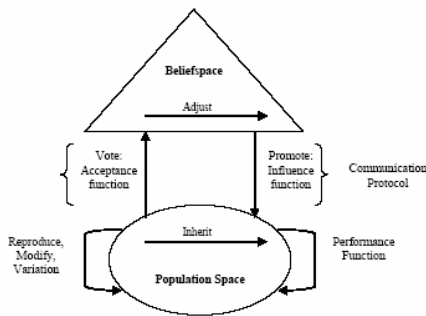


Fig. 2. Conceptual Diagram of Cultural Algorithms

According to [9], the human brain could be an instrument of cognitive metaphors – a metaphoric mind that could “to surmise” the obtained knowledge in previous form from previous experiences and to complete the missing knowledge to solve real world

problems. The investigation on the topic has worked in gradual different form to the models offered by the biology [6]. The Cultural Algorithms, for example, are based on the supposition of obtaining the best learning ranges, as it happens in an evolutionary algorithm (like in the genetic algorithms) [5] adding to this an element more of the evolutionary pressure, called “space of beliefs”, - a mechanism of cultural pressure. It has frequently suggested, that the cultural evolution allows to the societies to develop or to adapt them to their environments in ranges that exceed to the biological evolution only based on genetic inheritance [10].

Another research topic called Artificial Societies, according to [3,4], consist on the simulation of theories or social models expressed in the form of computer programs.

The main contribution of this paper, refers to the fact that the culture has its own cultural properties. In second place, we believe that some computational problems could benefit from a cultural method of resolution, and this can lead to the automation of processes by means of the use of intelligent agents [5].

Several authors have written about the confrontation between the individuals and the society, and how this last influences their behavior (changes of the society over the time) [5]. Gessler, makes an approach to the Artificial Societies regarding cultural aspects, in which proposes some key concepts in relation to the artificial cultures. These are: **Time** –this allows supporting the researcher to simulate the behavior of the society toward the future, the past, or in diverse scales according to the programmer's interest. **Space** – this can represent bi-dimensionally for a matrix. **Agents** – actors that interact to each other, that will have aspects of the behavior (as they interact in the world) and cognitive aspects (how they think). Some of these actions will be external; others will be internal in a cognitive sense. **Devices** – by mean of these elements, the agents can interact and exchange information.

Taking into consideration these concepts, we are able to elaborate a representation of what could be described as an artificial culture of agents [2].

2 Artificial Culture and Its Protocol

In this article, we focus our attention on a practical problem adapted from the related literature within the Societies Modeling, “the negotiation toward a common well-being” for a set of societies: to find a safe place (a place where attacks don't exist) in an unknown place, inside a hostile environment with unknown dimensions and populated by attackers in unknown locations.

This type of optimization problem can be represented by a two-dimensional matrix, called “dimension”, like is shown in the Figure 3, where A represents the group of societies, M and B the Goal and the attackers (both unknown for the societies) respectively, and the numbers in the dimension represent the experimentation cost for each space. The objective of the Cultural Algorithm is to find the goal in the minimum number of steps while the spaces are sorted where “attacks” can exist, characterized by penalties of anxiety.

The solution to this problem will be given by a sequence of agents' generations, denoted as “community.” The agents can only know the adjacent spaces to them, like in the colonies carried out by a society that only knows finite distances. The group of spaces around the agent is denominated “quadrant.” From the agent's point of view, this optimization problem is absolutely complex, because we don't know the location

of the goal – or if some exists – and it cannot see the world beyond its quadrant. Besides doesn't have any previous heuristic to try to improve the optimization. For better understanding of the selected cultural algorithm used to solve the optimization problem, now we introduce some basic concepts and representations of the artificial culture related to this problem. These representations are abstraction levels located between (the unknown part of the agent), the domain problem (dimension) and the agents. The union of these abstraction levels constitutes the artificial culture protocol – the model that links the agents' dynamic, the domain problem and its cultural representation.

In the algorithm of cultural change, the spaces of the space of beliefs (beliefspace) by means of the best paradigm (BestParadigm) are set to zero, representing the fact that the culture increases the quantity of pleasure associated with such spaces, giving an incentive to the behavior associated with the best paradigm (BestParadigm).

Agents

The agents are the actors those that will be able to experience each space in the dimension to what Freud refers as the “principle of satisfaction”, according to this, the agent will be able to select the spaces with the lower experimentation cost.

Paradigm

The paradigm is the agents' personal representation for the space of beliefs (beliefspace) or its personal interpretation of the cultural references. According to Gessler, this is the agent's cognition and its private vision of the cultural interpretation of the World. The paradigm could represent the best solution for the problem denoted as the best paradigm (BestParadigm).

Space of beliefs (Beliefspace)

The space of beliefs is the collective representation of the real World. In other words, this is the world as it is interpreted by one culture of the community, where the agents find the way to interact and moral values.

Dimension

The dimension is the real world, which never can be entirely known by the agent. This contains the experimentation cost and on which the agents are able to live when the optimization is improved.

Exploration

The agents belonging to one community search inside the dimension for the most appropriated place to be developed (goal). The obtained solution for the agents whom find the goal in the lesser number of steps could be considered as the community “model”, or the best paradigm (BestParadigm). According Geertz, this model or ideology is a “diagram of the psychological and social processes”. The culture could then try to lead the behavior of the new generations of agents by means of this best solution. The best solution for the optimization problem will be given by the agent's sequence of movements that find the totality of the optimum number of steps. Each agent in the community is leaded by one function that allows it to select the spaces with the lower quantity of anxiety. In case of more than one space in the quadrant posses a minimum identical anxiety, the agent will select one in a free way, in our

case, is represented by one random selection. It can be observed that the satisfaction principle does not affect the strategy for the global resolution of the problem at collective level (culture). To the contrary, this links the agent with an autonomous entity. The culture controls the behavior to be adopted as model, creating a strategy of global action –an ideology- regarding the given problem domain.

The agent selects the cell with the minimum anxiety, as the indicated for the space of beliefs (Beliefspace) adding to this the cultural value, as:

$$\text{beliefspace}(x) = \text{beliefspace}(x) + \text{dimension}(x)$$

Where x is a set of spaces in the dimension

In this research the functions represent the agent-culture interaction and are selected according the adopted problem.

Therefore, we cannot try to establish a mathematical model of how the cultural process occurs in the real world. Adopting a random function as was shown previously to explain how we insert, into the process, a system of multiple interactions between the agent and the culture. We try to analyze other mathematical representations in our future work.

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	B	3	1	1	1	1
1	B	B	B	1	1	1	1
1	B	1	1	1	1	1	1
1	1	1	2	1	M	1	1
1	1	A	1	1	1	1	1

Fig. 3. Representation of the Dimension

3 Cultural Algorithms Simulator

To prove and validate the theoretical concepts previously presented, we develop a Cultural Algorithms Simulator (Baharastar). Initially our intention was only to create an environment able to carry out analysis and experiments. When genetic algorithms are used it becomes more difficult of understanding the peculiarities for each solution. Each time that the system has a precise answer, the obtained solution can hardly being duplicated exactly. This property of the evolutionary algorithms in general and of the cultural algorithms in particular, has been little explored or discussed in the literature. The creation of systems with an individuality or “soul”, are our contribution in the area. For such purpose, we select 27 societies described in Memory Alpha [7] and we characterize their behavior using seven base attributes (agility, ability to fight, intelligence, forces, resistance, speed and emotional control), those which allowed describe as well to the society as to the individual.

The development of Baharastar is based on our desire of sharing an intuitive understanding about the treatment for a new class of systems, individuals able to possess unexpected creativity, typical characteristic of living entities. Baharastar is shown in the figure 6, the user has the possibility to choose the starting point and the goal to reach, joined to the places where one can receive an attack by part of the enemy, and the quantity of anxiety associated to each space of the dimension where the societies reside in (agents' communities). Our prototype was developed using JBuilder X platform.

4 Complementary Methodology

Data mining is the search of global patterns and the existent relationships among the data of immense databases, but that are hidden in them inside the vast quantity of information [13]. These relationships represent knowledge of value about the objects that are in the database. This information is not necessarily a faithful copy of the information stored in the databases. Rather, is the information that one can deduce from the database. One of the main problems in data mining is that the number of possible extracted relationships is exponential [12]. Therefore, there are a great variety of machine’s learning heuristics that have been proposed for the knowledge discovery in databases [11].

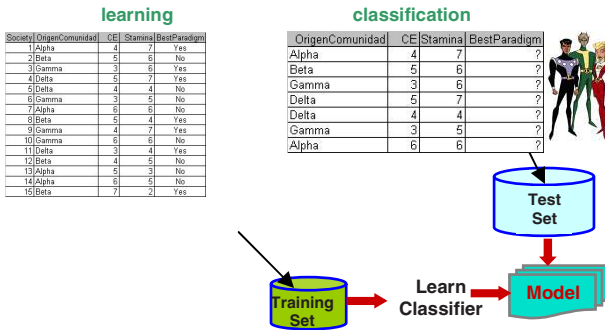


Fig. 4. Proposed decision tree, used to characterize the negotiation level inside each community

One of the most popular approaches to represent the results of data mining is to use decision trees [8]. A decision tree provides a procedure to recognize a given case for a concept. It is a “divide and conquer” strategy for the acquisition of the concept (instance). The decision trees have been useful in a great variety of practical cases in science and engineering, in our case we use data mining to characterize the individuals of each society (agents’ community) and to understand how they obtain the best paradigm, like it is shown in figure 4.

5 Experiments

In this section, we describe the developed experiments using Baharastar. We hope to contribute in the sense of making evident the importance of the creation of a new methodology to prove and to analyze the obtained results. This was not a trivial task, considering the diversity of behaviors of the provided solutions by Baharastar because it resembles more than a descriptive anthropology than a simple software test.

In the first experiment, we compared the performance of 27 communities of 50 agents, and on the other hand 27 communities of 500 agents each one. The associated points to the beginning and goal are shown in the figure 5. The optimal number of steps from the beginning to the goal is 12.

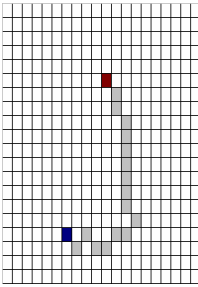


Fig. 5. Evaluation of a optimization problem using Baharastar

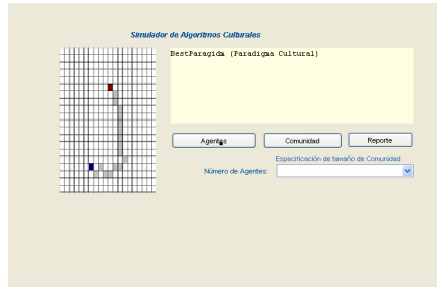


Fig. 6. Developed tool called Baharastar

One of the most interesting characteristics observed in this experiment is the diversity of cultural patterns established for each community. For the solutions with the same number of steps the provided result for the “beliefspace” is entirely different. The structured scenarios associated to the agents cannot be reproduced in general due they belong to a given instant in the time and space. They represent a unique, precise and innovative form of adaptive behavior which solves a computational problem followed by a complex change of relationships. The generated configurations can be metaphorically related to the knowledge of the community behavior regarding to an optimization problem (to make alliances, to defend from a possible invasion), or a tradition with which to emerge from the experience and with which to begin a dynamics of the process. Comparing the 50 agents of the first community regarding the 500 agents community, this last obtained a better performance in terms of the average number of steps from the beginning to the goal (13.05 versus 14.30), as well as a smaller standard deviation (1.96 versus 2.64). They also had a greater average number of changes in the paradigm (5.85 versus 4.25), which indicates that even the “less negotiating” generations, that explored less interesting parts of the dimension, could optimize their search to achieve better results. In the second experiment, we consider the same scenario for the experiment one, except that after having obtained a solution from a community of 50 agents, we place five near spaces to the goal and we begin with a new community of 500 agents. The new community was informed of the previous cultural configurations but should take into account the new scenario. The comparison among both solutions is not immediate, from the point of view that try to solve different problems.

In this experiment, it was surprising to see initially how the community of 500 agents uses the solution offered by the 50 agents, whenever these solutions were close the optimal grade, instead of finding entirely complete new solutions. These results make evident the conservation of a global action strategy which regulates the agents. This can be compared metaphorically with the concept of culture mentioned in the introduction.

6 Conclusions

Using Data Mining we improve the understanding of change for the best paradigm substantially, because we classify the communities of agents appropriately based on their related attributes approach, this allows us to understand that the “negotiation” concept exists with base on the determination of acceptance function by part of the remaining communities to a particular proposed solution by a community.

Cultural Algorithms offers a powerful alternative for optimization problems. For that reason is that provides a comprehensible panoramic of the cultural phenomenon.

This technology lead us about the possibility of the experimental knowledge generation, created by the community of agents for a given application domain. How much the degree of this knowledge is cognitive for the community of agents is a topic for the future work. The answer can be similar to the involved in the hard work of communication between two different cultures. The specification of each artificial culture could be a contradiction to the digital systems, characterized by an extensive reproduction, and lacking the concepts of the original and copy.

Otherwise to the traditional digital systems, characterized by an extensive reproduction, and annulled from the concepts associated to original or copy, as well as the implicit technological culture for the singular individuality of each system, that is totally different from the today’s industrial standards. A new Artificial Intelligence that can be in charge of these systems, continues being distant into the horizon, in the same way that we still lack of methods to understand the original and peculiar things of each society.

References

1. Desmond, A. & Moore J. (1995). Darwin – la vida de un evolucionista atormentado. Generación Editorial, São Paulo, Brazil.
2. Geertz, C. (1989). La interpretación de las culturas. Editorial Guanabara, RJ, Brazil.
3. Gessler, N. (1999). Artificial Culture – Experiments in Synthetic Anthropology. <http://www.soc.surrey.ac.uk/JASSS/1/1/4.html>
4. Memory Alpha., Proyecto de Wikipedia relacionado con Star Trek. <http://www.memory-alpha.org/en/wiki/Main>.
5. Ochoa A.et al. (2006) “Italianità: Discovering a Pygmalion effect on Italian Communities Using Data Mining”. In Proceedings of CORE’2006.
6. Pinker, S. (1999). ¿Cómo funciona nuestra mente?. Compañ de Letras, São Paulo, Brazil.
7. Reynolds, R.G. (1998). “An Introduction to Cultural Algorithms”. Cultural Algorithms Repository, <http://www.cs.wayne.edu/~jcc/car.html>.
8. Suaremi T. (2003) “Understand social groups” NDAM’2003, Reykiavik; Iceland.
9. Tang Huét et al. (2006) “Towards Robust Indexing for Ranked Queries”, VLDB’06, Seoul, Korea.
10. Zevzevič I. (2005) “Similar cultural relationships in Montenegro” JASSS’2005, England.

Author Index

- Abraham, Ajith 215, 255, 320
Abreu, Raquel 280
Adserias, José Francisco 401
Andrade, Carlos A. Camacho 239
Armengol, E. 112
- Bajo, Javier 296, 377
Baltazar, Rosario 486
Barber, Fernando 17
Baruque, Bruno 151
Biswas, Arijit 255
Blanco, Ángela 393
Botelho, Sílvia S.C. 425
Botía, Juan A. 9
Botti, Vicente 304
Burduk, Robert 48
BurgosArtizzu, Xavier-P. 72
Bursa, Miroslav 191
- Caballero-Gil, P. 183
Cantón, Francisco R. 353, 361
Carbonero, M. 88
Carreón, Carlos A. Hernández 247
Castro, Arnulfo 478, 486
Catalá, Alejandro 337
Chaudhry, Qasim 80
Chaves, Antonio Augusto 136
Chen, Zonghai 159
Chira, Camelia 33
Claros, M. Gonzalo 353, 361
Colares, Rafael G. 425
Conde, Eustorgio Meza 454
Corchado, Emilio 151, 320
Corchado, Juan M. 25, 377
- Corral, G. 112
Craciun, Marian 80
Crowther, R.A. 345
Cruz-Reyes, Laura 223
- Das, Swagatam 255
Dasgupta, Sambarta 255
de Antonio, Angélica 1
de Assis Correa, Francisco 136
De Las Rivas, Javier 393
de Lope, Javier 104, 167, 175
De Los Angeles Hernandez, Ma. 64
de Luis, Ana 25
de Miguel, P. 199
de Paz, Juan F. 25, 377
de Paz, Yanira 296, 377
del Olmo, Esther 401
del Rosario Suárez, María 56
Delgado, O. 183
Díaz, David 401
Díaz-Moreno, Sara 361
Domínguez, Enrique 128
Dorronsor, José R. 120
Dumitrescu, D. 33, 207
- Falcón, Antonio 369
Falgueras, Juan 353
Fan, Hao 462
Fernández, J.C. 88
Fernández, J.J. 345, 385
Fernandez, Karla Espriella 247
Flores, José A. Martínez 239
Fornells, A. 112, 312
Fúster-Sabater, A. 183

- Gaglio, Salvatore 329
 Galilea, Emiliano Hernández 418
 García, Andrés 104
 Garmendia, Luis 40
 Gatani, Luca 329
 Golobardes, E. 112, 312
 Gómez, Jorge J. 9
 González, Saúl 478, 486
 Grande, Javier 56
 Grando, Neusa 425
 Grimaldo, Francisco 17
 Gui, Hao 462
 Guijarro, María 280
 Guo, Li 433
 Gutierrez, German 231
 Gutiérrez, P.A. 88

 Hassan, Samer 40
 He, Qing 264
 Heras, Stella 304
 Hernández, Mario 369
 Herrera, David Vicente 288
 Herrero, Álvaro 320
 Hervás, C. 88
 Huacuja, Héctor J. Fraire 239, 247

 Izaguirre, Rogelio Ortega 454

 Jaen, Javier 337
 Jiménez, Álvaro Barbero 120
 Julián, Vicente 304
 Jurado, Francisco 470

 Lara, Antonio J. 353, 361
 LaTorre, A. 199
 Lázaro, Jorge López 120
 Lhotska, Lenka 191
 Li, S. 345, 385
 Li, Xiaoxia 433
 Liang, Chunyang 433
 López, José Luis 411
 López, Tania Turrubiates 454
 López-Pérez, José Luis 401
 Lo Re, Giuseppe 329
 Lorena, Luiz Antonio N. 136
 Lorenzo, Javier 369
 Lozano, Miguel 17
 Lu, Zhaogan 272
 Lucic, V. 385
 Lung, Rodica Ioana 207

 Maldonado, José Orlando 288
 Martín, Quintín 296
 Martín H., José Antonio 104, 175
 Martín-Merino, Manuel 393
 Medarde, Manuel 411
 Mendez, Gerardo M. 64
 Méndez, Juan 369
 Mocholí, Jose A. 337
 Moreno, María N. 446
 Muelas, S. 199
 Muñoz, José 128
 Muñoz, Lluís A. Belanche 96

 Neagu, Daniel 80
 Nieto-Yáñez, Diana Maritza 223

 Ochoa, Alberto 478, 486
 Ortega, Manuel 470
 Ortolani, Marco 329

 Padilla, Nahitt 478, 486
 Pajares, Gonzalo 72, 280
 Pant, Millie 215
 Pavón, Juan 9, 40
 Peláez, Rafael 411
 Pellicer, María A. 320
 Peña, J.M. 199
 Peralta, Juan 231
 Pereda, Juan C. 167
 Pérez-Trabado, Guillermo 353, 361
 Pinteá, Camelia-M. 33

 Redondo, Miguel A. 470
 Reyes, Laura Cruz 454
 Ribeiro, Angela 72
 Robles, V. 199
 Rodellar, Maria Victoria 167
 Rodríguez, Sara 25, 377
 Ramón Villar, José 56
 Romay, Manuel Graña 288
 Romero, Oscar J. 1

 Sánchez, Juan M. 25
 Sanchis, Araceli 231
 Santillán, Claudia Guadalupe Gómez
 454
 Santos, Matilde 104, 175
 Santos-García, Gustavo 418
 Segrera, Saddys 446
 Shen, Junyi 272

- Shi, Zhiping 264
Shi, Zhongzhi 264
Simas, Gisele M. 425
Suárez-Bárcena, Inés Franco 418
- Tapia, Dante I. 296
Tellaeché, Alberto 72
Thangaraj, Radha 215
Theron, Roberto 401
Tolama, Juana E. Mancilla 247
Tomás-Solis, Pedro 223
Trundle, Paul 80
- Valdez, Georgina Castillo 239
Valdez, Guadalupe Castilla 223, 239,
247
- Vaquero, Miguel 401
Vargas, David Romero 239
Vigueras, Guillermo 9
Villalobos, David P. 361
- Wang, Haibo 159
Wei, Shuzhi 439
Wozniak, Michal 144
- Xia, Zhaojie 433
- Yang, Genke 439
Yang, Zhangyuan 433
Ye, Zhongxing 439
- Zamarrón, Antonio 478
Zhang, Mingxin 272

