LNCS 5517

Joan Cabestany
Francisco Sandoval
Alberto Prieto
Juan M. Corchado (Eds.)

Bio-Inspired Systems: Computational and Ambient Intelligence

10th International Work-Conference on Artificial Neural Networks, IWANN 2009
Salamanca, Spain, June 2009, Proceedings, Part I

1
Part I

Springer

Lecture Notes in Computer Science 5517

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK
Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA
Josef Kittler
University of Surrey, Guildford, UK
Jon M. Kleinberg
Cornell University, Ithaca, NY, USA
Alfred Kobsa
University of California, Irvine, CA, USA
Friedemann Mattern
ETH Zurich, Switzerland
John C. Mitchell
Stanford University, CA, USA
Moni Naor
Weizmann Institute of Science, Rehovot, Israel
Oscar Nierstrasz
University of Bern, Switzerland
C. Pandu Rangan
Indian Institute of Technology, Madras, India
Bernhard Steffen
University of Dortmund, Germany
Madhu Sudan
Massachusetts Institute of Technology, MA, USA
Demetri Terzopoulos
University of California, Los Angeles, CA, USA
Doug Tygar
University of California, Berkeley, CA, USA
Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Joan Cabestany Francisco Sandoval
Alberto Prieto Juan M. Corchado (Eds.)

Bio-Inspired Systems: Computational and Ambient Intelligence

10th International Work-Conference
on Artificial Neural Networks, IWANN 2009
Salamanca, Spain, June 10-12, 2009
Proceedings, Part I

Springer

Volume Editors

Joan Cabestany
Universitat Politècnica de Catalunya - UPC
E.T.S.E. Telecomunicació, Barcelona, Spain
E-mail: cabestan@eel.upc.es

Francisco Sandoval
Universidad de Málaga
E.T.S.I. Telecomunicación, Málaga, Spain
E-mail: fsandoval@uma.es

Alberto Prieto
Universidad de Granada
E.T.S.I. Informática y Telecomunicación, Granada, Spain
E-mail: aprieto@ugr.es

Juan M. Corchado
Universidad de Salamanca
Departamento de Informática, Salamanca, Spain
E-mail: corchado@usal.es

Library of Congress Control Number: Applied for

CR Subject Classification (1998): J.3, I.2, I.5, C.2.4, H.3.4, D.1, D.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-02477-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-02477-1 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, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2009
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12695607 06/3180 5 4 3 2 1 0

In memoriam of Prof. Dr. José Mira,
advocate of scientific research on the interplay
between natural and artificial computation,
Co-chair of IWANN from 1991 to 2005,
and friend.

Preface

This volume presents the set of final accepted papers for the tenth edition of the IWANN conference "International Work-Conference on Artificial neural Networks" held in Salamanca (Spain) during June 10–12, 2009.

IWANN is a biennial conference focusing on the foundations, theory, models and applications of systems inspired by nature (mainly, neural networks, evolutionary and soft-computing systems). Since the first edition in Granada (LNCS 540, 1991), the conference has evolved and matured. The list of topics in the successive Call for Papers has also evolved, resulting in the following list for the present edition:

1. **Mathematical and theoretical methods in computational intelligence.** Complex and social systems. Evolutionary and genetic algorithms. Fuzzy logic. Mathematics for neural networks. RBF structures. Self-organizing networks and methods. Support vector machines.
2. **Neurocomputational formulations.** Single-neuron modelling. Perceptual modelling. System-level neural modelling. Spiking neurons. Models of biological learning.
3. **Learning and adaptation.** Adaptive systems. Imitation learning. Reconfigurable systems. Supervised, non-supervised, reinforcement and statistical algorithms.
4. **Emulation of cognitive functions.** Decision making. Multi-agent systems. Sensor mesh. Natural language. Pattern recognition. Perceptual and motor functions (visual, auditory, tactile, virtual reality, etc.). Robotics. Planning motor control.
5. **Bio-inspired systems and neuro-engineering.** Embedded intelligent systems. Evolvable computing. Evolving hardware. Microelectronics for neural, fuzzy and bio-inspired systems. Neural prostheses. Retinomorphic systems. Brain-computer interfaces (BCI). Nanosystems. Nanocognitive systems.
6. **Ambient intelligence**. Unobtrusive hardware. Seamless mobile/fixed communication and computing infrastructure. Dynamic and massively distributed device networks. Human-centric computer interfaces. Dependable and secure systems and devices. Ambient-assisted living (AAL).
7. **Applications.** Adaptive interfaces. Ambient intelligence. Biomimetic applications. Data analysis and pre-processing. Data mining. Economy and financial engineering. Fuzzy systems for control. Internet. Neural networks for control. Power systems. Signal processing. Telecommunication applications. Time series and prediction.

At the end of the scheduled period, we got more than 230 submitted papers under the above topics. A careful peer review process was organized, and, as its main result, we are able to offer you the following collection of 167 papers, including 3 invited conferences and a reduced number of tutorial contributions by some of the special sessions organizers, on the hot topic of their respective session.

It must be pointed out that, for the sake of consistency and readability of the book, the presented papers are not organized as they were presented in the IWANN sessions, but classified under 15 chapters with the listed specific topics:

1. Theoretical Foundations and Models
2. Learning and Adaptation
3. Self-Organizing Networks, Methods and Applications
4. Fuzzy Systems.
5. Evolutionary Computation and Genetic Algorithms
6. Pattern Recognition
7. Formal Languages in Linguistics
8. Agents and Multi-agents on Intelligent Systems
9. Brain-Computer Interfaces (BCI)
10. Multiobjetive Optimization
11. Robotics
12. Bioinformatics
13. Biomedical Applications
14. Ambient-Assisted Living (AAL) and Ambient Intelligence (AI)
15. Other Applications

IWANN 2009 was organized as the main conference, together with four complimentary events (accepted papers have been included in the LNCS 5518 volume):

- DCAI 2009 (International Symposium on Distributed Computing and Artificial Intelligence), covering artificial intelligence and its applications in distributed environments, such as the Internet, electronic commerce, mobile communications, wireless devices, distributed computing, and so on.
- IWAAL 2009 (International Workshop of Ambient-Assisted Living), covering solutions aimed at increasing the quality of life, safety and health problems of elderly and disabled people by means of technology.
- IWPACBB 2009 (Third International Workshop on Practical Applications of Computational Biology and Bioinformatics), covering computational biology and bioinformatics as a possibility for knowledge discovery, modelling and optimization tasks, aiming at the development of computational models so that the response of biological complex systems to any perturbation can be predicted.
- SOCO 2009 (4th International Workshop on Soft Computing Models in Industrial Applications), covering the implementation of soft computing in industrial applications.

The organizers decided to share the site and some of the plenary conferences and social events, trying to facilitate the attendance of delegates to all the conferences and workshops.

The tenth edition of IWANN was organized by the Universitat Politècnica de Catalunya, the Universidad de Málaga and the Universidad de Granada, together with the Universidad de Salamanca as the local manager. We wish to thank to the Spanish Ministerio de Ciencia e Innovación, the Universidad de Salamanca, the City Council of Salamanca and the Castilla-León Government for their support and grants.

We would also like to express our gratitude to the members of the different committees for their support, collaboration and good work. Finally, we want to thank Springer, and especially Alfred Hoffman and Anna Kramer for their continuous support and cooperation.

June 2009

Joan Cabestany
Francisco Sandoval
Alberto Prieto
Juan M. Corchado

Organization

IWANN 2009 Organizing Committee

Joan Cabestany (Chair)	Technical University of Catalonia (Spain)
Alberto Prieto (Chair)	University of Granada (Spain)
Francisco Sandoval (Chair)	University of Malaga (Spain)
Gonzalo Joya	University of Malaga (Spain)
Francisco García Lagos	University of Malaga (Spain)
Miguel Atencia	University of Malaga (Spain)
Pedro Castillo	University of Granada(Spain)
Alberto Guillén	University of Jaen (Spain)
Beatriz Prieto	University of Granada (Spain)
Juan M. Corchado	University of Salamanca (Spain)
Sara Rodríguez	University of Salamanca (Spain)
Juan F. De Paz	University of Salamanca (Spain)
Javier Bajo	Pontifical University of Salamanca (Spain)
Emilio S. Corchado	University of Burgos (Spain)

IWANN 2009 Program Committee

Igor Aleksander	Imperial College (UK)
Andreas Andreu	Johns Hopkins University (USA)
Plamen Angelov	University of Lancaster (UK)
Cecilio Angulo	Polytechnic University of Catalonia (Spain)
Antonio Artés	Carlos III University of Madrid (Spain)
Antonio Bahamonde	University of Oviedo (Spain)
Sergi Bermejo	Polytechnic University of Catalonia (Spain)
Piero Bonissone	GE CRD Information Technology Laboratory (USA)
Andreu Catalá	Polytechnic University of Catalonia (Spain)
Pert Cauwenberghs	The Johns Hopkins University (USA)
Jesus Cid-Sueiro	Carlos III University of Madrid (Spain)
Rafael Corchuelo	University of Seville (Spain)
Carlos Cotta	University of Malaga (Spain)
Marie Cottrell	University of Paris (France)
Alicia d'Anjou	University of País Vasco (EHU) (Spain)
Javier de Lope	Polytechnic University of Madrid (Spain)
Luiza de Macedo	University of Rio de Janeiro (Brazil)
Dante del Corso	Polytechnic of Turin (Italy)
Angel P. del Pobil	University of Jaume I (Spain)
Suash Deb	C.V. Raman College of Engineering (India)

Richard Duro	University of La Coruña (Spain)
Reinhard Eckhorn	Philipps University (Germany)
Marcos Faundez-Zanuy	Polytechnic University of Catalonia (Spain)
J. Manuel Fernández	Polytechnic University of Cartagena (Spain)
Ramon Ferrer Cancho	University of Rome (Italy)
Heinrich Flar	Technical University of Berlin (Germany)
Dario Floreano	Swiss NSF, EPFL (Switzerland)
Jean-Claude Fort	University of Paul Sabatier (France)
Kunihiko Fukushima	Osaka University (Japan)
Chistian Gamrat	French Atomic Energy Commission (France)
Patrik Garda	University of Paris Sud (France)
Karl Goser	University of Dortmund (Germany)
Manuel Graña	University of País Vasco (Spain)
Anne Guérin-Dugué	Signal and Image Laboratory (France)
Alister Hamilton	University of Edinburgh (UK)
Barbara Hammer	University of Osnabrück (Germany)
Martin Hasler	Federal Polytechnic School of Lausanne (Switzerland)
Jeanny Hérault	Grenoble Institute of Technology (France)
Francisco Herrera	University of Granada (Spain)
Cesar Hervás	University of Cordoba (Spain)
Tom Heskes	University of Nijmegen (The Netherlands)
Giacomo Indiveri	Institute of Neuroinformatics (Switzerland)
Pedro Isasi	Carlos III University of Madrid (Spain)
Simon Jones	Loughborough University (UK)
Christian Jutten	Grenoble Institute of Technology (France)
Tin Kam Ho	Bell Labs (USA)
Kathryn Klemic	University of Yale (USA)
Amaury Lendasse	Helsinki University of Technology (Finland)
Kurosh Madani	University of Paris XII (France)
Jordi Madrenas	Polytechnic University of Catalonia (Spain)
Luis Magdalena	Polytechnic University of Madrid (Spain)
Dario Maravall	Polytechnic University of Madrid (Spain)
Bonifacio Martín del Brio	University of Zaragoza (Spain)
Wolfgang Maass	Graz University of Technology (Austria)
Francesco Masulli	Sapienza University of Rome (Italy)
José M. Molina	Carlos III University of Madrid (Spain)
Augusto Montisci	University of Cagliari (Italy)
Claudio Moraga	Dortmund University (Germany)
Juan M. Moreno	Polytechnic University of Catalonia (Spain)
Klaus-Robert Müller	Fraunhofer Institute for Computer Architecture and Software Technology FIRST (Germany)
José Muñoz	University of Malaga (Spain)
Alan F. Murray	University of Edinburgh (UK)
Jean-Pierre Nadal	Normal Superior School (France)
Nadia Nedjah	State University of Rio de Janeiro (Brazil)
Erkki Oja	Helsinki University of Technology (Finland)

Juan Pavón	Complutense University of Madrid (Spain)
Julio Ortega	University of Granada (Spain)
Kevin M. Passino	The Ohio State University (USA)
Witold Pedrycz	University of Alberta (Canada)
Francisco José Pelayo	University of Granada (Spain)
Andrés Perez-Uribe	University of Applied Sciences of Western Switzerland
Vicenzo Piuiri	University of Milan (Italy)
Carlos G. Puntonet	University of Granada (Spain)
Leonardo Reyneri	Polytechnic of Turin (Italy)
Ignacio Rojas	University of Granada (Spain)
Eduardo Ros	University of Granada (Spain)
Ulrich Rückert	University of Paderborn (Germany)
Javier Ruiz-del-Solar	University of Chile (Chile)
Wei-Chiang Samuelson Hong	Oriental Institute of Technology (Taiwan)
Eduardo Sanchez	Federal Polytechnic School of Lausanne (Switzerland)
Juan V. Sanchez-Andrés	University of La Laguna (Spain)
Juan A. Sigüenza	Autonomous University of Madrid (Spain)
Jordi Solé-Casals	University of Vic (Spain)
Peter Szolgay	Hungarian Academy of Sciences (Hungary)
John Taylor	King's College London (UK)
Fabian Theis	University of Regensburg (Germany)
Carme Torras	Polytechnic University of Catalonia (Spain)
Joaquín J. Torres	University of Granada (Spain)
Mark Van Rossum	University of Edinburgh (UK)
Marley Vellasco	Pontifical Catholic University of Rio de Janeiro (Brazil)
Alfredo Vellido	Polytechnic University of Catalonia (Spain)
Michel Verleysen	Catholic University of de Louvain-la-Neuve (Belgium)
Thomas Villmann	University of Leipzig (Germany)
Changjiu Zhou	Singapore Polytechnic (Singapore)
Ahmed Zobaa	University of Cairo (Egypt)
Pedro Zufiria	Polytechnic University of Madrid (Spain)

Other Reviewers

Vanesa Aguiar
Arnufo Alanis
Amparo Alonso Betanzos
Gabriela Andrejkova
Davide Anguita
Luis Antunes
Miguel Atencia
Ismail Babaoglu
Davide Bacciu
Mª Encarnación Beato
Lluis Belanche
Roberto Berjón
José Luís Bernier
Guillermo Botella
Juan A. Botia
Charles Bouveyron
Joan Cabestany
Inma P Cabrera
Tomasa Calvo
Mario Cámara
Francesco Camastra
Carlos Carrascosa
Pedro Castillo

Henning Christiansen
Valentina Colla
Emilio Corchado
Juan Manuel Corchado
Pablo Cordero
Ulises Cortés
Raúl Cruz
Erzsébet Csuhaj-Varjú
Jürgen Dassow
Gregorio Díaz
Julián Dorado
Gerard Dreyfus
Csaba Egyhazy
Patrik Eklund
Alberto Fernández
Enrique Fernández
Aníbal R. Figueiras-Vidal
Leonardo Franco
Ana Freire
Mª Ángeles Galán
Juan Miguel García
María Isabel García Arenas
Esther García-Garaluz
J M García-Gómez
Francisco García-Lagos
Maite García-Sebastián
Petia Georgieva
Marcos Gestal
Peter Glösekötter
Vanessa Gómez
Jesús González
Juan-Manuel Gorriz
Janis Grundspenkis
Alberto Guillén
Rob Hierons
Christian Igel
Juan-Luis Jiménez
María-Dolores Jiménez
Gonzalo Joya
Vicente Julián
Johan Karlsson
Constantine Kotropoulos
Stanislav Krajci
Manfred Kudlek
Jorma Laaksonen
Alberto Labarga
Kui Lin
Paulo J.G. Lisboa
Miguel Ángel López-Gordo
Luis Martínez
Montserrat Mateos
Jesús Medina
Juan-Julián Merelo
Gustavo Meschino
Mónica Miguelez
Antonio Mora
Gin Moreno
Ramón Moreno Jiménez
Cristian R. Munteanu
Manuel Ojeda-Aciego
Iván Olier
Madalina Olteanu
Mohamed Oubbati
Rubem Pereira
Juan Luis Pérez
Hugo Picado
Héctor Pomares
Alberto Prieto
Beatriz Prieto
Juan R. Rabuñal
Eickhoff Ralf
Ander Ramos
Reinhard Rapp
Daniel Rivero
Sara Rodríguez
Samuel Romero
Ricardo Ron
Fabrice Rossi
Joseph Rynkiewicz
Miguel-Ángel Sánchez
Francisco Sandoval
José Antonio Seoane
Eduardo Serrano
Pilar Sobrevilla
Claude Touzet
Oswaldo Trelles
Ignacio Turias
Elif Derya Ubeyli
Pablo Varona
Eleni Vasilaki
Gyürgy Vaszil
Francisco Velasco-Álvarez
Carmen Vidaurre
Peter Vojtas

IWANN 2009 Invited Speakers

- Klaus-Robert Müller
  Technical University of Berlin, Germany
- Estebán Pérez Castrejón
  Telefónica I+D, Madrid, Spain
- Dario Floreano
  Laboratory of Intelligent Systems, EPFL Lausanne, Switzerland

IWANN 2009 Special Sessions Organizers

J. Bajo, J.M. Corchado, University of Salamanca
P. J.G. Lisboa, A. Vellido; Technical University of Catalonia - UPC
F. Pelayo, R. Ron, M.A. Lopez; University of Granada
M.A. Sanchez, M.E. Beato; University of Salamanca
J. Dorado, J.R. Rabuñal; University of Coruña
G. Bel-Enguix, M.D. Jimenez; University of Rovira i Virgili
H. Pomares, I. Rojas; University of Granada
M. Ojeda, P. Cordero, Inma P. Cabrera; University of Malaga
R. Marfil, F. Escolano; University of Malaga
I. Rodriguez; Complutense University of Madrid
M. Olteanu (SAMOS- University of Paris 1);
E. García-Garalúz (ISIS- University of Málaga)
J.M. Molina, A. Berlanga; University of Carlos III (Madrid)
C. Urdiales; University of Malaga

Table of Contents – Part I

1. Theoretical Foundations and Models

Lower Bounds for Approximation of Some Classes of Lebesgue Measurable Functions by Sigmoidal Neural Networks 1
José L. Montaña and Cruz E. Borges

A Wavelet Based Method for Detecting Multiple Encoding Rhythms in Neural Networks 9
Carlos Aguirre and Pedro Pascual

Switching Dynamics of Neural Systems in the Presence of Multiplicative Colored Noise 17
Jorge F. Mejias, Joaquin J. Torres, Samuel Johnson, and Hilbert J. Kappen

Gradient Like Behavior and High Gain Design of KWTA Neural Networks 24
Daniela Danciu and Vladimir Răsvan

Fast Evaluation of Connectionist Language Models 33
F. Zamora-Martínez, M.J. Castro-Bleda, and S. España-Boquera

Improving the Consistency of AHP Matrices Using a Multi-layer Perceptron-Based Model 41
Jose Antonio Gomez-Ruiz, Marcelo Karanik, and José Ignacio Peláez

Global and Local Modelling in Radial Basis Functions Networks 49
L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, G. Rubio, and J. Urquiza

A Preliminar Analysis of CO^{2}RBFN in Imbalanced Problems 57
M.D. Pérez-Godoy, A.J. Rivera, A. Fernández, M.J. del Jesus, and F. Herrera

Feature Selection in Survival Least Squares Support Vector Machines with Maximal Variation Constraints 65
V. Van Belle, K. Pelckmans, J.A.K. Suykens, and S. Van Huffel

A Simple Maximum Gain Algorithm for Support Vector Regression 73
Álvaro Barbero and José R. Dorronsoro

Domains of Competence of Artificial Neural Networks Using Measures of Separability of Classes 81
Julián Luengo and Francisco Herrera

Self-estimation of Data and Approximation Reliability through Neural Networks ... 89
Leonardo M. Reyneri, Valentina Colla, Mirko Sgarbi, and Marco Vannucci

FPGA Implementations Comparison of Neuro-cortical Inspired Convolution Processors for Spiking Systems ... 97
A. Linares-Barranco, R. Paz, F. Gómez-Rodríguez, A. Jiménez, M. Rivas, G. Jiménez, and A. Civit

2. Learning and Adaptation

Nonparametric Location Estimation for Probability Density Function Learning ... 106
Ezequiel López-Rubio, Juan Miguel Ortiz-de-Lazcano-Lobato, and María Carmen Vargas-González

An Awareness-Based Artificial Neural Network for Cooperative Distributed Environments ... 114
Mauricio Paletta and Pilar Herrero

Improving Classification under Changes in Class and Within-Class Distributions ... 122
Rocío Alaiz-Rodríguez, Alicia Guerrero-Curieses, and Jesús Cid-Sueiro

Improving Training in the Vicinity of Temporary Minima ... 131
Ido Roth and Michael Margaliot

Convergence in an Adaptive Neural Network: The Influence of Noise Inputs Correlation ... 140
Adel Daouzli, Sylvain Saïghi, Michelle Rudolph, Alain Destexhe, and Sylvie Renaud

Adaptative Resonance Theory Fuzzy Networks Parallel Computation Using CUDA ... 149
M. Martínez-Zarzuela, F.J. Díaz Pernas, A. Tejero de Pablos, M. Antón Rodríguez, J.F. Díez Higuera, D. Boto Giralda, and D. González Ortega

A Supervised Learning Method for Neural Networks Based on Sensitivity Analysis with Automatic Regularization ... 157
Beatriz Pérez-Sánchez, Oscar Fontenla-Romero, and Bertha Guijarro-Berdiñas

Ensemble Methods for Boosting Visualization Models ... 165
Bruno Baruque, Emilio Corchado, Aitor Mata, and Juan M. Corchado

New Artificial Metaplasticity MLP Results on Standard Data Base 174
Alexis Marcano-Cedeño, Aleksandar Jevtić, Antonio Álvarez-Vellisco, and Diego Andina

3. Self-organizing Networks, Methods and Applications

Probabilistic Self-Organizing Graphs 180
Ezequiel López-Rubio, Juan Miguel Ortiz-de-Lazcano-Lobato, and María Carmen Vargas-González

Spicules for Unsupervised Learning 188
J.A. Gómez-Ruiz, J. Muñoz-Perez, and M.A. García-Bernal

Topology Preserving Visualization Methods for Growing Self-Organizing Maps 196
Soledad Delgado, Consuelo Gonzalo, Estibaliz Martinez, and Agueda Arquero

Making Standard SOM Invariant to the Initial Conditions 204
Soukeina Ben Chikha and Kirmene Marzouki

The Complexity of the Batch Neural Gas Extended to Local PCA 212
Iván Machón-González, Hilario López-García, and José Luís Calvo-Rolle

Self Organized Dynamic Tree Neural Network 220
Juan F. De Paz, Sara Rodríguez, Javier Bajo, Juan M. Corchado, and Vivian López

Development of Neural Network Structure with Biological Mechanisms 228
Samuel Johnson, Joaquín Marro, Jorge F. Mejias, and Joaquín J. Torres

4. Fuzzy Systems

Fuzzy Logic, Soft Computing, and Applications 236
Inma P. Cabrera, Pablo Cordero, and Manuel Ojeda-Aciego

A Similarity-Based WAM for Bousi∼Prolog 245
Pascual Julián-Iranzo and Clemente Rubio-Manzano

On the Declarative Semantics of Multi-Adjoint Logic Programs 253
P. Julián, G. Moreno, and J. Penabad

A Complete Logic for Fuzzy Functional Dependencies over Domains with Similarity Relations 261
P. Cordero, M. Enciso, A. Mora, and I.P. de Guzmán

RFuzzy: An Expressive Simple Fuzzy Compiler 270
Susana Munoz-Hernandez, Victor Pablos Ceruelo, and Hannes Strass

Overcoming Non-commutativity in Multi-adjoint Concept Lattices 278
Jesús Medina

Evolutionary Fuzzy Scheduler for Grid Computing 286
R.P. Prado, S. García Galán, A.J. Yuste, J.E. Muñoz Expósito, A.J. Sánchez Santiago, and S. Bruque

Improving the Performance of Fuzzy Rule Based Classification Systems for Highly Imbalanced Data-Sets Using an Evolutionary Adaptive Inference System 294
Alberto Fernández, María José del Jesus, and Francisco Herrera

A t-Norm Based Approach to Edge Detection 302
C. Lopez-Molina, H. Bustince, J. Fernández, E. Barrenechea, P. Couto, and B. De Baets

5. Evolutionary Computation and Genetic Algoritms

Applying Evolutionary Computation Methods to Formal Testing and Model Checking 310
Pablo Rabanal, Ismael Rodríguez, and Fernando Rubio

Applying Evolutionary Techniques to Debug Functional Programs...... 318
Alberto de la Encina, Mercedes Hidalgo-Herrero, Pablo Rabanal, and Fernando Rubio

Aiding Test Case Generation in Temporally Constrained State Based Systems Using Genetic Algorithms 327
Karnig Derderian, Mercedes G. Merayo, Robert M. Hierons, and Manuel Núñez

Creation of Specific-to-Problem Kernel Functions for Function Approximation 335
Ginés Rubio, Héctor Pomares, Ignacio Rojas, and Alberto Guillén

Combining Genetic Algorithms and Mutation Testing to Generate Test Sequences 343
Carlos Molinero, Manuel Núñez, and César Andrés

Testing Restorable Systems by Using RFD 351
Pablo Rabanal and Ismael Rodríguez

RCGA-S/RCGA-SP Methods to Minimize the Delta Test for Regression Tasks.. 359
Fernando Mateo, Dušan Sovilj, Rafael Gadea, and Amaury Lendasse

An Evolutionary Hierarchical Clustering Method with a Visual Validation Tool 367
José A. Castellanos-Garzón, Carlos Armando García, and Luis A. Miguel-Quintales

An Adaptive Parameter Control for the Differential Evolution Algorithm 375
Gilberto Reynoso-Meza, Javier Sanchis, and Xavier Blasco

Parallelizing the Design of Radial Basis Function Neural Networks by Means of Evolutionary Meta-algorithms 383
M.G. Arenas, E. Parras-Gutiérrez, V.M. Rivas, P.A. Castillo, M.J. Del Jesus, and J.J. Merelo

A Genetic Algorithm for ANN Design, Training and Simplification 391
Daniel Rivero, Julian Dorado, Enrique Fernández-Blanco, and Alejandro Pazos

6. Pattern Recognition

Graph-Based Representations in Pattern Recognition and Computational Intelligence 399
R. Marfil, F. Escolano, and A. Bandera

Kernelization of Softassign and Motzkin-Strauss Algorithms 407
M.A. Lozano and F. Escolano

Connectivity Forests for Homological Analysis of Digital Volumes 415
Pedro Real

Energy-Based Perceptual Segmentation Using an Irregular Pyramid 424
R. Marfil and F. Sandoval

Hierarchical Graphs for Data Clustering 432
E.J. Palomo, J.M. Ortiz-de-Lazcano-Lobato, Domingo López-Rodríguez, and R.M. Luque

Real Adaboost Ensembles with Emphasized Subsampling 440
Sergio Muñoz-Romero, Jerónimo Arenas-García, and Vanessa Gómez-Verdejo

Using the Negentropy Increment to Determine the Number of Clusters 448
Luis F. Lago-Fernández and Fernando Corbacho

A Wrapper Method for Feature Selection in Multiple Classes Datasets 456
Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, and Rosa M. Calvo-Estévez

7. Formal Languages in Linguistics

New Challenges in the Application of Non-classical Formal Languages to Linguistics 464
Gemma Bel-Enguix and M. Dolores Jiménez-López

PNEPs, NEPs for Context Free Parsing: Application to Natural Language Processing 472
Alfonso Ortega, Emilio del Rosal, Diana Pérez, Robert Mercaş, Alexander Perekrestenko, and Manuel Alfonseca

A Hyprolog Parsing Methodology for Property Grammars 480
Veronica Dahl, Baohua Gu, and Erez Maharshak

Adaptable Grammars for Non-Context-Free Languages 488
Henning Christiansen

β-Reduction and Antecedent–Anaphora Relations in the Language of Acyclic Recursion 496
Roussanka Loukanova

Permutation Languages in Formal Linguistics 504
Benedek Nagy

8. Agents and Multi-agent on Intelligent Systems

Thomas: Practical Applications of Agents and Multiagent Systems 512
Javier Bajo and Juan M. Corchado

INGENIAS Development Process Assisted with Chains of Transformations 514
Iván García-Magariño, Rubén Fuentes-Fernández, and Jorge J. Gómez-Sanz

A Secure Group-Oriented Model for Multiagent Systems 522
Jose M. Such, Juan M. Alberola, Antonio Barella, Agustin Espinosa, and Ana Garcia-Fornes

Interactive Animation of Agent Formation Based on Hopfield Neural Networks 530
Rafael Kelly and Carmen Monroy

The INGENIAS Development Kit: A Practical Application for Crisis-Management 537
Iván García-Magariño, Celia Gutiérrez, and Rubén Fuentes-Fernández

The Delphi Process Applied to African Traditional Medicine 545
Ghislain Atemezing, Iván García-Magariño, and Juan Pavón

Composing and Ensuring Time-Bounded Agent Services 553
Martí Navarro, Elena del Val, Miguel Rebollo, and Vicente Julián

An Organisation-Based Multiagent System for Medical Emergency Assistance 561
Roberto Centeno, Moser Fagundes, Holger Billhardt, Sascha Ossowski, Juan Manuel Corchado, Vicente Julian, and Alberto Fernandez

TEMMAS: The Electricity Market Multi-agent Simulator 569
Paulo Trigo, Paulo Marques, and Helder Coelho

Two Steps Reinforcement Learning in Continuous Reinforcement Learning Tasks 577
Iván López-Bueno, Javier García, and Fernando Fernández

Multi-Agent System Theory for Modelling a Home Automation System 585
G. Morganti, A.M. Perdon, G. Conte, and D. Scaradozzi

THOMAS-MALL: A Multiagent System for Shopping and Guidance in Malls 594
S. Rodríguez, A. Fernández, V. Julián, J.M. Corchado, S. Ossowski, and V. Botti

Multiagent-Based Educational Environment for Dependents 602
Antonia Macarro, Alberto Pedrero, and Juan A. Fraile

Social and Cognitive System for Learning Negotiation Strategies with Incomplete Information 610
Amine Chohra, Arash Bahrammirzaee, and Kurosh Madani

Evaluation of Multi-Agent System Communication in INGENIAS 619
Celia Gutiérez, Iván García-Magariño, and Jorge J. Gómez-Sanz

Agents Jumping in the Air: Dream or Reality?...................... 627
Oscar Urra, Sergio Ilarri, and Eduardo Mena

Using Scenarios to Draft the Support of Intelligent Tools for Frail Elders in the *SHARE-it* Approach 635
R. Annicchiarico, F. Campana, A. Federici C. Barrué, U. Cortés, A. Villar, and C. Caltagirone

On the Road to an Abstract Architecture for Open Virtual Organizations 642
M. Rebollo, A. Giret, E. Argente, C. Carrascosa, J.M. Corchado, A. Fernandez, and V. Julian

9. Brain-Computer Interface (BCI)

Using Rest Class and Control Paradigms for Brain Computer Interfacing ... 651
Siamac Fazli, Márton Danóczy, Florin Popescu, Benjamin Blankertz, and Klaus-Robert Müller

The Training Issue in Brain-Computer Interface: A Multi-disciplinary Field ... 666
Ricardo Ron-Angevin, Miguel Angel Lopez, and Francisco Pelayo

A Maxmin Approach to Optimize Spatial Filters for EEG Single-Trial Classification ... 674
Motoaki Kawanabe, Carmen Vidaurre, Benjamin Blankertz, and Klaus-Robert Müller

Multiple AM Modulated Visual Stimuli in Brain-Computer Interface ... 683
M.-A. Lopez, H. Pomares, A. Prieto, and F. Pelayo

A Brain-Computer Interface Based on Steady State Visual Evoked Potentials for Controlling a Robot ... 690
Robert Prueckl and Christoph Guger

Asynchronous Brain-Computer Interface to Navigate in Virtual Environments Using One Motor Imagery ... 698
Francisco Velasco-Álvarez and Ricardo Ron-Angevin

Impact of Frequency Selection on LCD Screens for SSVEP Based Brain-Computer Interfaces ... 706
Ivan Volosyak, Hubert Cecotti, and Axel Gräser

10. Multiobjetive Optimization

Multiobjective Evolutionary Algorithms: Applications in Real Problems ... 714
Antonio Berlanga, Jesús García Herrero, and José Manuel Molina

Evolutionary Genetic Algorithms in a Constraint Satisfaction Problem: Puzzle Eternity II ... 720
Jorge Muñoz, German Gutierrez, and Araceli Sanchis

Multiobjective Algorithms Hybridization to Optimize Broadcasting Parameters in Mobile Ad-Hoc Networks ... 728
Sandra García, Cristóbal Luque, Alejandro Cervantes, and Inés M. Galván

Application Synthesis for MPSoCs Implementation Using Multiobjective Optimization 736
Marcus Vinícius Carvalho da Silva, Nadia Nedjah, and Luiza de Macedo Mourelle

Multi Objective Optimization Algorithm Based on Neural Networks Inversion 744
Sara Carcangiu, Alessandra Fanni, and Augusto Montisci

EMORBFN: An Evolutionary Multiobjetive Optimization Algorithm for RBFN Design 752
Pedro L. López, Antonio J. Rivera, M. Dolores Pérez-Godoy, María J. del Jesus, and Cristóbal Carmona

Performance Measures for Dynamic Multi-Objective Optimization 760
Mario Cámara, Julio Ortega, and Francisco de Toro

11. Robotics

Methods for Artificial Evolution of Truly Cooperative Robots 768
Dario Floreano and Laurent Keller

Social Robot Paradigms: An Overview 773
Sergi del Moral, Diego Pardo, and Cecilio Angulo

A Dual Graph Pyramid Approach to Grid-Based and Topological Maps Integration for Mobile Robotics 781
J.M. Pérez-Lorenzo, R. Vázquez-Martín, E. Antúnez, and A. Bandera

Integrating Graph-Based Vision Perception to Spoken Conversation in Human-Robot Interaction 789
Wendy Aguilar and Luis A. Pineda

From Vision Sensor to Actuators, Spike Based Robot Control through Address-Event-Representation 797
A. Jimenez-Fernandez, C. Lujan-Martinez, R. Paz-Vicente, A. Linares-Barranco, G. Jimenez, and A. Civit

Automatic Generation of Biped Walk Behavior Using Genetic Algorithms 805
Hugo Picado, Marcos Gestal, Nuno Lau, Luis P. Reis, and Ana M. Tomé

Motion Planning of a Non-holonomic Vehicle in a Real Environment by Reinforcement Learning 813
M. Gómez, L. Gayarre, T. Martínez-Marín, S. Sánchez, and D. Meziat

12. Bioinformatics

Applications in Bio-informatics and Biomedical Engineering 820
I. Rojas, H. Pomares, O. Valenzuela, and J.L. Bernier

A Large-Scale Genomics Studies Conducted with Batch-Learning SOM Utilizing High-Performance Supercomputers 829
Takashi Abe, Yuta Hamano, Shigehiko Kanaya, Kennosuke Wada, and Toshimichi Ikemura

Clustering Method to Identify Gene Sets with Similar Expression Profiles in Adjacent Chromosomal Regions 837
Min A. Jhun and Taesung Park

On Selecting the Best Pre-processing Method for Affymetrix Genechips 845
J.P. Florido, H. Pomares, I. Rojas, J.C. Calvo, J.M. Urquiza, and M. Gonzalo Claros

Method for Prediction of Protein-Protein Interactions in Yeast Using Genomics/Proteomics Information and Feature Selection 853
J.M. Urquiza, I. Rojas, H. Pomares, J.P. Florido, G. Rubio, L.J. Herrera, J.C. Calvo, and J. Ortega

Protein Structure Prediction by Evolutionary Multi-objective Optimization: Search Space Reduction by Using Rotamers 861
J.C. Calvo, J. Ortega, M. Anguita, J.M. Urquiza, and J.P. Florido

Using Efficient RBF Networks to Classify Transport Proteins Based on PSSM Profiles and Biochemical Properties 869
Yu-Yen Ou and Shu-An Chen

Artificial Neural Network Based Algorithm for Biomolecular Interactions Modeling 877
Christophe Lemetre, Lee J. Lancashire, Robert C. Rees, and Graham R. Ball

13. Biomedical Applications

Modelling Dengue Epidemics with Autoregressive Switching Markov Models (AR-HMM) 886
Madalina Olteanu, Esther García-Garaluz, Miguel Atencia, and Gonzalo Joya

A Theoretical Model for the Dengue Epidemic Using Delayed Differential Equations: Numerical Approaches 893
Andrés Sánchez Pérez, Héctor de Arazoza Rodríguez, Teresita Noriega Sánchez, Jorge Barrios, and Aymee Marrero Severo

System Identification of Dengue Fever Epidemics in Cuba 901
Esther García-Garaluz, Miguel Atencia, Francisco García-Lagos, Gonzalo Joya, and Francisco Sandoval

HIV Model Described by Differential Inclusions 909
Jorge Barrios, Alain Piétrus, Aymée Marrero, and Héctor de Arazoza

Data Mining in Complex Diseases Using Evolutionary Computation 917
Vanessa Aguiar, Jose A. Seoane, Ana Freire, and Cristian R. Munteanu

Using UN/CEFACT'S Modelling Methodology (UMM) in e-Health Projects 925
P. García-Sánchez, J. González, P.A. Castillo, and A. Prieto

Matrix Metric Adaptation for Improved Linear Discriminant Analysis of Biomedical Data 933
M. Strickert, J. Keilwagen, F.-M. Schleif, T. Villmann, and M. Biehl

SPECT Image Classification Techniques for Computer Aided Diagnosis of the Alzheimer Disease 941
J. Ramírez, R. Chaves, J.M. Górriz, M. López, D. Salas-Gonzalez, I. Álvarez, and F. Segovia

Automatic System for Alzheimer's Disease Diagnosis Using Eigenbrains and Bayesian Classification Rules 949
M. López, J. Ramírez, J.M. Górriz, I. Álvarez, D. Salas-Gonzalez, F. Segovia, and C.G. Puntonet

On the Use of Morphometry Based Features for Alzheimer's Disease Detection on MRI 957
Maite García-Sebastián, Alexandre Savio, Manuel Graña, and Jorge Villanúa

Selecting Regions of Interest for the Diagnosis of Alzheimer's Disease in Brain SPECT Images Using Welch's t-Test 965
D. Salas-Gonzalez, J.M. Górriz, J. Ramírez, M. López, I. Álvarez, F. Segovia, and C.G. Puntonet

Alzheimer's Diagnosis Using Eigenbrains and Support Vector Machines 973
I. Álvarez, J.M. Górriz, J. Ramírez, D. Salas-Gonzalez, M. López, F. Segovia, C.G. Puntonet, and B. Prieto

Artificial Intelligent Systems Based on Supervised HUMANN for Differential Diagnosis of Cognitive Impairment: Towards a 4P-HCDS ... 981
Patricio García Báez, Miguel Angel Pérez del Pino, Carlos Fernández Viadero, and Carmen Paz Suárez Araujo

Stratification Methodologies for Neural Networks Models of Survival 989
Ana S. Fernandes, Ian H. Jarman, Terence A. Etchells, José M. Fonseca, Elia Biganzoli, Chris Bajdik, and Paulo J.G. Lisboa

Model Comparison for the Detection of EEG Arousals in Sleep Apnea Patients 997
D. Álvarez-Estévez and V. Moret-Bonillo

Ranking of Brain Tumour Classifiers Using a Bayesian Approach 1005
Javier Vicente, Juan Miguel García-Gómez, Salvador Tortajada, Alfredo T. Navarro, Franklyn A. Howe, Andrew C. Peet, Margarida Julià-Sapé, Bernardo Celda, Pieter Wesseling, Magí Lluch-Ariet, and Montserrat Robles

Feature Selection with Single-Layer Perceptrons for a Multicentre ^{1}H-MRS Brain Tumour Database 1013
Enrique Romero, Alfredo Vellido, and Josep María Sopena

Weakly-Supervised Classification with Mixture Models for Cervical Cancer Detection 1021
Charles Bouveyron

Edges Detection of Clusters of Microcalcifications with SOM and Coordinate Logic Filters 1029
J. Quintanilla-Domínguez, B. Ojeda-Magaña, J. Seijas, A. Vega-Corona, and D. Andina

A New Methodology for Feature Selection Based on Machine Learning Methods Applied to Glaucoma 1037
Diego García-Morate, Arancha Simón-Hurtado, Carlos Vivaracho-Pascual, and Alfonso Antón-López

Tissue Recognition Approach to Pressure Ulcer Area Estimation with Neural Networks 1045
Francisco J. Veredas, Héctor Mesa, and Laura Morente

Classification of Schistosomiasis Prevalence Using Fuzzy Case-Based Reasoning 1053
Flávia T. Martins-Bedé, Lluís Godo, Sandra Sandri, Luciano V. Dutra, Corina C. Freitas, Omar S. Carvalho, Ricardo J.P.S. Guimarães, and Ronaldo S. Amaral

BAC Overlap Identification Based on Bit-Vectors 1061
Jens-Uwe Krause and Jürgen Kleffe

14. Ambient Assisted Living (AAL) and Ambient Intelligence (AI)

AAL and the Mainstream of Digital Home 1070
Esteban Pérez-Castrejón and Juan J. Andrés-Gutiérrez

Legal Concerns Regarding AmI Assisted Living in the Elderly, Worldwide and in Romania 1083
Luiza Spiru, Lucian Stefan, Ileana Turcu, Camelia Ghita, Ioana Ioancio, Costin Nuta, Mona Blaciotti, Mariana Martin, Ulises Cortes, and Roberta Annicchiarico

Construction and Debugging of a Multi-Agent Based Simulation to Study Ambient Intelligence Applications 1090
Emilio Serrano, Juan A. Botia, and Jose M. Cadenas

Easing the Smart Home: Translating Human Hierarchies to Intelligent Environments 1098
Manuel García–Herranz, Pablo A. Haya, and Xavier Alamán

Wireless Sensor Networks in Home Care 1106
Dante I. Tapia, Juan A. Fraile, Sara Rodríguez, Juan F. de Paz, and Javier Bajo

Indoor Localization Based on Neural Networks for Non-Dedicated ZigBee Networks in AAL 1113
Rubén Blasco, Álvaro Marco, Roberto Casas, Alejando Ibarz, Victorián Coarasa, and Ángel Asensio

Managing Ambient Intelligence Sensor Network Systems, an Agent Based Approach 1121
Guillermo Bosch and Cristian Barrué

Ambulatory Mobility Characterization Using Body Inertial Systems: An Application to Fall Detection 1129
Marc Torrent, Alan Bourke, Xavier Parra, and Andreu Català

User Daily Activity Classification from Accelerometry Using Feature Selection and SVM 1137
Jordi Parera, Cecilio Angulo, A. Rodríguez-Molinero, and Joan Cabestany

A Metrics Review for Performance Evaluation on Assisted Wheelchair Navigation 1145
Cristina Urdiales, Jose M. Peula, Ulises Cortés, Christian Barrué, Blanca Fernández-Espejo, Roberta Annichiaricco, Francisco Sandoval, and Carlo Caltagirone

Conventional Joystick vs. *Wiimote* for Holonomic Wheelchair Control 1153
L. Duran, M. Fernandez-Carmona, C. Urdiales, J.M Peula, and F. Sandoval

Normal versus Pathological Cognitive Aging: Variability as a Constraint of Patients Profiling for AmI Design 1161
Luiza Spiru, Camelia Ghita, Ileana Turcu, Lucian Stefan, Ioana Ioancio, Costin Nuta, Mona Blaciotti, Mariana Martin, Ulises Cortes, and Roberta Annicchiarico

15. Other Applications

Estimating the Embedding Dimension Distribution of Time Series with SOMOS 1168
Pedro J. Zufiria and Pascual Campoy

Training Methods and Analysis of Composite, Evolved, On-Line Networks for Time Series Prediction 1176
Russell Y. Webb

Special Time Series Prediction: Creep of Concrete 1184
Juan L. Pérez, Fernando Martínez Abella, Alba Catoira, and Javier Berrocal

Artificial Neural Networks in Urban Runoff Forecast 1192
Mónica Miguélez, Jerónimo Puertas, and Juan Ramón Rabuñal

A Secret Sharing Scheme for Digital Images Based on Cellular Automata and Boolean Functions 1200
Ángel Martín del Rey and Gerardo Rodríguez Sánchez

Shapes Description by a Segments-Based Neural Network 1208
J.A. Gómez-Ruiz, J. Muñoz-Perez, M.A. García-Bernal

Protecting DCT Templates for a Face Verification System by Means of Pseudo-random Permutations 1216
Marco Grassi and Marcos Faundez-Zanuy

Efficient Parallel Feature Selection for Steganography Problems 1224
Alberto Guillén, Antti Sorjamaa, Yoan Miche, Amaury Lendasse, and Ignacio Rojas

Mobile Applications: MovilPIU and Mobiblio 1232
Roberto Berjón Gallinas, M. Encarnación Beato Gutiérrez, Montserrat Mateos Sánchez, Miguel Ángel Sánchez Vidales, and Ana Fermoso García

A Case Study of a Pull WAP Location-Based Service Incorporating Maps Services 1240
Montserrat Mateos Sanchez, Roberto Berjon Gallinas, Miguel Angel Sanchez Vidales, Encarnacion Beato Gutierrez, and Ana Fermoso Garcia

A Mobile Tourist Decision Support System for Small Footprint Devices 1248
Wouter Souffiau, Joris Maervoet, Pieter Vansteenwegen, Greet Vanden Berghe, and Dirk Van Oudheusden

Stereo-MAS: Multi-Agent System for Image Stereo Processing 1256
Sara Rodríguez, Juan F. De Paz, Javier Bajo, Dante I. Tapia, and Belén Pérez

Participatory EHPR: A Watermarking Solution 1264
David Lowe and B.R. Matam

Bus Network Scheduling Problem: GRASP + EAs with PISA * Simulation 1272
Ana C. Olivera, Mariano Frutos, Jessica A. Carballido, Ignacio Ponzoni, and Nélida B. Brignole

Wine Classification with Gas Sensors Combined with Independent Component Analysis and Neural Networks 1280
Jesús Lozano, Antonio García, Carlos J. García, Fernándo Alvarez, and Ramón Gallardo

Experiments and Reference Models in Training Neural Networks for Short-Term Wind Power Forecasting in Electricity Markets 1288
Juan Méndez, Javier Lorenzo, and Mario Hernández

Intrusion Detection Method Using Neural Networks Based on the Reduction of Characteristics 1296
Iren Lorenzo-Fonseca, Francisco Maciá-Pérez, Francisco José Mora-Gimeno, Rogelio Lau-Fernández, Juan Antonio Gil-Martínez-Abarca, and Diego Marcos-Jorquera

Evaluating the Performance of the Multilayer Perceptron as a Data Editing Tool 1304
M^a-Dolores Cubiles-de-la-Vega, Esther-Lydia Silva-Ramírez, Rafael Pino-Mejías, and Manuel López-Coello

A.N.N. Based Approach to Mass Biometry Taking Advantage from Modularity 1312
Kurosh Madani, Abdennasser Chebira, and Véronique Amarger

Thresholded Neural Networks for Sensitive Industrial Classification Tasks 1320
Marco Vannucci, Valentina Colla, Mirko Sgarbi, and Orlando Toscanelli

ANN Based Solutions: It Is Time to Defeat Real-World and Industrial Dilemmas 1328
Kurosh Madani, Véronique Amarger, and Christophe Sabourin

Pollution Alarm System in Mexico 1336
M.G. Cortina-Januchs, J.M. Barrón-Adame, A. Vega-Corona, and D. Andina

Author Index 1345

Table of Contents – Part II

Neuro-control and Its Applications to Electric Vehicle Control 1
Sigeru Omatu

1. Multi-agent Systems I

Multi-agent Data Fusion Architecture Proposal for Obtaining an Integrated Navigated Solution on UAV's 13
José Luis Guerrero, Jesús García, and José Manuel Molina

Towards a Multiagent Approach for the VERDINO Prototype 21
Evelio J. González, Leopoldo Acosta, Alberto Hamilton, Jonatán Felipe, Marta Sigut, Jonay Toledo, and Rafael Arnay

BDI Planning Approach to Distributed Multiagent Based Semantic Search Engine 25
Mehta Shikha, Banati Hema, and Bedi Punam

Methodology vs. Development Process: A Case Study for AOSE 29
Alma Gómez-Rodríguez and Juan C. González-Moreno

2. New Algorithms and Applications

Designing Radial Basis Function Neural Networks with Meta-evolutionary Algorithms: The Effect of Chromosome Codification 37
Elisabet Parras-Gutierrez, Victor M. Rivas, M. Jose del Jesus, and Juan J. Merelo

Hyperheuristics for a Dynamic-Mapped Multi-Objective Island-Based Model 41
Coromoto León, Gara Miranda, and Carlos Segura

High Level Abstractions for Improving Parallel Image Reconstruction Algorithms 50
Jose A. Álvarez and Javier Roca Piera

A Group k-Mutual Exclusion Algorithm for Mobile Ad Hoc Networks 58
Ousmane Thiare and Mohamed Naimi

3. Semantic, Ontologies

Boosting Annotated Web Services in SAWSDL 67
Antonio J. Roa-Valverde, Jorge Martinez-Gil, and José F. Aldana-Montes

Creation of Semantic Overlay Networks Based on Personal Information 75
Alberto García-Sola and Juan A. Botia

Adding an Ontology to a Standardized QoS-Based MAS Middleware ... 83
José L. Poza, Juan L. Posadas, and José E. Simó

OntologyTest: A Tool to Evaluate Ontologies through Tests Defined by the User 91
Sara García-Ramos, Abraham Otero, and Mariano Fernández-López

4. Distributed Systems I

A Case Study in Distributing a SystemC Model..................... 99
V. Galiano, M. Martínez, H. Migallón, D. Pérez-Caparrós, and C. Quesada

A Snapshot Algorithm for Mobile Ad Hoc Networks 107
Dan Wu, Chi Hong Cheong, and Man Hon Wong

Introducing a Distributed Architecture for Heterogeneous Wireless Sensor Networks 116
Dante I. Tapia, Ricardo S. Alonso, Juan F. De Paz, and Juan M. Corchado

OCURO: Estimation of Space Occupation and Vehicle Rotation in Controlled Parking Areas 124
Julián Lamas-Rodríguez, Juan Arias, José R.R. Viqueira, and José Varela

5. Multi-agent System II

A Distributed Architectural Strategy towards Ambient Intelligence 130
Maria J. Santofimia, Francisco Moya, Felix J. Villanueva, David Villa, and Juan C. Lopez

Reviewing the Use of Requirements Engineering Techniques in the Development of Multi-Agent Systems..................... 134
David Blanes, Emilio Insfran, and Silvia Abrahão

Testing in Agent Oriented Methodologies 138
Mailyn Moreno, Juan Pavón, and Alejandro Rosete

Composition of Temporal Bounded Services in Open MAS 146
Elena del Val, Miguel Rebollo, and Vicente Botti

Organizational-Oriented Methodological Guidelines for Designing Virtual Organizations 154
E. Argente, V. Botti, and V. Julian

6. Genetic Algorithms

Pervasive Evolutionary Algorithms on Mobile Devices 163
Pablo Garcia-Sanchez, Juan P. Sevilla, Juan J. Merelo, Antonio M. Mora, Pedro A. Castillo, Juan L.J. Laredo, and Francisco Casado

A New Method for Simplifying Algebraic Expressions in Genetic Programming Called Equivalent Decision Simplification 171
Mori Naoki, Bob McKay, Nguyen Xuan, Essam Daryl, and Saori Takeuchi

A Hybrid Differential Evolution Algorithm for Solving the Terminal Assignment Problem 179
Eugénia Moreira Bernardino, Anabela Moreira Bernardino, Juan Manuel Sánchez-Pérez, Juan Antonio Gómez-Pulido, and Miguel Angel Vega-Rodríguez

An Iterative GASVM-Based Method: Gene Selection and Classification of Microarray Data 187
Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris, and Michifumi Yoshioka

Privacy-Preserving Distributed Learning Based on Genetic Algorithms and Artificial Neural Networks 195
Bertha Guijarro-Berdiñas, David Martínez-Rego, and Santiago Fernández-Lorenzo

7. Real Time and Parallel Systems

Development of a Camera-Based Portable Automatic Inspection System for Printed Labels Using Neural Networks 203
Yuhki Shiraishi and Fumiaki Takeda

Towards Compositional Verification in MEDISTAM-RT Methodological Framework 211
Kawtar Benghazi, Miguel J. Hornos, and Manuel Noguera

Universal Global Optimization Algorithm on Shared Memory Multiprocessors 219
Juana L. Redondo, Inmaculada García, and Pilar Martínez-Ortigosa

Efficiency Analysis of Parallel Batch Pattern NN Training Algorithm on General-Purpose Supercomputer 223
Volodymyr Turchenko and Lucio Grandinetti

Evaluation of Master-Slave Approaches for 3D Reconstruction in Electron Tomography 227
M. Laura da Silva, Javier Roca-Piera, and José-Jesús Fernández

General Purpose Agent-Based Parallel Computing 232
David Sánchez, David Isern, Ángel Rodríguez, and Antonio Moreno

8. Neural Networks

VS-Diagrams Identification and Classification Using Neural Networks ... 240
Daniel Gómez, Eduardo J. Moya, Enrique Baeyens, and Clemente Cárdenas

Visual Surveillance of Objects Motion Using GNG 244
José García-Rodríguez, Francisco Flórez-Revuelta, and Juan Manuel García-Chamizo

Forecasting the Price Development of Crude Oil with Artificial Neural Networks 248
Richard Lackes, Chris Börgermann, and Matthias Dirkmorfeld

Invariant Features from the Trace Transform for Jawi Character Recognition 256
Mohammad Faidzul Nasrudin, Khairuddin Omar, Choong-Yeun Liong, and Mohamad Shanudin Zakaria

An Ensemble Based Translator for Natural Languages 264
Gustavo A. Casañ and Mª. Asunción Castaño

Verification of the Effectiveness of the Online Tuning System for Unknown Person in the Awaking Behavior Detection System 272
Hironobu Satoh and Fumiaki Takeda

9. Models for Soft Computing, Applications and Advances

An Evolutionary Algorithm for the Surface Structure Problem 280
J. Martínez, M.F. López, J.A. Martín-Gago, and V. Martín

In Premises Positioning – Fuzzy Logic 284
Rubén González Crespo, Gloria García Fernandez, Oscar Sanjuán Martínez, Vicente García-Díaz, Luis Joyanes Aguilar, and Enrique Torres Franco

GIS Applications Use in Epidemiology GIS-EPI 292
Rubén González Crespo, Gloria García Fernandez, Daniel Zapico Palacio, Enrique Torres Franco, Andres Castillo Sanz, and Cristina Pelayo García-Bustelo

TALISMAN MDE Framework: An Architecture for Intelligent Model-Driven Engineering 299
Vicente García-Díaz, Jose Barranquero Tolosa, B. Cristina Pelayo G-Bustelo, Elías Palacios-González, Óscar Sanjuan-Martínez, and Rubén González Crespo

Electronic Nose System by Neural Networks 307
Sigeru Omatu, Michifumi Yoshioka, and Kengo Matsuyama

Towards Meta-model Interoperability of Models through Intelligent Transformations 315
José Barranquero Tolosa, Vicente García-Díaz, Oscar Sanjuán-Martínez, Héctor Fernández-Fernández, and Gloria García-Fernández

MDE for Device Driver Development 323
Gloria García Fernández, Óscan Sanjuan-Martinez, Rubén González Crespo, Cristina Pelayo García-Bustelo, and José Barranquero Tolosa

Image/Video Compression with Artificial Neural Networks 330
Daniel Zapico Palacio, Rubén González Crespo, Gloria García Fernández, and Ignacio Rodríguez Novelle

10. New Intelligent and Distributed Computing Solutions for Manufacturing Systems

A Distributed Intelligent Monitoring System Applied to a Micro-scale Turning Process 338
Raúl M. del Toro, Rodolfo E. Haber, and Michael Schmittdiel

Simulation of Dynamic Supply Chain Configuration Based on Software Agents and Graph Theory 346
Arkadiusz Kawa

Use of Distributed IT Tools for Assessment of Manufacturing Processes 350
Pawel Pawlewski and Zbigniew J. Pasek

Emerging Trends in Manufacturing Systems Management – IT Solutions 358
Marek Fertsch, Pawel Pawlewski, and Paulina Golińska

Engineering Web Service Markets for Federated Business Applications 366
Nico Brehm and Paulina Golinska

Implication of Reasoning in GRAIXPERT for Modeling Enterprises 374
Paul-Eric Dossou and Philip Mitchell

The Concept of an Agent-Based System for Planning of Closed Loop Supplies in Manufacturing System 382
Paulina Golinska

Application of Distributed Techniques for Resources Modeling and Capacity Management 390
Agnieszka Stachowiak and Paweł Pawlewski

11. Development Metodologies of Web Service Systems

Web-Based Membership Registration System of Japan Volleyball Association 397
Hiroaki Oiso, Ayako Hiramatsu, Norhisa Komoda, Akira Ito, Toshiro Endo, and Yasumi Okayama

A Web Application Development Framework Using Code Generation from MVC-Based UI Model 404
Keisuke Watanabe, Makoto Imamura, Katsushi Asami, and Toshiyuki Amanuma

The System Enhancement Method for Combining a Legacy Client-Server System and a Web Based New System 412
Junichiro Sueishi and Hiroshi Morihisa

An Empirical Study of an Extended Technology Acceptance Model for Online Video Services 416
Ayako Hiramatsu, Takahiro Yamasaki, and Kazuo Nose

12. Applications I

A Post-optimization Method to Improve the Ant Colony System Algorithm 424
M.L. Pérez-Delgado and J. Escuadra Burrieza

From the Queue to the Quality of Service Policy: A Middleware Implementation 432
José L. Poza, Juan L. Posadas, and José E. Simó

Planning with Uncertainty in Action Outcomes as Linear Programming Problem 438
Adam Galuszka and Andrzej Holdyk

An Optimized Ant System Approach for DNA Sequence Optimization 446
Tri Basuki Kurniawan, Zuwairie Ibrahim, Noor Khafifah Khalid, and Marzuki Khalid

Implementation of Binary Particle Swarm Optimization for DNA Sequence Design 450
Noor Khafifah Khalid, Zuwairie Ibrahim, Tri Basuki Kurniawan, Marzuki Khalid, and Andries P. Engelbrecht

13. Distributed Systems II

Multi-colony ACO and Rough Set Theory to Distributed Feature Selection Problem 458
Yudel Gómez, Rafael Bello, Ann Nowé, Enrique Casanovas, and J. Taminau

Improving the Performance of Bandwidth-Demanding Applications by a Distributed Network Interface 462
Andres Ortiz, Julio Ortega, Antonio F. Diaz, and Alberto Prieto

Agrega: A Distributed Repository Network of Standardised Learning Objects 466
Antonio Sarasa, Jose Manuel Canabal, and Juan Carlos Sacristán

DIAMI: Distributed Intelligent Environment for Blind Musicians 475
José E. Díaz, Juan L. Márquez, Miguel Sánchez, José M. Sánchez-Aguilera, Miguel A. Sánchez, and Javier Bajo

14. Data Mining and Data Classification

Design of a Decision Support System for Classification of Natural Risk in Maritime Construction Based on Temporal Windows 483
Marco Antonio García Tamargo, Alfredo S. Alguero García, Andrés Alonso Quintanilla, Amelia Bilbao Terol, and Víctor Castro Amigo

Using Data-Mining for Short-Term Rainfall Forecasting 487
David Martínez Casas, José Ángel Taboada González, Juan Enrique Arias Rodríguez, and José Varela Pet

An Integrated Solution to Store, Manage and Work with Datasets Focused on Metadata in the Retelab Grid Project 491
David Mera, José M. Cotos, Joaquín A. Trinanes, and Carmen Cotelo

An Improved Binary Particle Swarm Optimisation for Gene Selection in Classifying Cancer Classes ... 495
Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris, Michifumi Yoshioka, and Anazida Zainal

15. Applications II

A Computer Virus Spread Model Based on Cellular Automata on Graphs ... 503
Angel Martín del Rey

Rank-Based Ant System to Solve the Undirected Rural Postman Problem ... 507
María Luisa Pérez-Delgado

Design of a Snort-Based Hybrid Intrusion Detection System ... 515
J. Gómez, C. Gil, N. Padilla, R. Baños, and C. Jiménez

Flexible Layered Multicasting Method for Multipoint Video Conference in Heterogeneous Access Environment ... 523
Hideki Tode, Kanako Uchida, and Koso Murakami

Modular and Scalable Multi-interface Data Acquisition Architecture Design for Energy Monitoring in Fishing Vessels ... 531
Sebastián Villarroya, M^a. Jesús L. Otero, Luís Romero, José M. Cotos, and Víctor Pita

Validator for Clinical Practice Guidelines ... 539
Fernando Pech-May, Ivan Lopez-Arevalo, and Victor Sosa-Sosa

16. Knowledge Discovery, Reasoning, Meta-Learning

Using Gaussian Processes in Bayesian Robot Programming ... 547
Fidel Aznar, Francisco A. Pujol, Mar Pujol, and Ramón Rizo

Optimising Machine-Learning-Based Fault Prediction in Foundry Production ... 554
Igor Santos, Javier Nieves, Yoseba K. Penya, and Pablo G. Bringas

Optimizing the Use of an Integrated LMS: Hardware Evolution through Distributed Computing. Experience from the Universitat de València ... 562
Paloma Moreno-Clari, Sergio Cubero-Torres, and Agustín López-Bueno

A Process Model for Group Decision Making with Quality Evaluation ... 566
Luís Lima, Paulo Novais, and José Bulas Cruz

Abstract Models for Redesign of Technical Processes 574
Ivan Lopez-Arevalo, Victor Sosa-Sosa, and Edgar Tello-Leal

Towards a Support for Autonomous Learning Process 582
Lorenzo Moreno, Evelio J. González, Carina S. González, and J.D. Piñeiro

17. Applications III

DNA Electrophoresis Simulation 586
Andrés de la Peña, Francisco J. Cisneros, Ángel Goñi, and Juan Castellanos

Classification of Fatigue Bill Based on Support Vector Machine by Using Acoustic Signal 590
Dongshik Kang, Masaki Higa, Nobuo Shoji, Masanobu Fujita, and Ikugo Mitsui

Artificial Ants and Packaging Waste Recycling 596
María Luisa Pérez-Delgado

Analysis of Geometric Moments as Features for Identification of Forensic Ballistics Specimen 604
Nor Azura Md Ghani, Choong-Yeun Liong, and Abdul Aziz Jemain

18. Commnications and Image Processing

Colour Image Compression Based on the Embedded Zerotree Wavelet 612
Francisco A. Pujol, Higinio Mora, Antonio Jimeno, and José Luis Sánchez

Camera Calibration Method Based on Maximum Likelihood Estimation 616
Michifumi Yoshioka and Sigeru Omatu

Neural Networks Applied to Fingerprint Recognition 621
Angélica González Arrieta, Griselda Cobos Estrada, Luis Alonso Romero, and Ángel Luis Sánchez Lázaro y Belén Pérez Lancho

Wireless Communications Architecture for "Train-to-Earth" Communication in the Railway Industry 626
Itziar Salaberria, Roberto Carballedo, Unai Gutierrez, and Asier Perallos

Emergence of Communication in Foraging Behavior 634
Siavash Kayal, Alireza Chakeri, Abdol Hossein Aminaiee, and Caro Lucas

19. Data/Information Management on Large-Scale Distributed Environments

WiFi Location Information System for Both Indoors and Outdoors 638
Nobuo Kawaguchi

A Peer-to-Peer Information Sharing Method for RDF Triples Based on RDF Schema 646
Kohichi Kohigashi, Kentaro Takahashi, Kaname Harumoto, and Shojiro Nishio

Toward Virtual Machine Packing Optimization Based on Genetic Algorithm 651
Hidemoto Nakada, Takahiro Hirofuchi, Hirotaka Ogawa, and Satoshi Itoh

MetaFa: Metadata Management Framework for Data Sharing in Data-Intensive Applications 655
Minoru Ikebe, Atsuo Inomata, Kazutoshi Fujikawa, and Hideki Sunahara

Design and Implementation of Wireless LAN System for Airship 659
Hideki Shimada, Minoru Ikebe, Yuki Uranishi, Masayuki Kanbara, Hideki Sunahara, and Naokazu Yokoya

20. Home Care Applications 1

Heterogeneous Wireless Sensor Networks in a Tele-monitoring System for Homecare 663
Ricardo S. Alonso, Óscar García, Alberto Saavedra, Dante I. Tapia, Juan F. de Paz, and Juan M. Corchado

BIOHOME: A House Designed for Assisted Living 671
Begoña García, Ibon Ruiz, Javier Vicente, and Amaia Méndez

Supervision and Access Control System for Disabled Person's Homes ... 675
Lara del Val, María I. Jiménez, Alberto Izquierdo, Juan J. Villacorta, David Rodriguez, Ramón de la Rosa, and Mariano Raboso

An Intelligent Agents Reasoning Platform to Support Smart Home Telecare 679
Miguel A. Valero, Laura Vadillo, Iván Pau, and Ana Peñalver

21. Home Care Applications 2

Multimodal Classification of Activities of Daily Living Inside Smart Homes 687
Vit Libal, Bhuvana Ramabhadran, Nadia Mana, Fabio Pianesi, Paul Chippendale, Oswald Lanz, and Gerasimos Potamianos

Modular Framework for Smart Home Applications 695
Javier Blesa, Pedro Malagón, Álvaro Araujo, José M. Moya, Juan Carlos Vallejo, Juan-Mariano de Goyeneche, Elena Romero, Daniel Villanueva, and Octavio Nieto-Taladriz

Ambient Information Systems for Supporting Elder's Independent Living at Home 702
Juan P. Garcia-Vazquez, Marcela D. Rodriguez, and Angel G. Andrade

A Centralized Approach to an Ambient Assisted Living Application: An Intelligent Home 706
Nayat Sánchez-Pi and José Manuel Molina

22. Medical Applications

A Web Based Information System for Managing and Improving Care Services in Day Centres 710
José A. Alvarez, Dolores M. Hernández-Capel, and Luis J. Belmonte

Web Application and Image Analysis Tool to Measure and Monitoring the Density in Bone Fractures with a Predictive Approach 718
B. Rosario Campomanes Álvarez, Ángel Martínez Nistal, José Paz Jiménez, Marco A. García Tamargo, Alfredo S. Alguero García, and José Paz Aparicio

Virtual Center for the Elderly: Lessons Learned 722
Laura M. Roa, Javier Reina-Tosina, and Miguel A. Estudillo

Remote Health Monitoring: A Customizable Product Line Approach ... 727
Miguel A. Laguna, Javier Finat, and José A. González

A Memory Management System towards Cognitive Assistance of Elderly People 735
Fouad Khelifi, Jianmin Jiang, and Paul Trundle

23. Adaptable Models

Building Self-adaptive Services for Ambient Assisted Living 740
Pau Giner, Carlos Cetina, Joan Fons, and Vicente Pelechano

User Configuration of Activity Awareness 748
Tony McBryan and Philip Gray

Low-Cost Gesture-Based Interaction for Intelligent Environments 752
José M. Moya, Ainhoa Montero de Espinosa, Álvaro Araujo, Juan-Mariano de Goyeneche, and Juan Carlos Vallejo

HERMES: Pervasive Computing and Cognitive Training for Ageing Well 756
Cristina Buiza, John Soldatos, Theodore Petsatodis, Arjan Geven, Aitziber Etxaniz, and Manfred Tscheligi

An Ambient Intelligent Approach to Control Behaviours on a Tagged World 764
María Ros, Miguel Delgado, and Amparo Vila

Adaptive Interfaces for People with Special Needs 772
Pablo Llinás, Germán Montoro, Manuel García-Herranz, Pablo Haya, and Xavier Alamán

24. AI Techniques

Human Memory Assistance through Semantic-Based Text Processing ... 780
Paul R. Trundle and Jianmin Jiang

Case-Based Reasoning Decision Making in Ambient Assisted Living 788
Davide Carneiro, Paulo Novais, Ricardo Costa, and José Neves

Activity Recognition from Accelerometer Data on a Mobile Phone 796
Tomas Brezmes, Juan-Luis Gorricho, and Josep Cotrina

Image Processing Based Services for Ambient Assistant Scenarios 800
Elena Romero, Álvaro Araujo, José M. Moya, Juan-Mariano de Goyeneche, Juan Carlos Vallejo, Pedro Malagón, Daniel Villanueva, and David Fraga

25. Applied Technologies 1

Outdoors Monitoring of Elderly People Assisted by Compass, GPS and Mobile Social Network 808
Roberto Calvo-Palomino, Pedro de las Heras-Quirós, José Antonio Santos-Cadenas, Raúl Román-López, and Daniel Izquierdo-Cortázar

Biometric Access Control System for AAL 812
Begoña García, Amaia Méndez, Ibon Ruiz, and Javier Vicente

Detecting Domestic Problems of Elderly People: Simple and Unobstrusive Sensors to Generate the Context of the Attended 819
Juan A. Botia, Ana Villa, Jose T. Palma, David Pérez, and Emilio Iborra

A Wireless Infrastructure for Assisting the Elderly and the Mobility Impaired 827
J. Antonio García-Macías, Luis E. Palafox, and Ismael Villanueva

26. Applied Technologies 2

A Device Search Strategy Based on Connections History for Patient Monitoring 831
José-Alfredo Abad and Juan-Luis Gorricho

A Robot Controlled by Blinking for Ambient Assisted Living 839
Alonso A. Alonso, Ramón de la Rosa, Lara del Val, María I. Jiménez, and Samuel Franco

Service-Oriented Device Integration for Ubiquitous Ambient Assisted Living Environments 843
Javier Andréu Pérez, Juan Antonio Álvarez, Alejandro Fernández-Montes, and Juan Antonio Ortega

Variabilities of Wireless and Actuators Sensor Network Middleware for Ambient Assisted Living 851
Flávia C. Delicato, Lidia Fuentes, Nadia Gámez, and Paulo F. Pires

Technological Solution for Independent Living of Intellectual Disabled People 859
Ibon Ruiz, Begoña García, and Amaia Méndez

27. Frameworks and Platforms

The UVa-Neuromuscular Training System Platform 863
Ramón de la Rosa, Sonia de la Rosa, Alonso Alonso, and Lara del Val

A Proposal for Mobile Diabetes Self-control: Towards a Patient Monitoring Framework 870
Vladimir Villarreal, Javier Laguna, Silvia López, Jesús Fontecha, Carmen Fuentes, Ramón Hervás, Diego López de Ipiña, and Jose Bravo

ALADDIN, A Technology pLatform for the Assisted Living of Dementia elDerly INdividuals and Their Carers 878
Konstantinos Perakis, Maria Haritou, and Dimitris Koutsouris

An Architecture for Ambient Assisted Living and Health Environments 882
Antonio J. Jara, Miguel A. Zamora, and Antonio F.G. Skarmeta

28. Theoretical Approaches

Shape Memory Fabrics to Improve Quality Life to People with Disability (PWD) 890
Juan C. Chicote

Ontologies for Intelligent e-Therapy: Application to Obesity 894
Irene Zaragozá, Jaime Guixeres, and Mariano Alcañiz

A Contribution for Elderly and Disabled Care Using Intelligent Approaches 902
Gabriel Fiol-Roig and Margaret Miró-Julià

Quality of Life Evaluation of Elderly and Disabled People by Using Self-Organizing Maps 906
Antonio Bono-Nuez, Bonifacio Martín-del-Brío, Rubén Blasco-Marín, Roberto Casas-Nebra, and Armando Roy-Yarza

Analysis and Design of an Object Tracking Service for Intelligent Environments 914
Ignacio Recio, José M. Moya, Álvaro Araujo, Juan Carlos Vallejo, and Pedro Malagón

Using Business Process Modelling to Model Integrated Care Processes: Experiences from a European Project 922
Ingrid Svagård and Babak A. Farshchian

29. Text Mining

Classification of MedLine Documents Using MeSH Terms 926
Daniel Glez-Peña, Sira López, Reyes Pavón, Rosalía Laza, Eva L. Iglesias, and Lourdes Borrajo

GREAT: Gene Regulation EvAluation Tool 930
Catia Machado, Hugo Bastos, and Francisco Couto

Identifying Gene Ontology Areas for Automated Enrichment 934
Catia Pesquita, Tiago Grego, and Francisco Couto

Identification of Chemical Entities in Patent Documents 942
Tiago Grego, Piotr Pęzik, Francisco M. Couto, and Dietrich Rebholz-Schuhmann

Applying Text Mining to Search for Protein Patterns 950
Pablo V. Carrera, Daniel Glez-Peña, Eva L. Iglesias, Lourdes Borrajo, Reyes Pavón, Rosalía Laza, and Carmen M. Redondo

Biomedical Text Mining Applied to Document Retrieval and Semantic Indexing 954
Anália Lourenço, Sónia Carneiro, Eugénio C. Ferreira, Rafael Carreira, Luis M. Rocha, Daniel Glez-Peña, José R. Méndez, Florentino Fdez-Riverola, Fernando Diaz, Isabel Rocha, and Miguel Rocha

30. Microarrays

CBR System with Reinforce in the Revision Phase for the Classification of CLL Leukemia 964
Juan F. De Paz, Sara Rodríguez, Javier Bajo, and Juan M. Corchado

An Evolutionary Approach for Sample-Based Clustering on Microarray Data 972
Daniel Glez-Peña, Fernando Díaz, José R. Méndez, Juan M. Corchado, and Florentino Fdez-Riverola

EDA-Based Logistic Regression Applied to Biomarkers Selection in Breast Cancer 979
Santiago González, Victor Robles, Jose Maria Peña, and Oscar Cubo

Oligonucleotide Microarray Probe Correction by FixedPoint ICA Algorithm 988
Raul Malutan, Pedro Gómez, and Monica Borda

31. Cluster

Group Method of Documentary Collections Using Genetic Algorithms 992
José Luis Castillo S., José R. Fernández del Castillo, and León González Sotos

Partitional Clustering of Protein Sequences – An Inductive Logic Programming Approach 1001
Nuno A. Fonseca, Vitor S. Costa, Rui Camacho, Cristina Vieira, and Jorge Vieira

Segregating Confident Predictions of Chemicals' Properties for Virtual Screening of Drugs 1005
Axel J. Soto, Ignacio Ponzoni, and Gustavo E. Vazquez

Efficient Biclustering Algorithms for Time Series Gene Expression Data Analysis 1013
Sara C. Madeira and Arlindo L. Oliveira

32. Pattern Recognition

Robust Association of Pathological Respiratory Events in SAHS Patients: A Step towards Mining Polysomnograms 1020
Abraham Otero and Paulo Félix

Population Extinction in Genetic Algorithms: Application in Evolutionary Studies 1028
Antonio Carvajal-Rodríguez and Fernando Carvajal-Rodríguez

Tabu Search for the Founder Sequence Reconstruction Problem: A Preliminary Study 1035
Andrea Roli and Christian Blum

Visually Guiding and Controlling the Search While Mining Chemical Structures 1043
Max Pereira, Vítor Santos Costa, Rui Camacho, and Nuno A. Fonseca

Analysing the Evolution of Repetitive Strands in Genomes 1047
José P. Lousado, José Luis Oliveira, Gabriela R. Moura, and Manuel A.S. Santos

33. Systems Biology

A SIS Epidemiological Model Based on Cellular Automata on Graphs 1055
María J. Fresnadillo, Enrique García, José E. García, Ángel Martín, and Gerardo Rodríguez

A Critical Review on Modelling Formalisms and Simulation Tools in Computational Biosystems 1063
Daniel Machado, Rafael S. Costa, Miguel Rocha, Isabel Rocha, Bruce Tidor, and Eugénio C. Ferreira

A Software Tool for the Simulation and Optimization of Dynamic Metabolic Models 1071
Pedro Evangelista, Isabel Rocha, Eugénio C. Ferreira, and Miguel Rocha

Large Scale Dynamic Model Reconstruction for the Central Carbon Metabolism of *Escherichia coli* 1079
Rafael S. Costa, Daniel Machado, Isabel Rocha, and Eugénio C. Ferreira

34. Bioinformatic Applications

Intuitive Bioinformatics for Genomics Applications: Omega-Brigid Workflow Framework 1084
David Díaz, Sergio Gálvez, Juan Falgueras, Juan Antonio Caballero, Pilar Hernández, Gonzalo Claros, and Gabriel Dorado

Current Efforts to Integrate Biological Pathway Information 1092
Daniel Glez-Peña, Rubén Domínguez, Gonzalo Gómez-López, David G. Pisano, and Florentino Fdez-Riverola

BioCASE: Accelerating Software Development of Genome-Wide Filtering Applications 1097
Rosana Montes and María M. Abad-Grau

DynamicFlow: A Client-Side Workflow Management System 1101
Pedro Lopes, Joel Arrais, and José Luís Oliveira

Bayesian Joint Estimation of CN and LOH Aberrations 1109
Paola M.V. Rancoita, Marcus Hutter, Francesco Bertoni, and Ivo Kwee

Development of a Workflow for Protein Sequence Analysis Based on the Taverna Workbench® Software 1118
Mariana B. Monteiro, Manuela E. Pintado, Francisco X. Malcata, Conrad Bessant, and Patrícia R. Moreira

Automatic Prediction of the Genetic Code 1125
Mateus Patricio, Jaime Huerta-Cepas, Toni Gabaldón, Rafael Zardoya, and David Posada

35. Phylogenetic

Computational Challenges on Grid Computing for Workflows Applied to Phylogeny 1130
Raúl Isea, Esther Montes, Antonio J. Rubio-Montero, and Rafael Mayo

ZARAMIT: A System for the Evolutionary Study of Human Mitochondrial DNA 1139
Roberto Blanco and Elvira Mayordomo

A First Insight into the *In Silico* Evaluation of the Accuracy of AFLP Markers for Phylogenetic Reconstruction 1143
María Jesús García-Pereira, Humberto Quesada, and Armando Caballero

A Method to Compare MALDI—TOF MS PMF Spectra and Its Application in Phyloproteomics 1147
Ignacio Ortea, Lorena Barros, Benito Cañas, Pilar Calo-Mata, Jorge Barros-Velázquez, and José M. Gallardo

36. Proteins

A Screening Method for Z-Value Assessment Based on the Normalized Edit Distance 1154
Guillermo Peris and Andrés Marzal

On the Bond Graphs in the Delaunay-Tetrahedra of the Simplicial Decomposition of Spatial Protein Structures 1162
Rafael Ördög and Vince Grolmusz

A New Model of Synthetic Genetic Oscillator Based on *Trans*-Acting Repressor Ribozyme 1170
Jesús M. Miró Bueno and Alfonso Rodríguez-Patón

Efficient Exact Pattern-Matching in Proteomic Sequences 1178
Sérgio Deusdado and Paulo Carvalho

Iterative Lattice Protein Design Using Template Matching 1187
David Olivieri

37. Soco.1

Rotor Imbalance Detection in Gas Turbines Using Fuzzy Sets 1195
Ilaria Bertini, Alessandro Pannicelli, Stefano Pizzuti, Paolo Levorato, and Riccardo Garbin

Practical Application of a KDD Process to a Sulphuric Acid Plant 1205
Victoria Pachón, Jacinto Mata, and Manuel J. Maña

Heat Consumption Prediction with Multiple Hybrid Models 1213
Maciej Grzenda and Bohdan Macukow

38. Soco.2

Multi-Objective Particle Swarm Optimization Design of PID Controllers 1222
P.B. de Moura Oliveira, E.J. Solteiro Pires, J. Boaventura Cunha, and Damir Vrančić

Design of Radio-Frequency Integrated CMOS Discrete Tuning Varactors Using the Particle Swarm Optimization Algorithm 1231
E.J. Solteiro Pires, Luís Mendes, P.B. de Moura Oliveira, J.A. Tenreiro Machado, João C. Vaz, and Maria J. Rosário

Algorithms for Active Noise Control 1240
M. Dolores Redel-Macías, Antonio J. Cubero-Atienza, Paul Sas, and Lorenzo Salas-Morera

39. Soco.3

License Plate Detection Using Neural Networks 1248
Luis Carrera, Marco Mora, José Gonzalez, and Francisco Aravena

Control of Mobile Robot Considering Actuator Dynamics with Uncertainties in the Kinematic and Dynamic Models 1256
Nardênio A. Martins, Douglas W. Bertol, Edson R. De Pieri, and Eugênio B. Castelan

Data Mining for Burr Detection (in the Drilling Process) 1264
Susana Ferreiro, Ramón Arana, Gotzone Aizpurua, Gorka Aramendi, Aitor Arnaiz, and Basilio Sierra

A Neural Recognition System for Manufactured Objects 1274
Rafael M. Luque, Enrique Dominguez, Esteban J. Palomo, and Jose Muñoz

A Soft Computing System to Perform Face Milling Operations 1282
Raquel Redondo, Pedro Santos, Andres Bustillo, Javier Sedano, José Ramón Villar, Maritza Correa, José Ramón Alique, and Emilio Corchado

Author Index 1293

Lower Bounds for Approximation of Some Classes of Lebesgue Measurable Functions by Sigmoidal Neural Networks

José L. Montaña and Cruz E. Borges

Departamento de Matemáticas, Estadística y Computación,
Universidad de Cantabria, 39005 Santander, Spain
{montanjl,borgesce}@unican.es

Abstract. We propose a general method for estimating the distance between a compact subspace K of the space $L^1([0,1]^s)$ of Lebesgue measurable functions defined on the hypercube $[0,1]^s$, and the class of functions computed by artificial neural networks using a single hidden layer, each unit evaluating a sigmoidal activation function. Our lower bounds are stated in terms of an invariant that measures the oscillations of functions of the space K around the origin. As an application we estimate the minimal number of neurons required to approximate bounded functions satisfying uniform Lipschitz conditions of order α with accuracy ϵ.

Keywords: Mathematics of Neural Networks, Approximation Theory.

1 Introduction

During the last decade a great deal of research in the field of approximation theory has been done to approximate real valued functions using artificial neural networks (ANN's) with one or more hidden layers, each neuron evaluating a sigmoidal or radial basis function (see [1,2,3,4,5,6,7,8,9]). A typical result in this context is a density result showing that an ANN can approximate a given function in a given class to any degree of accuracy provided that enough number of neurons can be used.

One of the most general results in the core of approximation by ANN's states that any continuous function can be approximated to any degree of accuracy by a single hidden layer ANN having a non-polynomial activation function provided that sufficiently many hidden units are available ([10]). This powerful statement extends the seminal theorems by Hornik, Stinchcombe and White([11]) where approximation by multilayer ANN's is proven for L^p functions under the L^p norm and for continuous functions with respect to the uniform norm. Regarding the single hidden layer case it is also known that ANN's with a hidden layer are capable of providing an optimal order of approximation for functions assumed to admit a given number of derivatives, if the activation functions evaluated by each principal element satisfy certain technical conditions (see [6]). Under these conditions it is also possible to construct networks that provide a geometric order of approximation for analytic target functions.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1–8, 2009.
© Springer-Verlag Berlin Heidelberg 2009

A related important problem is the *complexity problem*; i. e. to determine the number of neurons required to guarantee that all functions, assumed to belong to a certain function class, can be approximated within a prescribed accuracy ϵ. For this problem upper and lower bounds are of interest. An upper bound for the complexity problem is provided by the result of Barron ([1]). He proved that if the function is assumed to satisfy certain conditions in terms of its Fourier transform, and each of the neurons evaluates a sigmoidal activation function, at most $O(\epsilon^{-2})$ neurons are needed to achieve the order of approximation ϵ. A very common assumption about the function class is defined in terms of the number of derivatives that a function possesses. For example one is interested in approximating all real valued functions of s real variables having a continuous gradient. It is shown in [12] that any reasonable scheme to approximate up to order ϵ any function of this class must depend upon at least $\Omega(\epsilon^{-s})$ parameters. If one restricts to functions having r continuous derivatives then the general results by Devore, Howare and Michelli [12] indicate that the number of neurons is $\Omega(\epsilon^{-\frac{s}{r}})$.

While the complexity problem of approximation by ANN's has been studied in depth for classes of functions satisfying certain conditions on the derivatives, the problem of whether a certain class of integrable functions without assuming differentiable conditions can be optimally approximated is less known. However this last problem is of practical interest since very often target functions coming from real world problems cannot be assumed to possess differentiability properties.

Through out this paper we focus on the complexity problem for L^1 functions, i.e., given a class of non-polynomial activation functions (like for instance the sigmoidal function), determine lower bounds for the number of neurons needed to approximate compact subspaces of Lebesgue measurable functions defined on hypercubes of the kind $[0,1]^s$ to achieve order of approximation ϵ. More concretely, we provide a general method for estimating the distance between a compact subspace K of $L^1([0,1]^s)$ and the class of functions computed by ANNs using a single hidden layer, each unit evaluating a sigmoidal activation function. Our lower bounds are stated in terms of an invariant that measures the oscillations of functions of the space K around the origin. As an application we estimate the minimal number of neurons required to approximate bounded continuous functions satisfying an uniform Lipschitz condition of given order α with accuracy ϵ. Our lower bound for these functions is of the kind (using the *Big O* notation for complexity bounds)

$$\Omega\left(\left[\frac{s!}{(s+\alpha)\cdots(1+\alpha)}\epsilon\right]^{-\frac{s}{\alpha}}\right).$$

The paper is organized as follows. Section 2 describes the mathematical formalization of our approximation problem and the statement of the main results. Section 3 is devoted to explain the main technical details of our method for computing lower bounds. Section 4 and Section 5 sketch the proofs of the main result and their applications.

2 Main Results

Let s, $n \geq 1$ be integers. An artificial network $\mathcal{N}$ with a single hidden layer consisting of n neurons with sigmoidal activation function σ evaluates a function $\varPhi : \mathbb{R}^s \to \mathbb{R}$ of the form

$$\varPhi(x) = \sum_{k=1}^{n} a_k \sigma(w_k \cdot x + b_k), \tag{1}$$

where $x \in \mathbb{R}^s$, the weights w_k are vectors with s real components, the thresholds $b_k \in \mathbb{R}$, the coefficients $a_k \in \mathbb{R}$ for $1 \leq k \leq n$, $\sigma : \mathbb{R} \to \mathbb{R}$, $\sigma(y) = \frac{1}{1+e^{-y}}$, is the activation function and '$\cdot$' denotes the inner product. We shall denote by $\varPi_{\varPhi,n,s}$ the set of functions:

$$\varPi_{\varPhi,n,s} := \left\{ x \to \sum_{k=1}^{n} a_k \sigma(w_k \cdot x + b_k) : \; a_k, \; b_k \in \mathbb{R} \;\; w_k \in \mathbb{R}^s \right\} \tag{2}$$

Let $L^1([0,1]^s)$ be the normed linear space of Lebesgue measurable functions on the hypercube $[0,1]^s$. We measure the degree of approximation of $f \in L^1([0,1]^s)$ by the quantity:

$$\varepsilon_{\varPhi,n,s}(f) := \inf_{g \in \varPi_{\varPhi,n,s}} \{ ||f - g|| \}, \tag{3}$$

where $|| \cdot ||$ denotes the $L^1(\mu)$-norm in $L^1([0,1]^s)$, i.e., $||f|| = \int_{[0,1]^k} |f| d\mu$, and μ is the Lebesgue measure. Since the target function f is unknown the quantity of interest is the degree of approximation of subspaces K of L by functions of $\varPi_{\varPhi,n,s}$, that is:

$$\varepsilon_{\varPhi,n,s} := \sup_{f \in K} \varepsilon_{\varPhi,n,s}(f) \tag{4}$$

Our search for lower bounds of $\varepsilon_{\varPhi,n,s}$ leads to the following definition.

Definition 1. *For any compact subset K of $L^1([0,1]^s)$ and any non-negative integer m, the invariant $\jmath(K,m)$ is defined as the supremum of all real numbers $\delta \geq 0$, such that there are points $x_1, \ldots, x_m \in [0,1]^s$ satisfying:*

1. *The morphism $eval(f) = (f(x_1), \ldots, f(x_m))$, $f \in K$, has S-norm bounded by 1, where S-norm is defined as $||(x_1, \ldots, x_m)||_S = \sum_{1 \leq i \leq m} |x_i|$.*
2. *For every sign sequence $s = (s_1, \ldots, s_m) \in \{-1, +1\}^m$ there is $f \in K$ such that $s_i f(x_i) \geq \delta$ for $i = 1, \ldots, m$.*

Next we state our main result.

Theorem 1. *There is an integer number $p > 0$, such that for m big enough depending only on n and s and for any compact subset K of the normed linear space $L^1([0,1]^s)$ the following holds:*

$$\varepsilon_{\varPhi,n,s}(K) \geq \frac{m}{p} \jmath(K,m) \tag{5}$$

As application of Theorem 1 we have the following result.

Corollary 1. *Let $||\ ||$ denotes the Euclidean norm in $[0,1]^s$. For $\alpha \in (0,1]$ let $H_{\alpha,s}$ be the set of real valued functions f on $[0,1]^s$ bounded by 1, satisfying an uniform Lipschitz condition of order α, i.e., a function $f \in H_{\alpha,s}$ satisfies:*

$$|f(x+t) - f(x)| \leq ||t||^{\alpha} \tag{6}$$

for all x and $x+t$ in $[0,1]^s$. Then in $L^1([0,1]^s)$ the following holds:

$$\varepsilon_{\Phi,n,s}(H_{\alpha,s}) = \Omega\left(n^{-\frac{\alpha}{s}} \frac{s!}{(s+\alpha)\cdots(1+\alpha)}\right) \tag{7}$$

Remark 1. Note that from Corollary 1 we immediately obtain a lower bound for the minimal number of neurons required to approximate within accuracy ϵ functions in $H_{\alpha,s}$. This bound is of the kind:

$$\Omega\left(\left[\frac{s!}{(s+\alpha)\cdots(1+\alpha)}\epsilon\right]^{-\frac{s}{\alpha}}\right),$$

as it was claimed in the introduction.

3 Partitions of the Space of Parameters

Let $x_1, \ldots, x_m \in [0,1]^s$ be a set of m points. Let us consider the set of m functions given by:

$$\Phi_i(a,b,w) := \sum_{k=1}^{n} a_k \sigma(w_k \cdot x_i + b_k),\ 1 \leq i \leq m, \tag{8}$$

where $a := (a_1, \ldots, a_n) \in \mathbb{R}^n$, $b := (b_1, \ldots, b_n) \in \mathbb{R}^n$, $w := (w_1, \ldots, w_n)$ with $w_k \in \mathbb{R}^s$ for all k, $1 \leq k \leq n$ and σ being the sigmoid function. In what follows for each k, $1 \leq k \leq n$ and for each j, $1 \leq j \leq s$, the $j-th$ component of vector w_k will be denoted by w_{kj}. Similarly the $j-th$ component of point x_i will be denoted by x_{ij}. Our analysis of functions Φ_i relays in the properties of Pfaffian functions. We briefly recall this notion (see [13] for a complete exposition of the theory of Pfaffian functions).

Definition 2. *Let $U \subset \mathbb{R}^n$ be an open domain. A Pfaffian chain of length $q \geq 0$ and degree $D \geq 1$ in U is a sequence of real analytic functions $f_1, \ldots, f_q$ in U satisfying differential equations*

$$\frac{\partial f_i}{\partial y_j} = P_{i,j}(y, f_1(y), \ldots, f_i(y)) \tag{9}$$

for $i = 1, \ldots q$ where $P_{i,j} \in [y_1, \ldots, y_n, z_1, \ldots, z_i]$ are polynomials of degree $\leq D$.

Definition 3. *A function f on U is called a Pfaffian function of order q and degree (D, d) if*

$$f(y) = P(y, f_1(y), \ldots, f_q(y)) \tag{10}$$

Proposition 1. *Functions $\varPhi_i : \mathbb{R}^{n(s+2)} \to \mathbb{R}$ are Pfaffian of order $q = nm$ and degree $(D, d) = (2, 2)$*

Proof. For each k, $1 \leq k \leq n$, for each i, $1 \leq i \leq m$ define the functions $f_{ki} : \mathbb{R}^{n(s+2)} \to \mathbb{R}$ as follows.

$$f_{ki}(a, b, w) := \sigma(w_k.x_i + b_k), \tag{11}$$

where σ is the sigmoid function. Since the derivative of σ, σ', satisfies $\sigma' = \sigma(1-\sigma)$, functions f_{ki} satisfies the following partial differential equations for all k, i with $1 \leq k \leq n$ $1 \leq i \leq m$.

$$\frac{\partial f_{ki}}{\partial a_j} = 0, \ 1 \leq j \leq n \tag{12}$$

$$\frac{\partial f_{ki}}{\partial b_j} = f_{ki}(1 - f_{ki}), \ if \ k = j \tag{13}$$

and

$$\frac{\partial f_{ki}}{\partial b_j} = 0, \ if \ k \neq j \tag{14}$$

$$\frac{\partial f_{ki}}{\partial w_{kj}} = x_{ij} f_{ki}(1 - f_{ki}), \ if \ 1 \leq j \leq s \tag{15}$$

$$\frac{\partial f_{ki}}{\partial w_{lj}} = 0, \ if \ l \neq k \ and \ 1 \leq j \leq s \tag{16}$$

It follows from Equations 12, 13, 14, 15 and 16 that the family of nm functions $\{f_{ki}\}_{1 \leq k \leq n, 1 \leq i \leq m}$ is a Pfaffian chain of length $q = nm$ of degree bounded by $D = 2$. Let $y_1, \ldots, y_n$ be new variables. For $1 \leq i \leq n$ define de degree two polynomials:

$$P_i(a, b, w, y_1, \ldots, y_n) := \sum_{1 \leq k \leq n} a_k y_k \tag{17}$$

It is clear from the definition of functions $\varPhi_i$ that for each i, with $1 \leq i \leq m$, the following holds.

$$\varPhi_i(a, b, w) = P_i(a, b, w, f_{1i}, \ldots, f_{ni}) \tag{18}$$

Next, from Equation 18 we conclude that functions $\varPhi_i$'s are Pfaffian of length nm and degree $(D, d) = (2, 2)$ as wanted.

3.1 Counting the Number of Consistent Sign Assignments to Functions Φ_i

A non zero sign assignment to a real valued function f is one of the (in)equalities: $f > 0$, $f < 0$. A non zero sign assignment to a set of m real valued functions is consistent if all m inequalities can be satisfied simultaneously by some assignment of real numbers to the variables. We state here a bound on the number of consistent non zero sign assignments to a finite set of m functions $\{\Phi_i\}_{1\leq i\leq m}$ as defined in Equation 8.

Lemma 1. *The number of consistent non-zero sign assignments to any subset of m functions $\{\Phi_i\}_{1\leq i\leq m}$ is bounded above by the quantity:*

$$(4em)^{8n^2(s+2)^2} \tag{19}$$

Lemma 1 is now a consequence of Proposition 1 and the following technical statement. The technical details of the proof are omitted.

Theorem 2. *([14]) Suppose $k \geq r$. Let Q_i $(1 \leq i \leq r)$ be elements of the polynomial ring $\mathbf{R}[y_1, \ldots, y_k, f_1, \ldots, f_q]$ of degree at most d where the $f_i(y_1, \ldots, y_q)$ form a Pfaffian chain of length q and degree $D \geq 1$. Suppose $(\epsilon_1, \ldots, \epsilon_r)$ is a regular value of $Q = (Q_1, \ldots, Q_r) : \mathbf{R}^k \to \mathbf{R}^r$ (i.e. $Q^{-1}(\epsilon_1, \ldots, \epsilon_r)$ is a $k - r$ dimensional $\mathcal{C}^\infty$-submanifold of $\mathbf{R}^k$). Then, the number B of connected components of $Q^{-1}(\epsilon_1, \ldots, \epsilon_r)$ satisfies*

$$B \leq 2^{q(q-1)/2} d^k (k(d + D))^k (k^2(d + D))^q \tag{20}$$

Corollary 2. *Let $\Phi_1(a, b, w), \ldots, \Phi_m(a, b, w)$ as in Equation 8. There is a polynomial function $g(n, s)$ depending only on n and s and a constant $p > 0$ such that for each $m \geq g(n, s)$ there exists a sign sequence $(s_1, \ldots, s_m) \in \{-1, 1\}^m$ which differs in more than m/p places from any sequence of terms $+1, -1$ taken on by sign $\Phi(a, b, w) := ($sign $\Phi_1(a, b, w)$ $, \ldots,$ sign $\Phi_m(a, b, w))$ when $(a, b, w) \in \mathbb{R}^{(n(s+2)}$.*

Remark 2. The proof of Corollary 2 is a consequence of Lemma 1, Stirling approximation of $k!$ and some easy combinatorics.

4 Lower Bounds for $\varepsilon_{\Phi,n,s}$

This section sketches the proof of Theorem 1. Let m be an integer greater than or equal to $g(n, s)$, where $g(n, s)$ is as in Corollary 2. The case $\jmath(K, m) = 0$ is trivial. Otherwise, let δ be any positive number $< \jmath(K, m)$. Choose real numbers $x_1, \ldots, x_m \in [0, 1]^s$ satisfying the conditions of Definition 1, that is:

1. The morphism $eval(f) := (f(x_1), \ldots, f(x_m))$ has S-norm bounded by 1.
2. For every sign sequence $s = (s_1, \ldots, s_m) \in \{-1, +1\}^m$ there is $f \in K$ such that $s_i f(x_i) \geq \delta$ $i = 1, \ldots, m$.

Let Φ as in Equation 1. Consider the functions $\Phi_i(a, b, w)$ defined in Equation 8. By Corollary 2 there exists a sign sequence $s = (s_1, \ldots, s_m) \in \{0, 1\}^m$ which differs in more than m/p places from any sequence of terms $+1, -1$ taken on by $sign\ \Phi(a, b, w) := (sign\ \Phi_1(a, b, w), \ldots, sign\ \ \Phi_m(a, b, w))$. Let us take $f \in K$ satisfying condition 2 above. Note that for every $(a, b, w) \in \mathrm{I\!R}^{s(n+2)}$, there is $i \in \{1, \ldots, m\}$ such that $|f(x_i) - \Phi_i(a, b, w)| \geq \delta$. Then define $S(f) := (f(x_1), \ldots, f(x_m)$. It is straightforward to check that

$$||S(f) - (\Phi_1(a, b, w), \ldots, \Phi_m(a, b, w))||_S \geq m\delta. \tag{21}$$

Finally, we have:

$$\varepsilon_{\Phi,n,s} \geq \varepsilon_{\Phi,n,s}(f) \geq \varepsilon_{(S(\Phi),n,s)}(S(f)) \geq$$

$$\geq \inf_{(a,b,w)\in \mathrm{I\!R}^{s(n+2)}} ||S(f) - (\Phi_1(a, b, w), \ldots, \Phi_m(a, b, w))|| \geq m\delta$$

and Theorem 1 follows.

5 Approximation in the L^1 Norm

This section illustrates the application of Theorem 1. We sketch here the proof of Corollary 1. Let r be the least integer such that $r^s \geq g(n, s)$, where $g(n, s)$ is as in Corollary 2. Put $m = r^s$ and divide $I = [0, 1]^s$ into m disjoint cubes K_i each of side $1/r$. Define linear functionals $a_i(f) := \int_{K_i} f d_\mu$ where μ is the Lebesgue measure on I. Note that the mapping $a(f) := (a_1(f), \ldots, a_m(f))$ has norm one with respect to the S-norm on $\mathrm{I\!R}^m$. Given any sign sequence $(s_1, \ldots, s_m)$, $s_i \in \{+1, -1\}$, define h on I by $h(x) = s_i(distance(x, \partial K_i))^\alpha$, for x in K_i, where ∂K_i denotes the boundary of K_i and *distance* is Euclidean distance. It is easy to check that $h \in H_{\alpha,s}$. For $i = 1, \ldots, m$,

$$s_i a_i(h) = \int_{K_i} s_i(distance(x, \partial K_i))^\alpha = d_\mu \tag{22}$$

$$2^s s! \int_0^{1/2r} \int_0^{y_i} \cdots \int_0^{y_{s-1}} y_s^\alpha d_{y_s} \cdots d_{y_1} \tag{23}$$

$$m^{-1} r^{-\alpha} 2^{-\alpha} \frac{s!}{(s+\alpha)\ldots(1+\alpha)} \tag{24}$$

Thus the invariant $\jmath(K, m)$ introduced in Definition 1 is bounded below by 24. Assuming that $r \geq 2$ Equation 7 of Corollary 1 follows from Theorem 1.

Acknowledgments

This work was partially supported by Spanish Grants TIN2007-67466-C02-02, MTM2007-62799 and Spanish FPU program.

References

1. Barron, R.: Universal approximation bounds for superposition of a sigmoidal function. IEEE Trans. Information Theory 39, 930–945 (1993)
2. Barron, A., Barron, A.L.: Statistical learning networks: a unified view. In: Symp. on the Interface: Statistics and Computing Science, Reston, Virginia (1988)
3. Broomhead, D.S., Lowe, D.: Multivariable functional interpolation and adaptative networks. Complex Systems 2, 321–355 (1988)
4. Cybenko, G.: Approximation by superposition of sigmoidal functions. Mathematics of Control, signal and Systems 2, 303–314 (1989)
5. Girosi, F., Jones, M., Poggio, T.: Regularization theory and neuronal networks arquitectures. Neural Computation 7, 219–269 (1995)
6. Mhaskar, H.N.: Neural networks for optimal approximation of smooth and analytic functions. Neural Computation 8, 164–167 (1996)
7. Moody, J., Darken, C.: Fast learning in networks of locally tuned processing units. Neural Computation 1(2), 282–294 (1989)
8. Park, J., Sandberg, I.w.: Universal approximation using radial basis function networks. Neural Computation 3(2), 246–257 (1991)
9. Poggio, T., Girosi, F., Jones, M.: From regularization to radial tensor and additive splines. In: Proc. Neural Networks for signal Processing, vol. III, pp. 3–10. IEEE, New York (1993)
10. Leshno, M., Lin, V., Pinkus, A., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neuronal Networks 2, 359–366 (1989)
11. Hornink, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neuronal Networks 2, 359–366 (1989)
12. Devore, R., Howard, R., Micchelli, C.A.: Optimal non-linear approximation. Manuscripta Mathematica 63, 469–478 (1989)
13. Gabrielov, A.N., Vorobjov, N.: Complexity of computations with Pfaffian and Noetherian functions. Normal Forms, Bifurcations and Finiteness Problems in Differential Equations. Kluwer, Dordrecht (2004)
14. Khovanskii, A.: On a class of systems of transcendental equations. Soviet Math. Dokl. 22, 762–765 (1980)

A Wavelet Based Method for Detecting Multiple Encoding Rhythms in Neural Networks

Carlos Aguirre and Pedro Pascual

GNB, Escuela Politécnica Superior, Universidad Autonoma de Madrid,
28049 Madrid, Spain
{`Carlos.Aguirre`,`Pedro.Pascual`}`@uam.es`

Abstract. In this work we propose the use of the discrete wavelet transform for the detection of multiple encoding rhythms that appear, for example, in spatio-temporal patterns generated by neuronal activity in a set of coupled neurons. The method here presented allows a quantitative characterization of spatio-temporal patterns and is based on the behavior of a compression-like scheme. The wavelet-based method is faster than the two-dimensional spectral methods for finding different rhythms on spatio-temporal patterns, as it has a computational complexity $O(width \times height)$ for each 2D-frame of the spatio-temporal pattern. The method also provides a easy method for classifying different qualitative behaviors of the patterns.

1 Introduction

Spatio-temporal patterns of neuron activity are obtained in many experiments both from in-vitro registrations of several neural circuits, for example, in mammals [6] and from computer simulations of these real circuits [9]. The role of many of these circuits is still unclear, but some of them are supposed to play a role of control and coordination between the intrinsic oscillatory properties of the neurons and the nature of their connections. These networks produce spatio-temporal patterns where many rhythms can co-exist, produced both from the behavior of each individual neuron in the network and from the interactions between neurons.

The usual tools for detecting rhythms in spatio-temporal patterns are based on two dimensional spectral methods [3][2]. An important drawback of the spectral methods is the fact that the coefficients of the transformed pattern are hard to interpretate and present many artifacts (due mainly to border conditions and aliasing effects), furthermore, spectral methods are relatively slow when compared with other methods, as the two dimensional Fourier transform in a square of side length N requires $O(N^2 logN)$ floating point operations for the transform of a single frame of the pattern, even using the Fast Fourier Transform (FFT).

Wavelet based techniques has proven to be a useful tool for signal analysis and [7], in particular, for the study of images or sequences of images [8]. For the study of spatio-temporal patterns (composed of a sequence of images in time)

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 9–16, 2009.
© Springer-Verlag Berlin Heidelberg 2009

we propose a wavelet based compression scheme. Wavelet based compression schemes has been used, for example, for one dimensional signal segmentation through 'wavelet probing' techniques [5]. The idea under the wavelet based compression scheme is that wavelet coefficients that correspond with parts of the image that are smooth have a small value. Complex images present a low number of small parameters of the two dimensional wavelet transform (2DWT) and their compression ratios are lower. For each image in the pattern the number of coefficients of the 2DWT that are higher than a given threshold are computed. This produces a one dimensional signal representing the evolution in the 'complexity' of each image in the spatio-temporal pattern. Simple observation of this new signal allows to see more clearly the presence of different rhythmic patterns generated by network activity. Furthermore, these rhythms can now be easily calculated by direct use of the one dimensional FFT.

The Wavelet transform is related with multiresolution analysis and presents a hierarchical structure that is particularly suited for fast numerical algorithms [4]. The 2DWT algorithm has a computational complexity as low as $O(WHt)$ where W and H are respectively the image width and height in pixels of each image and t is the number of images. A similar compression scheme based on spectral techniques has a computational complexity for the same pattern that is $O(WHmax\{log(W) * log(H)\}t)$.

We use here the above wavelet based scheme to study the behavior of a simulated network of neurons, with different degree of coupling, and show how the discrete wavelet transform may be used to analyze the network activity and give also a quantitative characterization of the spatio-temporal patterns that arise for different degrees of coupling between the neurons that compose the network. This characterization can be utterly used, for example, for pattern discrimination or classification.

2 Wavelet Based Scheme for Rhythms Detection

In general, any function $f \in L(R^n)$ (the square summable functions in R^n) can be expressed in the following way

$$f = \sum_k \lambda_{\nu_0,k}\psi_{\nu_0,k} + \sum_{\nu \geq \nu_0,k} \mu_{\nu,k}\psi_{\nu,k} \tag{1}$$

De wavelet transform of f $(WT(f))$ is the set of coefficients

$$WT(f) = \{\lambda_{\nu_0,k}\} \cup \{\mu_{\nu,k}\} \quad with \quad k = ... - 1, 0, , 1... \quad and \;\; \nu \geq \nu_0 \tag{2}$$

where ν_0 usually takes the value 0.

The set of functions $\psi_{\nu_0,k}$ and $\psi_{\nu,k}$ compose the wavelet basis. Each function of the basis is obtained by dilation and translation of a single function ψ named *the mother wavelet*. If the mother wavelet and the set of dilations/translations is selected adequately [4], the set of functions obtained form an orthonormal basis of R^n.

By the orthonormality of the wavelets basis, each coefficient can be directly calculated by direct integration, this is.

$$\lambda_{\nu_0,k} = \int_{R^n} f(\mathbf{x})\psi_{\nu_0,k}(\mathbf{x})d\mathbf{x} \tag{3}$$

$$\mu_{\nu,k} = \int_{R^n} f(\mathbf{x})\psi_{\nu,k}(\mathbf{x})d\mathbf{x} \tag{4}$$

For some families of wavelets, a multiresolution process can be performed [4], this process allows to compute the coefficients of the wavelet transform by means of the discrete wavelet transform with only $O(N)$ operations, where N is the number of samples in f.

For most wavelet basis, the coefficients of the Wavelet transform represent resolution content of the data. In one dimensional signals, spectral methods are suitable for the detection of the possible rhythms present in the signal, however in higher dimensions the coefficients produced by the multidimensional Fourier transform are hard to interpretate and they present a high number or artifacts not directly related with the behavior of the signal but with sampling (aliasing effects) or border conditions. On the other hand two dimensional wavelets basis are easy to generate by direct Cartesian product of one dimensional wavelet transform. The two dimensional wavelet transform have been used for image compression, as they present both high compression levels and a low computational effort.

In difference with the Fourier transform coefficients, where the 'frequency' content of the signal can not be localized in time (or space), the wavelet transform coefficients $\{\lambda_{\nu_0,k}\} \cup \{\mu_{\nu,k}\}$ are determined both by a resolution component (ν) and a time (or space) component (k), this is, the coefficients of the wavelet represents the resolution content at a given portion of the original signal. In general, wavelet compression techniques are based on the fact that the wavelet coefficients that correspond to smooth portions of the data become small on the finer levels of resolution and therefore, smooth portions can be represented only with a few low resolution components. To represent correctly complex components we need both high resolution (details) and low resolution components.

The number of coefficients of the transform that are higher than a given threshold, or alternatively, the number of coefficients that comprise a given percentage of the total energy of the signal, gives an insight of the whole complexity of the signal. When we have a high number of coefficients higher that the fixed threshold means that the signal content in the details is high and low compression can be performed. When the number of coefficients higher than the threshold is low means that the image is smooth or is composed of smooth components and high compression can be achieved as coefficients with small values (mainly on the finer levels) can be set to zero without highly distorting the original data.

The previous idea allows the study in the evolution of a spatio-temporal pattern composed of a succession of images. The method consists in estimating the compression rate of each image in a spatio-temporal pattern by calculating the number of coefficients higher than a given threshold for that image. This method produces a new one dimensional signal $C(t)$ that at time t represent the 'complexity' of the corresponding image of the spatio-temporal pattern at time t. This one dimensional signal can be now studied with standard statistical (self-correlation, mean value, etc) or spectral (FFT) methods.

The method can be resumed as follows:

```
C WaveletAnalysis(pattern, w,h, n,basis,th)

Input
  // pattern  spatio-temporal pattern
  // w        width of each frame in pixels
  // h        height of each frame in pixels
  // n        number of images
  // basis    wavelet basis for the transform
  // th       threshold for wavelet coefficients
Output
  // C        estimation of compression level

  for i = 1 to n
    image=pattern(i) // extract image at time i
    imagetran = WaveletTransform2D(image,w,h,basis)
    C[i]=0
    for j= 1 to width
      for k= 1 to height
        if imagetran[j][k]>th
          C[i]=C[i]+1

  return C
```

2.1 Wavelet Analysis of Network Activity

Spatio-temporal patterns of neuron activity are of special interest nowadays. They have been observed in some in-vitro experiments in networks like the Inferior Olive (IO) [6], though the inner working of sub-cellular and network processes that give rise to this phenomena is not yet well understood.

Some numerical simulations have been carried out trying to reproduce the observed patterns, so that they can be used to study the relevant aspects of these patterns and the neuron characteristic behavior that may originate them. In [9], an HH differential equation model is used to reproduce these patterns. In this work, it is stated the importance of sub-threshold oscillations and spiking in the generation of spatio-temporal patterns, and they also show how networks of this kind may encode different coexisting rhythms. We have used the simple neuron model presented at [1] to reproduce these patterns, this model allows subthreshold oscillations, bursting and neuron coupling. We have built a two dimensional network of 50×50 identical neurons, whose behavior is governed by this model.

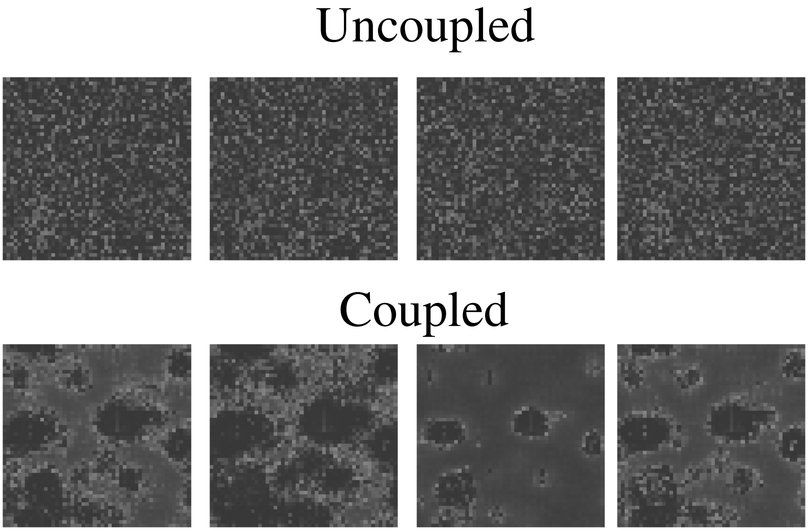


Fig. 1. Spatio-temporal patterns

Each neuron is connected to its 8 closest neighbors in a rectangular grid, with periodic boundary conditions. Low row of figure 1 shows the kind of patterns that arise in such a network after the injection of a uniform external current on a small area of the network; these patterns are very similar to those obtained in-vivo [6] or with more realistic, and complex, models [9]. In the upper row of figure 1 we show the result of uncoupling the neurons as happens, for example, when a synapses inhibitor is injected, so that they behave independently, no patterns are observed, and every neuron fires at its own.

Following the previous scheme we have used the method presented in section 2 to characterize quantitatively the spatio-temporal patterns of the network. This measure gives us a better insight than a mere visual inspection of the movies of the simulations (available under request from the authors).

A two-dimensional basis was generated by direct Cartesian product of the one-dimensional Haar basis [4]. The generation of a two dimensional basis by Cartesian product is equivalent to perform a one dimensional wavelet transform for each row of the image followed by a one dimensional transform for each column of the image [8]. Figure 2 shows the evolution of the number of coefficients for two different networks: a network with electrical coupling $g_{ij} = 0.0075$ and a network without coupling $g_{ij} = 0$. The average higher value or $C(n)$ in the latter shows a more complex spatial structure, where every neuron behaves independently, and no patterns are present. The spiky waveform in the evolution of $C(n)$ in the case of coupled neurons shown in fig. 2 also indicates the presence of multiple spiking frequencies evoked by the stimuli in the network, and the periodic evolution of spatial patterns.

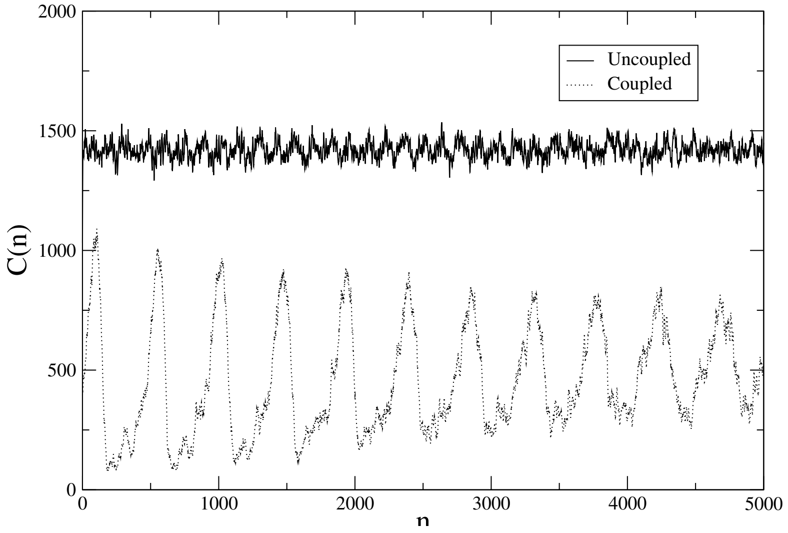


Fig. 2. Wavelet analysis of spatio-temporal patterns

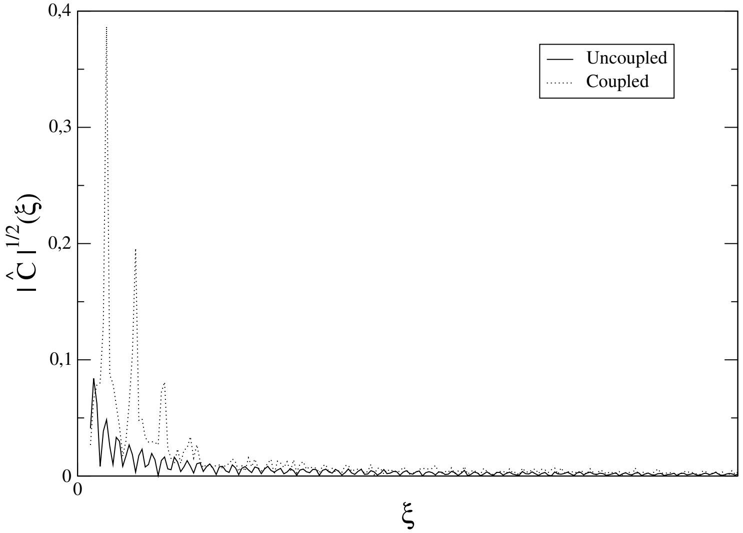


Fig. 3. Power spectrum of the wavelet analysis of spatio-temporal patterns

The encoding frequencies present in the network can be calculated by means of the Fourier transform of the wavelet analysis of the network activity. In figure 3 the Fourier transform of the wavelet analysis for both networks is presented.

3 Discussion

The coupled network present a clear set of dominant frequencies in the network activity that correspond with the different frequencies of the spatio-temporal patterns present in the network as it is shown in figure 3, on the other hand, the uncoupled network present a much less spiky distribution of the frequencies spectrum due to the absence of spatio-temporal patterns and reflecting exclusively the periodic behavior of the individual neurons in the network.

4 Conclusions

The discrete wavelet transform may be a useful tool when studying encoding rhythms that appear in spatio-temporal patterns generated both by in-vitro recordings and computer simulations of several neural networks. The method here presented allowed the detection and quantification of this encoding rhythms and could be useful for the better understanding of the functional role of such neural networks.

The low computational resources required by the method also could help as real time tool for discrimination or classification on the behavior of neural networks under different conditions of coupling or input.

Acknowledgments

(CA) is supported by MEC BFU2006-07902/BFI, CAM S-SEM-0255-2006 and CAM/UAM CCG08-UAM/TIC4428. (PP) is also partially supported by MEC (Spain), project TIN2007-66862-C02-01. We thank the discussions with Prof. Luis Vzquez in the framework of the project AYA2008-06420-C04-03/ESP supported by Ministerio de Ciencia e Innovacion of Spain.

References

1. Aguirre, C., Campos, D., Pascual, P., Serrano, E.: Neuronal behavior with subthreshold oscillations and spiking/bursting activity using a piecewise linear two-dimensional map. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 103–108. Springer, Heidelberg (2005)
2. Bai, O., Mari, Z., Vorbach, S., Hallett, M.: Asymmetric spatiotemporal patterns of event-related desynchronization preceding voluntary sequential finger movements: a high-resolution EEG study. Clinical Neurophysiology 116(5), 1213–1221 (2005)
3. Daffertshofer, A., Peper, C.E., Frank, T.D., Beek, P.J.: Spatio-temporal patterns of encephalographic signals during polyrhythmic tapping. Human Movement Science 19, 475–498 (2000)
4. Daubechies, I.: Ten Lectures on Wavelets. In: CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 61 (1992)
5. Deng, B., Jawerth, B., Peters, G., Sweldens, W.: Wavelet Probing for Compression Based Segmentation SPIE Conference, San Diego (1993)

6. Leznik, E., Contreras, D., Makarenko, V., Llinas, R.: Markov field analysis of inferior olivary oscillation determined with voltage-dependent dye imaging in vitro. Society for Neuroscience 25, 1252 (1999)
7. Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press, London (1999)
8. Stollnitz, E.J., Derose, T.D., Salesin, D.H.: Wavelets for Computer Graphics. Morgan Kaufman Publishers, Los Altos (1996)
9. Varona, P., Aguirre, C., Torres, J.J., Rabinovich, M.I., Abarbanel, H.D.I.: Spatiotemporal patterns of network activity in the inferior olive. Neurocomputing 44–46, 685–690 (2002)

Switching Dynamics of Neural Systems in the Presence of Multiplicative Colored Noise

Jorge F. Mejias[1], Joaquin J. Torres[1], Samuel Johnson[1],
and Hilbert J. Kappen[2]

[1] Department of Electromagnetism and Matter Physics,
University of Granada, 18071 Granada, Spain
`jmejias@onsager.ugr.es`
`http://ergodic.ugr.es/neurofisica/neurofisica.html`
[2] Department of Biophysics, Radboud University of Nijmegen
6525 EZ Nijmegen, The Netherlands

Abstract. We study the dynamics of a simple bistable system driven by multiplicative correlated noise. Such system mimics the dynamics of classical attractor neural networks with an additional source of noise associated, for instance, with the stochasticity of synaptic transmission. We found that the multiplicative noise, which performs as a fluctuating barrier separating the stable solutions, strongly influences the behaviour of the system, giving rise to complex time series and scale-free distributions for the escape times of the system. This finding may be of interest to understand nonlinear phenomena observed in real neural systems and to design bio-inspired artificial neural networks with convenient complex characteristics.

Keywords: Bistable systems, switching dynamics, multiplicative colored noise, neural up and down states.

1 Introduction

Attractor neural networks (ANN) have been a topic of high interest in the last decades. Most of the paradigmatical approaches in this field consider a number of bio-inspired elements from biological neural systems and study the computational properties of the resulting model, leading to hints and developments in neuroscience and computer algorithm design. One of the most notable models of ANN is the one proposed by Amari and Hopfield [1,2]. This model assumes a network of N binary neurons linked by connections of certain strength, also called synapses. By considering a simple learning rule based on neurophysiological data [3], this network is able to store a certain number P of patterns of activity. After this learning, the network is able to recover one of these activity patterns from an initial configuration correlated with this pattern, a property which is called *associative memory*. While the behaviour of such models is highly nonlinear, one can derive analytical solutions [4] which help to reach a better understanding for simple and hypothetical situations. For instance, if one assumes that the number of stored patterns does not scale with the size of the network, the previous model

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 17–23, 2009.
© Springer-Verlag Berlin Heidelberg 2009

is simplified to a bistable system. This is a common approach employed when one is interested in the dynamics of the network activity instead on its storing capacities [5]. It has been employed, for instance, to study the influence of the network topology [6], or the switching between different patterns of activity due to short-term synaptic mechanisms [7,8].

In this work, we employ a simplified bistable model, which mimics the dynamics of attractor neural networks in the limit of $P \ll N$, to study the effect of multiplicative colored noise in the dynamics of the system. Such noise resembles the stochastic nature of synaptic transmission [9] (which may be relevant to transmit information through dynamic synapses [10,11]), or other sources of colored noise which could affect the dynamics in a multiplicative way. We found that this multiplicative colored noise strongly affects the dynamics of the system, giving rise to complex time series and scale-free distributions for the escape times of the dynamics. Our results could be of interest to understand nonlinear phenomena observed in real neural systems and to design new paradigms in bio-inspired artificial neural networks.

2 Model and Results

We consider a bistable system characterized by the variable $y(t)$, which represents the level of activity of the neural network. This variable evolves according to the following discrete dynamics

$$y(t+1) = \tanh[z(t)y(t)] + \xi(t), \tag{1}$$

Here, the variable $z(t)$ is a Gaussian colored noise with mean one, standard deviation σ and correlation time τ. It represents a source of correlated noise associated with the stochasticity of the synaptic transmission, for instance. The term $\xi(t)$ is an additive Gaussian white noise of zero mean and standard deviation δ. This term takes into account other possible sources of non-multiplicative noise, and is also employed to prevent the system to be locked in the solution $y = 0$, since the fluctuations due to the multiplicative noise cannot influence the system in this point. In the following, we focus on the role of the correlation time in the dynamics, and therefore we fix the level of the fluctuations $\sigma = 50$ and $\delta = 0.1$. Similar models of bistable systems driven by multiplicative noise have been previously studied [12,13,14]. However, a detailed study of the statistics of the dynamics, which is highly relevant to understand certain collective behaviour in neural systems, is still lacking. A more complete study of the influence of these parameters will be published elsewhere [15].

From a mathematical point of view, we can see in Eq. (1) that the variable $z(t)$ represents the barrier height of our bistable system. For fixed $z < 1$, our system can be viewed as a particle in a single well potential in the presence of fluctuations (given by $\xi(t)$). Thus, $y(t)$ will be fluctuating around the only stable solution of the dynamics, $y = 0$. On the other hand, for fixed $z > 1$ we have a particle in a double well potential in the presence of fluctuations (given by $\xi(t)$). In this situation, the particle will be jumping, or switching, from one stable

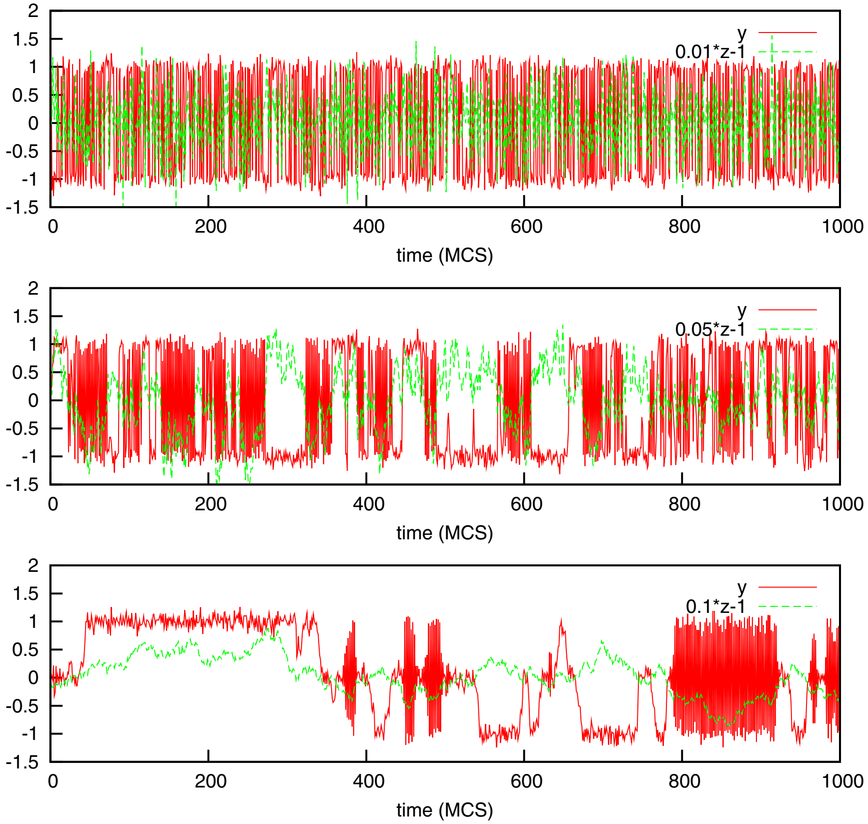


Fig. 1. (Color online) Several realizations of the time evolution of $y(t)$ and $z(t)$, for different values of the correlation time τ. The series correspond, from top to bottom, to $\tau = 0$, $\tau = 10$ and $\tau = 100$, respectively. The mean and variance of $z(t)$ have been conveniently rescaled in each case for a better visualization (see labels in each plot).

solution to the other in a stochastic manner. The fact that z has not a fixed value but it is also a fluctuating magnitude introduces a high level of complexity which is the aim of this study. For instance, we can control the characteristics of the barrier height by varying the correlation time τ. This variation has a dramatic effect in the dynamics of $y(t)$, as one can see in figure 1. The three plots in the figure shows the relation between the dynamics of $y(t)$ and $z(t)$: for $z(t) < 1$, the variable $y(t)$ rapidly fluctuates around the solution $y = 0$, and for $z(t) > 1$ we enter in the double well regime and $y(t)$ starts to swith between the symmetric stable solutions $y_+ \simeq +1$ and $y_- \simeq -1$. The switching dynamics has a strong dependence with τ, as the figure also shows. For low values of τ the switching is random and has a high frequency, but when τ is increased, some intervals of prolonged permanence in a particular solution appear. Concretely, high values of the correlation time induces drastic modifications in the probability of the

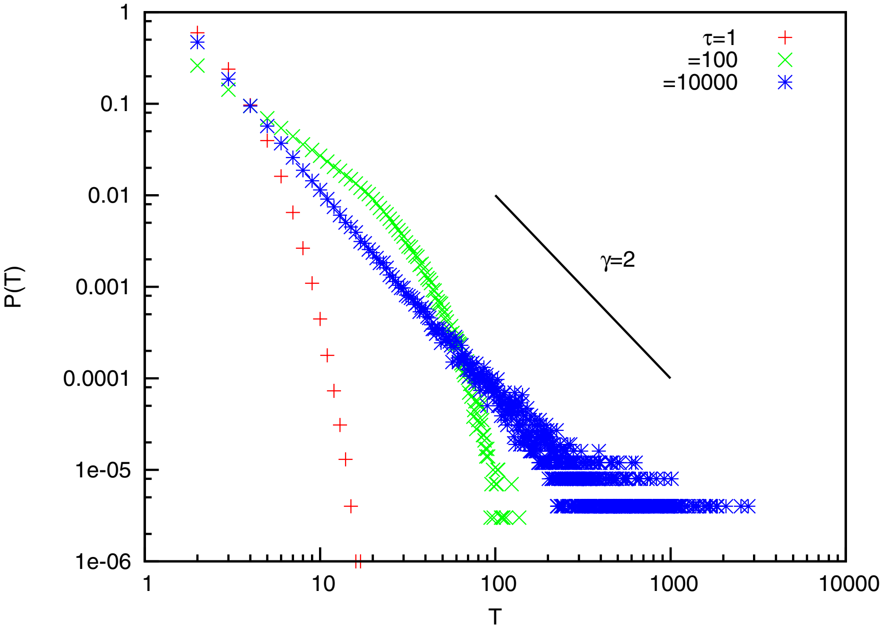


Fig. 2. Probability distribution of escape times for different values of τ. We can see that increasing of the correlation time yields the appearance of power-law distributions in escape times, which reflects the change in the statistics of the dynamics of $y(t)$. The obtained power-law distribution is defined as $P(T) \sim T^{-\gamma}$, with the exponent taking a value of $\gamma \simeq 2$.

system to jump between stable solutions after a given interval T, that is, in the probability distribution of escape times.

A more detailed study of the influence of τ in the escape time probability is shown in figure 2. One can observe that the probability distribution of the escape times is an exponential distribution for low values of τ, as it is well known. In this situation, the variable $z(t)$ behaves approximately as a white noise, and therefore $z(t)$ is continuously crossing the value $z = 1$. As a consequence of that, the double well configuration is only maintained by a very short time, and long periods of permanence in the solutions y_+ or y_- are unlikely to occur. However, when the value of τ is increased, the excursions of the variable $z(t)$ in the region of $z > 1$ become longer in time and $z(t)$ can eventually take values which are far from $z = 1$. These two factors combined allow the system to eventually stay in one of the stable solutions y_+, y_- for long periods of time, as it is shown in bottom panel of figure 1. These eventually long intervals of permanence in a double well solution are reflected in the probability distribution of escape times as a scale-free relationship, which mathematically corresponds to a power-law function $P(T) \sim T^{-\gamma}$. We found an approximate value of $\gamma \simeq 2$ for the exponent.

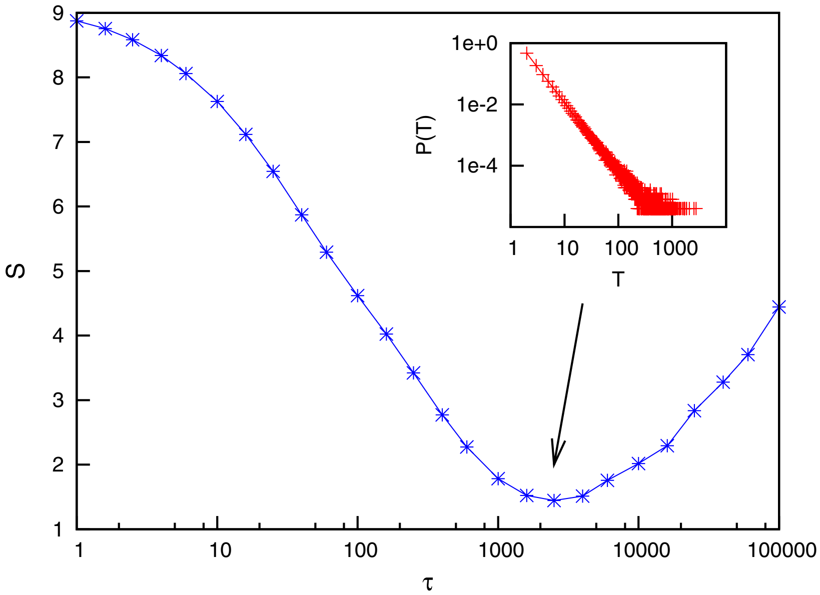


Fig. 3. The entropy function, as defined in the main text, for different values of the correlation time τ. We can observe a minimum in the entropy which approximately corresponds to the value of τ for which scale-free escape times distribution appears.

Finally, to complete the study of the dynamics of the system, we have computed a measure of the irregularity of the time series of $y(t)$, for different values of the correlation time τ. This measure is defined as an entropy of the form

$$S \equiv -\sum_x p(x)\ \log_2\ p(x) \tag{2}$$

Here, $p(x)$ is the normalized power spectrum of the time series of $y(t)$ for a given value of τ. It is worthy to note that this entropy depends strongly on the dynamics of the system (via the power spectrum), and therefore it is influenced by the value of τ. In figure 3, we can see that the entropy reaches a minimum around the value of τ for which the scale-free distributions for the escape times appear. This is due to the fact that, in this situation, long permanence times in stable double well solutions are allowed eventually, and such permanence decreases significatively the complexity of the time series of the system. Thus, the influence of τ in the dynamics of $y(t)$ is also notorious when studying the complexity of the dynamics with multiple methods.

3 Conclusions

We have studied the dynamics of a simple bistable system under the influence of multiplicative colored noise. The results show a high impact of considering high values of the correlation time in the dynamics of the system, and some nonlinear characteristics such as scale-free distributions and minima of entropy are found. One can think that the consequences of such complex behaviour in real neural systems, such as populations of cortical neurons connected by highly fluctuating synapses, could be relevant for different neural protocols of processing and coding of information. It has been experimentally found, for instance, that a multiplicative modulation of the noise intensity (that is, modulations in the variance of the noisy input) can be tracked much faster than additive modulations of noise (that is, modulations in the mean input) by neural populations [16]. Such fast tracking properties associated with multiplicative noises could be related with the high stimulus sensitivity present in critical systems as the one we have analyzed. On the other hand, several studies indicate the possible existence of criticality in other situations in the brain (see, for instance, [17]). Further study of these characteristics could also reveal strategies to implement the benefits obtained with this type of dynamics in bio-inspired computer algorithms.

Acknowledgements. This work was supported by the Junta de Andalucía project FQM-01505 and by the MEC-FEDER project FIS2005-00791. We thank S. de Franciscis for useful discussions.

References

1. Amari, S.: Characteristics of random nets of analog neuron-like elements. IEEE Trans. Syst. Man. Cybern. 2, 643–657 (1972)
2. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79, 2554–2558 (1982)
3. Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory. Wiley, Chichester (1949)
4. Amit, D.J., Gutfreund, H., Sompolinsky, H.: Statistical mechanics of neural networks near saturation. Ann. Phys. 173, 30–67 (1987)
5. Mejias, J.F., Torres, J.J.: Maximum memory capacity on neural networks with short-term depression and facilitation. Neural Comp. 21(3), 851–871 (2009)
6. Johnson, S., Marro, J., Torres, J.J.: Functional optimization in complex excitable networks. Europhys. Lett. 83, 46006 (1–6) (2008)
7. Pantic, L., Torres, J.J., Kappen, H.J., Gielen, S.C.A.M.: Associative memmory with dynamic synapses. Neural Comput. 14, 2903–2923 (2002)
8. Torres, J.J., Cortes, J.M., Marro, J., Kappen, H.J.: Competition between synaptic depression and facilitation in attractor neural networks. Neural Comput. 19, 2739–2755 (2007)
9. Allen, C., Stevens, C.F.: An evaluation of causes for unreliability of synaptic transmission. Proc. Natl. Acad. Sci. USA 91, 10380–10383 (1994)
10. de la Rocha, J., Parga, N.: Short-term synaptic depression causes a non-monotonic response to correlated stimuli. J. Neurosci. 25(37), 8416–8431 (2005)
11. Mejias, J.F., Torres, J.J.: The role of synaptic facilitation in spike coincidence detection. J. Comp. Neurosci. 24(2), 222–234 (2008)
12. Madureira, A.J.R., Hanggi, P., Buonomano, V., Rodrigues Jr., W.A.: Escape from a fluctuating double well. Phys. Rev. E 51, 3849–3861 (1995)

13. Ya, J., Jia-rong, L., Yi-cheng, C.: A novel phase transition phenomenon in bistable system. Chin. Phys. Lett. 14, 245–247 (1997)
14. Can-Jun, W., Shi-Bo, C., Dong-Cheng, M.: Steady-state analysis of a bistable system subject to a coloured multiplicative noise and a white additive noise with coloured cross-correlated noises. Chin. Phys. 15, 1435–1440 (2006)
15. Torres, J.J., Mejias, J.F., Kappen, H.J.: Bistable neural dynamics driven by multiplicative colored noise (submitted)
16. Boucsein, C., Tetzlaff, T., Meier, R., Aertsen, A., Naundorf, B.: Dynamical response properties of neocortical neuron ensembles: multiplicative versus additive noise. J. Neurosci. 29, 1006–1010 (2009)
17. Chialvo, D.R.: Psychophysics: are our senses critical? Nat. Phys. 2, 301–302 (2006)

Gradient Like Behavior and High Gain Design of KWTA Neural Networks

Daniela Danciu and Vladimir Răsvan

University of Craiova, Department of Automatic Control
13, A.I. Cuza str., 200585 - Craiova, Romania
{Daniela.Danciu,Vladimir.Rasvan}@automation.ucv.ro

Abstract. It is considered the static and dynamic analysis of an analog electrical circuit having the structure of the Hopfield neural network, the KWTA (K-Winners-Take-All) network. The mathematics of circuit design and operation is discussed *via* two basic tools: the Liapunov function ensuring the gradient like behavior and the rational choice of the weights that stands for network training to ensure order-preserving trajectories. Dynamics and behavior at equilibria are considered in their natural interaction, and some connections to the ideas in general dynamical systems of convolution type are suggested.[1]

1 Introduction and Basic Problems

A. The KWTA (K-Winners-Take-All) artificial neural networks are a special type of analog electrical circuit that selects K largest elements from a given set of N real numbers. Such kind of neural networks (NN) may be subsystems in larger systems of decision taking, pattern recognition, sorting systems, digital processing [1]. As special applications of KWTA networks we may cite telecommunications - namely control of data packet switches or analog decoding of the error correcting codes [2,9]. It is mentioned there that *KWTA networks for telecommunication applications have been realized and implemented as analog VLSI circuits*; usual difficulties of analog circuitry such as circuit noise that "may kill any useful property" are here overcome by the stability, convergence and robustness properties induced by the high gain approach.

The interconnecting structure of KWTA networks is similarly to analog Hopfield NN where the "training" of the network is ensured by choosing the network structure and the weights "once for ever" and by taking advantage of some high gain introduced in the network.

B. Following [2] we explain the basic principle of such a network on a simple example. Consider 3 real numbers d_1, d_2, d_3 such that $d_2 > d_3 > d_1$. The technical problem is that of designing a circuit with 3 inputs (where the d_i-s are applied) and 3 outputs which at equilibrium have to be *order-preserving* i.e. such that $\bar{v}_2 > \bar{v}_3 > 0 > \bar{v}_1$; this inequality chain signifies that d_2, d_3 are the largest of the sequence and $\bar{v}_2$, $\bar{v}_3$ thus "won" the competition (see Fig. 1 reproduced after [2]). This is a "2 of 3" WTA network. In the general case, there are given N numbers $d_1, \ldots, d_N$ and the network has

---

[1] This work has been realized within the framework of the Research Project CNCSIS ID-95.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 24–32, 2009.
© Springer-Verlag Berlin Heidelberg 2009

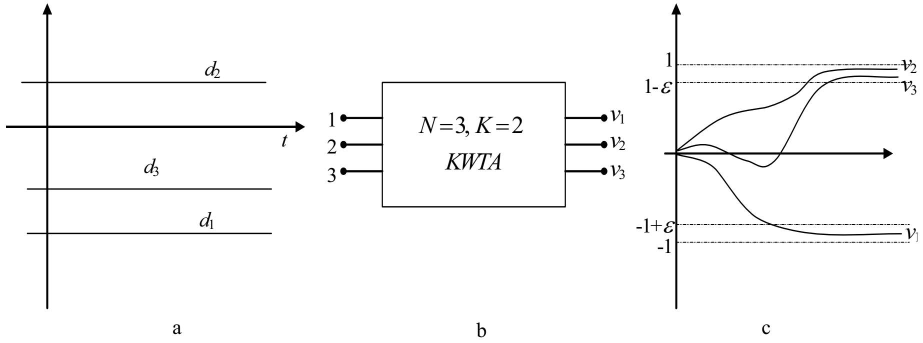


Fig. 1. 2WTA network principle: a. the input; b. the system; c. the output - reproduced after [2]

to select the highest K among them, $1 \leq K \leq N-1$. As in Fig.1, a σ-permutation of the number sequence is applied to the input terminals, the sequence being σ-ordered.

$$d_{\sigma(1)} > d_{\sigma(2)} > \ldots > d_{\sigma(N)} \tag{1}$$

The network having such a sequence at its inputs $1, \ldots, N$ must provide a N-dimensional output vector $\bar{v}$ such that

$$\bar{v}_{\sigma(i)} > 0 \ , \ i = 1, \ldots, K \ ; \ \bar{v}_{\sigma(i)} < 0 \ , \ i = K+1, \ldots, N \tag{2}$$

In fact it is required more, as shows Fig.1c : for $\bar{v}_{\sigma(i)} > 0$ it follows that $\bar{v}_{\sigma(i)} \approx 1 - \varepsilon$ and for $\bar{v}_{\sigma(i)} < 0$ it follows that $\bar{v}_{\sigma(i)} \approx -1 + \varepsilon$. The equilibria are thus "forced" in the neighborhood of some binary "point" of the output space with the coordinates 1 and -1: for instance, the case (2) is associated to the binary sequence $\{\underbrace{1, \ldots, 1}_{K}, \underbrace{-1, \ldots, -1}_{N-K}\}$; in the example above the sequence is $\{1, 1, -1\}$ - see Fig.1c. In this way the steady state output variables $\bar{v}_i$ are well delimited from 0 and the method achieves some robustness with respect to the parameter uncertainty or to the circuit noise previously mentioned.

A significant feature of this structure is that *only a single equilibrium is associated to the correct accomplishment of the task* - the equilibrium located in some ε-neighborhood of the binary point corresponding to the ordered sequence of 1 and -1 associated to the input sequence $\{d_{\sigma(i)}\}_i$ unlike standard situations when the "power" of a neural network is given by its emergent properties induced by a large number of equilibria.

C. From the dynamical point of view, the emergent computational properties of the NN are obtained not only by the existence of a (quite) large number of equilibria but also by the fact that these equilibria (or at least a subset of them) can be effectively attained i.e. the system has some global asymptotic properties; the best case would be that of the gradient like property when each transient solution is bounded and approaches asymptotically some equilibrium.

We have to point out here a contradiction that has been analyzed in our previous work [3,4,5,6]: the functionality of the neural network is based on its steady state - more precisely equilibrium - properties, but this functionality may be effectively implemented provided the dynamic properties allow it. Usually the network is designed and/or trained to achieve the steady state property while the dynamical properties as Liapunov stability of local equilibria and gradient like behavior are checked only a posteriori.

In the present paper, for the KWTA application we shall take a slightly different approach, namely to ensure the dynamical properties from the beginning. This will result in a rather simplified "homogeneous" structure where the steady state properties will be secured also from the dynamics point of view. Here we may remark some similarity not only to e.g. [2,9] but also to some earlier reference (cited there) such as [8]; the difference is given by the use of the high gain and of the Liapunov approach rather than of the first approximation which would send to consideration of a single equilibrium rather than the entire stationary set. Consequently, the remaining of the paper is organized as follows: 1) the dynamical properties of the Hopfield like networks are considered; 2) conditions for order-preserving of the dynamical trajectories are analyzed and the structure of the weights is obtained in a rational way; 3) the steady state properties are analyzed by pointing out the role of the high gain and of the low gain (for network reset) - all this analysis is performed by taking into account all the time the stability conditions; 4) some conclusions are drawn, including the possibilities of replacing the high/low gain by an adaptive gain.

2 The Network Structure and Some Dynamical Properties

A. We shall start with the well known differential equations of the analog nonlinear RC circuit having a Hopfield NN structure

$$R_iC_i\frac{\mathrm{d}u_i}{\mathrm{d}t} = -u_i - R_i\sum_1^N w_{ij}g_j(\lambda u_j) + R_iI_i \ , \ i=1,\dots,N \tag{3}$$

with N "neurons": the state variables u_i are voltages and the input variables I_i are currents; R_i are resistances, C_i are capacities hence R_iC_i are time constants; the weights w_{ij} and the gain $\lambda > 0$ are real numbers; obviously the output variables $v_i = g_i(\lambda u_i)$ are currents. The functions $g_i : \mathbb{R} \mapsto [-1,1]$ are sigmoid functions having the following properties: i) strictly monotonic and continuous; ii) $g_i(0) = 0$, $\lim_{\sigma\to\pm\infty} g_i(\sigma) = \pm 1$; iii) Lipschitz, hence differentiable a.e. with integrable derivative. We may denote $\nu_{ij} := R_iw_{ij}$, $b_i := R_iI_i$, $T_i := R_iC_i$ to obtain

$$T_i\frac{\mathrm{d}u_i}{\mathrm{d}t} = -u_i - \sum_1^N \nu_{ij}g_j(\lambda u_j) + b_i \ , \ i=1,\dots,N \tag{4}$$

a system of ODE (ordinary differential equations) with absolutely continuous and Lipschitz RHS (right hand side). For this system all results of the basic theory are true:

Theorem 1. *For system (4) having each $\lambda > 0$ and any initial condition $u(0) = u_0$ there exists a unique solution (state trajectory) which is locally defined on some interval $t_0 < 0 < t_1$ and continuously differentiable on that interval; also this solution is at least absolutely continuous with respect to $\lambda > 0$, uniformly on the definition interval.*

B. Another standard result concerns qualitative behavior of the solutions for large values of $t > 0$ including asymptotic behavior for $t \to \infty$. The result is based on the associated "natural" Liapunov function which is of the Cohen-Grossberg type [11]

$$V(u_1,\ldots,u_N) = \frac{1}{2}\sum_1^N\sum_1^N \nu_{ij} g_i(\lambda u_i) g_j(\lambda u_j) + \sum_1^N \int_0^{\lambda u_i} (\theta - b_i\lambda) g_i'(\theta)\mathrm{d}\theta \qquad (5)$$

With the *standard symmetry assumption for the weights* ($\nu_{ij} = \nu_{ji}$) system (4) appears to be a gradient system

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\frac{1}{\lambda T_i g_i'(\lambda u_i)}\frac{\partial V}{\partial u_i}\ ,\ i = 1,\ldots,N \qquad (6)$$

(since $g'(\cdot)$ are strictly monotonic we have $g'(\cdot) \neq 0$). The derivative function of the Liapunov function will thus be

$$W(u_1,\ldots,u_N) = -\sum_1^N \frac{1}{\lambda T_i g_i'(\lambda u_i)}\left(\frac{\partial V}{\partial u_i}\right)^2 \qquad (7)$$

and it vanishes on the equilibria set $\mathcal{E}$ only. We need here to recall some definitions [11]

Definition 1. 1° *Any constant solution of (4) is called* equilibrium. *The set of equilibria $\mathcal{E}$ is called* stationary set. *A solution of (4) is called* convergent *if it approaches asymptotically some equilibrium:* $\lim_{t\to\infty} u(t) = \bar{u} \in \mathcal{E}$.

A solution is called quasi-convergent *if it approaches asymptotically the stationary set:* $\lim_{t\to\infty} d(x(t),\mathcal{E}) = 0$.

2° *System (4) is called* monostable *if every bounded solution is convergent; it is called* quasi-monostable *if every bounded solution is quasi-convergent.*

3° *System (4) is called* gradient-like *if every solution is convergent; it is called* quasi-gradient-like *if every solution is quasi-convergent.*

Now we may apply the following Liapunov like results obtained by using the Barbashin-Krasovskii-La Salle invariance results

Lemma 1. 1° *Consider the nonlinear system $\dot{x} = f(x)$ and assume existence of a continuous function $V : \mathbb{R}^n \to \mathbb{R}$ that is non increasing along any solution of the system. If, additionally, a bounded on $\mathbb{R}^+$ solution $x(t)$ for which there exists some $\tau > 0$ such that $V(x(\tau)) = V(x(0))$ is an equilibrium then the system is quasi-monostable.* 2° *If the assumptions of 1° hold and either $V(x) \to \infty$ for $|x| \to \infty$ or all solutions of the system are bounded, then the system is quasi-gradient like.* 3° *If the assumptions of 2° hold and the set $\mathcal{E}$ is discrete (i.e. it consists of isolated equilibria only) then the system is gradient-like.*

Since all solutions of (4) are bounded, we can apply 2°. From (6) it follows that system's equilibria, if they exist, coincide with the critical points of V; therefore the local minima are stable while the local maxima and the saddle points are unstable in the sense of Liapunov. Computing the Hessian matrix of V at the possible critical points we find

$$\mathcal{H}(V)(\bar{u}) = \lambda G'(\lambda\bar{u}) + (\lambda G'(\lambda\bar{u}))\Gamma(\lambda G'(\lambda\bar{u})) \tag{8}$$

where $G'(\lambda\bar{u}) := \mathrm{diag}\{g_1'(\lambda\bar{u}_1), \ldots, g_N'(\lambda\bar{u}_N)\}$ and Γ is the symmetric matrix of the weights ν_{ij}. If Γ is nonnegative definite then $\mathcal{H}(V)(\bar{u})$ is positive definite and all critical points are minima hence stable; if the system is gradient like then the equilibria are asymptotically stable. All these results may be summarized as follows

Theorem 2. *Consider system (4) with the symmetry assumption $\nu_{ij} = \nu_{ji}$. Then the system is quasi-gradient like and if the equilibria set $\mathcal{E}$ is discrete, it is gradient like. Moreover, if the matrix of the weights Γ is nonnegative definite ($\Gamma \geq 0$) all equilibria are Liapunov stable and if $\mathcal{E}$ is discrete their stability is also asymptotic.*

C. We shall discuss now the problem of the equilibria set $\mathcal{E}$ that has to be non-void. The equilibria are solutions to the system

$$u_i - b_i + \sum_1^N \nu_{ij} g_j(\lambda u_j) = 0\ ,\ i = 1, \ldots, N \tag{9}$$

and they are the *fixed points* of the one parameter family of mappings $F^\lambda : \mathbb{R}^N \mapsto \mathbb{R}^N$ defined by the N-dimensional vector with the components

$$F_i^\lambda(u_1, \ldots, u_N) = b_i - \sum_1^N \nu_{ij} g_j(\lambda u_j) \tag{10}$$

We may use the *contraction principle* on $\mathbb{R}^N$ and find that if the contraction condition

$$\lambda\left(\max_i \sum_1^N |\nu_{ij}| L_j\right) < 1 \tag{11}$$

is met, then each mapping of the family *has a unique fixed point* in $\mathbb{R}^N$ for fixed $\lambda > 0$ satisfying (11). Here L_j are the Lipschitz constants of the sigmoid functions g_i and the norm on $\mathbb{R}^N$ is $\|x\| = \max_i |x_i|$ with the induced norm for matrices. Making use of the properties of $g_i(\cdot)$ it is possible to introduce the "swollen" variables $\bar{\lambda}\bar{u}_i$, $\bar{\lambda} > 1$ and to obtain the contraction condition (11) as

$$\lambda\left(\max_i \sum_1^N |\nu_{ij}| L_j\right) < \bar{\lambda} \tag{12}$$

which gives existence and uniqueness of the equilibrium in $\mathbb{R}^N$ for each $\lambda > 0$. Taking into account also Theorem 2 we may state

Theorem 3. *Consider system (4) under the basic assumptions on the sigmoid functions. Then there exists for each* $\lambda > 0$ *a unique equilibrium point* $\bar{u}^\lambda \in \mathbb{R}^N$ *which is globally asymptotically stable.*

With this result and taking into account the restriction for the matrix Γ ($\Gamma \geq 0$) some important basic functionality conditions are secured and we may proceed to investigate specific properties of the NN like analog circuit described above.

3 Order-Preserving Properties of the KWTA Circuit

A. The functional properties of the circuit have thus to be ensured by its unique equilibrium and these properties are first of all *order-preserving properties*. As described in Section 1, the equilibrium output should be in some neighborhood of an associated "corner" (vertex) of the hypercube $[-1, 1]^N$. Obviously the equilibrium output location closely to a "corner" (vertex) may be ensured by choosing $\lambda > 0$ high enough. Since the equilibrium is reached for $t \to \infty$, due to the sigmoidal properties an output displaying the equilibrium may occur while the circuit is still under transients. In order to give some robustness to this (very important) order-preserving property, it will be secured for the transient processes also.

If the sequence $\{d_i\}_i$ whose ordering has to be recognized is applied through the input currents, the circuit may be used starting at $t = 0$ from the "ground state" (corresponding to the zero initial conditions). We have the following

Proposition 1. *If the input sequence* $\{d_i\}_i$ *is* σ*-ordered and the corresponding* $\{b_i\}_i$ *are such that they are also* σ*-ordered, then for each* $\lambda > 0$ *and* $t > 0$ *the trajectory* $u^\lambda(0; t)$ *and the equilibrium* $\bar{u}^\lambda$ *are* σ*-ordered provided all time constants are equal, the nonlinear functions are identical and the weights* ν_{ij} *satisfy the following conditions:* $\nu_{ii} = \nu_0$, $\forall i$, $\nu_{ij} = \nu_1$, $\forall i \neq j$. *The same order-preserving properties are valid also for the output* $v^\lambda(0; t)$ *and* $\bar{v}^\lambda$.

The significance of the assumptions is that some "uniformity" of the equations with respect to the Hopfield-like structure is necessary to secure order-preserving for arbitrary input permutations (i.e. input order). It is easily seen that under these assumptions we shall have for arbitrary indices i, j

$$\frac{\mathrm{d}}{\mathrm{d}t}(u_i^\lambda - u_j^\lambda) = -\frac{1}{T_0}\left[u_i^\lambda - u_j^\lambda + (\nu_0 - \nu_1)(g(\lambda u_i^\lambda) - g(\lambda u_j^\lambda)) - b_i + b_j\right] \tag{13}$$

and denoting $\xi^\lambda(t) := u_i^\lambda(t) - u_j^\lambda(t)$ we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi^\lambda = -\frac{1}{T_0}\left[(1 + (\nu_0 - \nu_1)\lambda g'(\zeta(t)))\xi^\lambda - b_i + b_j\right] \tag{14}$$

and if the Cauchy formula of the variation of constants is used, then if $\xi^\lambda(0) = u_i^\lambda(0) - u_i^\lambda(0) \leq 0$ and $b_i - b_j < 0$ it follows that $\xi^\lambda(t) = u_i^\lambda(t) - u_i^\lambda(t) < 0$ for all $t > 0$.

Before analyzing the new circuit structure it is necessary to check nonnegative definiteness for the matrix Γ of the weights under the assumptions of Proposition 1. An induction approach over the principal minors of Γ shows that $\Gamma \geq 0$ iff the following inequalities are true

$$\nu_0 > 0 \quad , \quad -\frac{\nu_0}{N-1} \leq \nu_1 \leq \nu_0 \tag{15}$$

B. We now follow the line of [2] to obtain detailed asymptotic order-preserving. Inequalities (15) allow the following choices of the parameters and input sequences

$$\begin{aligned} &i) \quad \nu_0 > 0 \quad , \quad \nu_1 < 0 \quad , \quad \nu_0 > -\nu_1 \\ &ii) \quad 0 < \delta < \nu_0 + \nu_1 = \nu_0 - |\nu_1| \quad , \quad |d_i| \leq \nu_0 + \nu_1 - \delta \\ &iii) \quad b_i = d_i - \nu_1(N - 2K) \end{aligned} \tag{16}$$

We may state now

Theorem 4. *Assume the choice for Γ as in Proposition 1 with ν_0, ν_1 satisfying (15) and such that all options in (16) hold. Let $\{d_i\}_i$ be σ-ordered. Then, for $\varepsilon > 0$ arbitrarily small and $\lambda > 0$ sufficiently large the following properties hold: 1) The output equilibrium $\bar{v}^\lambda$ has its components ordered like those of d - the d_i's; moreover*

$$\bar{v}^\lambda_{\sigma(1)} > \ldots > \bar{v}^\lambda_{\sigma(K)} > 1 - \varepsilon > 0 > -1 + \varepsilon > \bar{v}^\lambda_{\sigma(K+1)} > \ldots > \bar{v}^\lambda_{\sigma(N)} \tag{17}$$

2) The equilibrium $\bar{u}^\lambda$ is globally asymptotically stable; 3) There exists some $t_* > 0$ such that for $t > t_*$ the transient output should satisfy

$$v^\lambda_{\sigma(1)}(t) > \ldots > v^\lambda_{\sigma(K)}(t) > 1 - \varepsilon > 0 > -1 + \varepsilon > v^\lambda_{\sigma(K+1)}(t) > \ldots > v^\lambda_{\sigma(N)}(t) \tag{18}$$

while for $t = t_*$ at least one output equals either $1 - \varepsilon$ or $-1 + \varepsilon$.

The significance of the above results is as follows. First of all, the network has the KWTA property at any time after t_*; this is important since the circuit operation required a stopping mechanism after all outputs reached $1-\varepsilon$ or $-1+\varepsilon$ (quasi-equilibrium state); this is secured also by order-preserving during transients. The Liapunov like results of Section 2 ensure only (global) asymptotic stability. Exponential stability (local) follows from the theorem on stability by the first approximation provided the linearized system has its eigenvalues with strictly negative real parts. The linearized system is as follows (see also (13))

$$T_0 \frac{\mathrm{d}z}{\mathrm{d}t} = -(I + \Gamma G'(\lambda\bar{u}))z \tag{19}$$

where $G'(\lambda\bar{u})$ is the diagonal matrix with $g_i'(\lambda\bar{u}_i) > 0$ as diagonal elements; if $\Gamma \geq 0$, $\Gamma G'(\lambda\bar{u})$ is such (the proof is straightforward) hence $I + \Gamma G'(\lambda\bar{u}) \geq I$ and has its eigenvalues real, positive and larger than 1; therefore all eigenvalues of (19) are real and lower than $-1/T_0$. Global exponential stability is more difficult to prove : the task may be accomplished based on the estimates given by the Liapunov function and the theorems called “of Persidskii type” in stability theory.

Worth mentioning that stability is valid for all $\lambda > 0$ - see (9) - hence the equilibrium at the origin, corresponding to the network reset after some input processing, is also globally asymptotically stable. Consequently no gain reset is necessary in network operation.

4 Conclusions and Future Research

The set of results concerning the KWTA neural network contains several design conditions ensuring functionality; their feature is a rational interaction between equilibrium functional conditions and dynamical ones deduced from a naturally associated Liapunov function while in the standard approaches these aspects are discussed separately.

The static and dynamical “uniformity” of the KWTA circuit gave to (4) a structure almost identical to the “model system” of [10] which reads

$$\frac{\mathrm{d}z}{\mathrm{d}t} + z + Ph(z) + f(t) = 0 \tag{20}$$

where $h : \mathbb{R}^N \mapsto \mathbb{R}^N$ has as components the sigmoid like functions $h_i(z_i)$, P is a symmetric positive definite matrix. The paper [10] is a pioneering one concerning the qualitative behavior of the systems with several equilibria, but neural networks are never mentioned. As a counterpart, the paper is developing a comparison of the properties of the model system (which is finite dimensional) and those of a convolution system

$$x(t) + P\int_0^t G(t-\tau)h(x(\tau))\mathrm{d}\tau + f(t) = 0 \tag{21}$$

with G a diagonal and bounded kernel. For the dynamics of the neural networks these comparison results may be useful for robustness analysis with respect to the time delays occurring on network interconnections. In general such transfer of concepts among various fields is only fruitful in pure and applied research. This final remark could be true also for another methodological hint: replace high gain by adaptive gain since they produce similar transient properties being thus equivalent in some sense [7].

References

1. Brockett, R.W.: Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. Lin. Algeb. Applicat. 146, 79–91 (1991)
2. Calvert, B.D., Marinov, C.A.: Another k-winners-take-all analog neural network. IEEE Trans. Neural Networks 11, 829–838 (2000)
3. Danciu, D.: Dynamics of Neural Networks as Nonlinear Systems with Several Equilibria. In: Porto, A.B., Pazos, A., Buno, W. (eds.) Advancing Artificial Intelligence through Biological Process Applications, pp. 331–357. Medical Information Science Reference, London (2008)
4. Danciu, D., Rasvan, V.: Stability results for cellular neural networks with time delays. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 366–373. Springer, Heidelberg (2005)
5. Danciu, D., Rasvan, V.: Dynamics of neural networks - some qualitative properties. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 8–15. Springer, Heidelberg (2007)
6. Danciu, D., Rasvan, V.: Neural Networks - Equilibria, Synchronization, Time Lags. In: Rabunal, J.R., Dorado, J., Pazos, A. (eds.) Encyclopedia of Artificial Intelligence. Information Science Reference, pp. 1219–1225. Idea Group, London (2008)
7. Drăgan, V., Halanay, A.: Stabilization of linear systems. Birkhäuser, Basel (1999)

8. Majani, E., Erlanson, R., Abu Mostafa, Y.: On the K-winners-take-all network. In: Touretzky, D.S. (ed.) Advances in Neural Information Processing Systems, vol. 1, pp. 634–642. Morgan Kaufmann, San Mateo (1989)
9. Marinov, C.A.: Dynamical systems modelling computational circuits. Math. Reports 9(59), 55–60 (2007)
10. Popov, V.M.: Monotonicity and Mutability. Journ. Diff. Eqs. 31, 337–358 (1979)
11. Rasvan, V.: Dynamical systems with several equilibria and natural Liapunov functions. Archivum mathematicum (Brno) 34, 207–215 (1998)

Fast Evaluation of Connectionist Language Models

F. Zamora-Martínez[1], M.J. Castro-Bleda[2], and S. España-Boquera[2]

[1] Departamento de Ciencias Físicas, Matemáticas y de la Computación
Universidad CEU-Cardenal Herrera
46115 Alfara del Patriarca (Valencia), Spain
[2] Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Valencia, Spain
`{fzamora,mcastro,sespana}@dsic.upv.es`

Abstract. Connectionist language models offer many advantages over their statistical counterparts, but they also have some drawbacks like a much more expensive computational cost. This paper describes a novel method to overcome this problem. A set of normalization values associated to the most frequent n-grams is pre-computed and the model is smoothed with lower n-gram connectionist or statistical models. The proposed approach is favourably compared to standard connectionist language models and with statistical back-off language models.

1 Introduction

Language modeling is the attempt to characterize, capture and exploit regularities in natural language. In problems such as automatic speech or handwritten text recognition, machine translation, text classification or other pattern recognition tasks, language models (LM) are useful to adequately guide the search for the optimal response and to increase the success rate of the system.

Under the statistical framework to language modeling [1,2], n-grams are the most popular LM [3]. They are simple and robust models which adequately capture local restrictions between words. Their parameters are learned from text corpora using the occurrence frequencies of subsequences of n word units. Deciding a value for n is a trade-off: larger values of n can capture longer-term dependencies between words; however, the number of different n-grams grows very fast (exponentially) with n, and more parameters requires more training data to estimate them reliably. To alleviate this problem, some techniques can be applied, such as smoothing or clustering techniques, class n-grams, integrated models, etc. (for a review, see [2]).

We have worked in a different line to solve the above problem by applying neural networks (NN) to language modeling [4,5,6]. Despite their theoretical advantages, these LM are more expensive to compute. This paper is structured as follows: The connectionist approach to language modeling is introduced in

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 33–40, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Section 2. A novel technique to speedup the computation of connectionist language models is presented in Section 3. Learning n-gram models with feed-forward neural networks is described in Section 4, along with some results and analysis of their performance with the LOB corpus. The evaluation of the proposed approach to speedup the evaluation of such language models is presented in Section 5. Finally, we draw some conclusions and propose new directions in connectionist language modeling in Section 6.

2 Connectionist Language Models

Under a statistical framework, a language model is used to assign to every possible word sequence S an estimation of the a priori probability of being the correct system response $p(S) = p(s_1 \ldots s_{|S|})$. Statistical language models are usually based on the prediction of each linguistic unit in the sequence given the preceding ones [1,2]:

$$p(s_1 \ldots s_{|S|}) = p(s_1) \cdot \ldots \cdot p(s_{|S|}|s_1 \ldots s_{|S|-1}) = \prod_{i=1}^{|S|} p(s_i|h_i)\,, \tag{1}$$

where $h_i = s_1 \ldots s_{i-1}$ denotes the history from which unit s_i has to be predicted. The number of parameters to estimate becomes intractable as the length of the sentence increases. n-gram models [1] are the most extended method to reduce this number approximating the probability of a word as if only the last $n-1$ words have influence:

$$p(s_1 \ldots s_{|S|}) \approx \prod_{i=1}^{|S|} p(s_i|s_{i-n+1} \ldots s_{i-1})\,. \tag{2}$$

A Neural Network Language Model (NN LM) is a statistical LM which follows the same equation (2) as n-grams and where the probabilities that appear in that expression are estimated with a NN [4,5,6]. The model naturally fits under the probabilistic interpretation of the outputs of the NNs: if a NN, in this case a Multilayer Perceptron (MLP), is trained as a classifier, the outputs associated to each class are estimations of the posterior probabilities of the defined classes [7].

The training set for a LM is a sequence $s_1 s_2 \ldots s_{|S|}$ of words from a vocabulary Ω. In order to train a NN to predict the next word given a history of length $n-1$, each input word must be encoded. A natural representation is a local encoding following a "1-of-$|\Omega|$" scheme. The problem of this encoding for tasks with large vocabularies (as is typically the case) is the huge size of the resulting NN. We have solved this problem following the ideas of [4,6], learning a distributed representation for each word. Figure 1 illustrates the architecture of the feed-forward NN used to estimate the NN LM:

- The input is composed of words $s_{i-n+1}, \ldots, s_{i-1}$ of equation (2). Each word is represented using a local encoding.
- P is the projection layer of the input words, formed by $P_{i-n+1}, \ldots, P_{i-1}$ subsets of projection units. The subset of projection units P_j represents the distributed encoding of input word s_j. The weights of this projection layer are linked, that is, the weights from each local encoding of input word s_j to

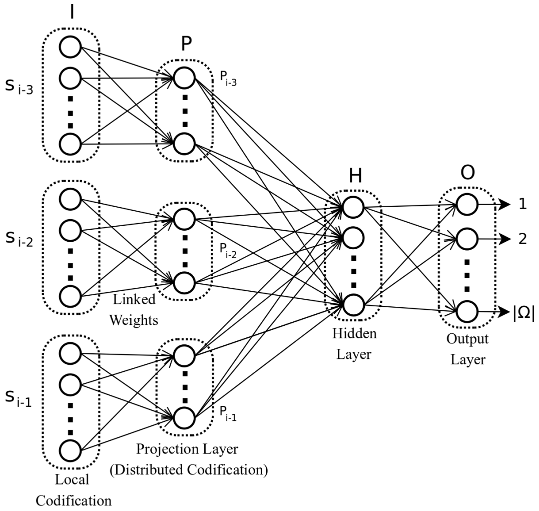


Fig. 1. Architecture of the continuous space NN LM. The input words are s_{i-n+1}, $\dots, s_{i-1}$ (in this example, the input words are s_{i-3}, s_{i-2}, and s_{i-1} for a 4-gram). I, P, H and O are the input, projection, hidden and output layer, respectively, of the MLP.

the corresponding subset of projection units P_j are the same for all input words j. After training, the codification layer is removed from the network by pre-computing a table of size $|\Omega|$ which serves as a distributed encoding.

- H denotes the hidden layer.
- The output layer O has $|\Omega|$ units, one for each word of the vocabulary.

This NN predicts the posterior probability of each word of the vocabulary given the history, i.e., $p(s_i|s_{i-n+1}\dots s_{i-1})$ is computed by feeding the encoded input words $s_{i-n+1},\dots,s_{i-1}$ to the input layer. A single forward pass of the MLP gives $p(\omega|s_{i-n+1}\dots s_{i-1})$ for every word $\omega \in \Omega$.

The advantages of the connectionist approach to language modeling are due to their automatic estimation (as with statistical LM), the lowest (in general) number of parameters of the obtained models and the automatic smoothing performed by the neural networks estimators. Estimation of NN LM in ambitious tasks is needed in order to fulfill these advantages. This is not trivial: the larger the lexicon is, the larger the number of parameters the neural network needs. We think that the use of a distributed representation for each word of the vocabulary (as in [4]) can be a successful approach for tasks with large lexica, but the high cost of using NN LMs remains a problem.

3 Fast Evaluation of NNLMs

Despite the theoretical advantages of connectionist language models, some difficulties appear when they are used in practice. In order to compute the language model probability of a given sentence, the number of n-gram conditional probabilities to be computed is roughly the length (in words) of the sentence. However, when the same language model is used in speech or handwritten recognition or in translation tasks, the number of language model lookups is typically huge since there are many different hypotheses to be considered.

The activation function of the hidden neurons is usually the logistic or the hyperbolic tangent. The output units can use the logistic or the softmax activation function whose values lie between zero and one. The softmax activation function [7] ensures also that the output values sum to one:

$$o_i = \frac{exp(a_i)}{\sum_{j=1,\ldots,|\Omega|} exp(a_j)}, \tag{3}$$

being a_i the activation value of the i-th output unit and o_i is its output value.

The softmax gives much better results in term of perplexity than the logistic function, but requires the computation of every output value, even though only some of them are used, due to the normalization term of equation (3). The computation of the output layer dominates the cost of the forward pass in a typical NN LM topology. This problem has been noticed before in the literature and two solutions are proposed to this regard [6]:

- To decrease the size of the output layer by removing less frequent words. This sort list approach requires smoothing language modeling techniques to estimate the conditional probability of rare words.
- To collect and to group together all the n-grams which share a common $n-1$ prefix, since all of them require the same forward pass. Note that this can also be useful when other activation functions are used because the number of hidden layers computations is anyway reduced. Unfortunately, this is not always possible in some online systems.

The proposed approach consists on pre-computing and storing the softmax normalization constants most probably needed during the LM evaluation, since the cost of retrieving this value from a table is negligible compared with the cost of computing it. A space/time trade-off has to be considered: the more space is dedicated to store pre-computed softmax normalization constants, the more time reduction can be obtained. When a given normalization constant is not found, it can be computed on-the-fly or some kind of smoothing must be applied. We have followed the latter idea: when a softmax normalization constant is not found, another simpler model (for instance, a lower order NN LM or statistical n-gram model) is used.[1] An experimental evaluation is needed to study how the combined language model behaves and how it is affected by the number of pre-computed normalization constants.

4 Estimation of the NNLMs

In order to test the proposed technique, experiments with the LOB text corpus [8] have been conducted. The whole corpus consists of 53 694 sentences comprising

---

[1] Note that storing the softmax normalization constants for a bigram NN LM only needs a $|\Omega|$-sized table.

Table 1. Number of weights of the NN LMs configurations. The first column shows the size of the projection of each word and the second column shows the size of the hidden layer. Following, the number of weights from the projection layer to the hidden layer for bigram, trigram and 4-gram NN LMs are shown and, finally, the number of weights from the hidden layer to the output layer.

Configuration		# weights PH			# weights HO
Proj	H	Bigram	Trigram	4-gram	
80	128	10 368	20 608	30 848	387 258
128	192	24 768	49 344	73 920	579 386
192	192	37 056	73 920	110 789	579 386

more than one million word instances. A subcorpus with a lexicon of 3 000 words has been built as follows: from a random ordering of the sentences, we have selected those whose lexicon lies in the first different 3 000 words. This subcorpus was partitioned into three sets:

- Training 4 303 sentences with 37 606 running words.
- Validation 600 sentences with 5 348 running words.
- Test 600 sentences with 5 455 running words.

We have estimated bigrams, trigrams and 4-gram NN LMs for the subcorpus. In order to select a good configuration for each NN LM, scanning of the network topology and other parameters has been performed using the validation set. Three different topologies have been tested, as shown in Table 1, which also shows the number of weights of the trained NN LMs.

The hyperbolic tangent function was used in the hidden layer and the softmax function was chosen for the output units. Training has been performed with one of the most standard methods to train MLPs, the back-propagation with momentum algorithm [9]. The stochastic version with weight-decay version of this algorithm has been used using the April toolkit [10], developed for neural networks and pattern recognition tasks by the authors. Every NN LM has been trained during 50 to 80 epochs using the mean square error on the validation set as stopping criteria.

The measure of the quality of a estimated LM is usually the perplexity of a test set which can be intuitively interpreted as the geometric mean of the branch-out factor of the language [2,11]: a language with perplexity x has roughly the same difficulty as another language in which every word can be followed by x different words with equal probabilities. Thus, the lower the perplexity of a language model, the better the language model is.

The perplexity of the bigram, trigram and 4-gram NN LMs on the validation sets are shown in Table 2. For every n-gram, a mixed NN LM has been built by linearly combining the NN LMs models. The coefficients have been obtained by using the SRI Language Modeling Toolkit [12] with the "Compute Best Mix" option. Note that the best perplexities with the validation set are always obtained with the Mixed NN LM. Therefore, test set perplexities have been computed for

Table 2. Validation set perplexity for the estimated NN LMs

Language Model	Bigram	Trigram	4gram
NN LM 80–128	82.17	73.62	74.07
NN LM 128–192	82.52	73.30	72.50
NN LM 192–192	80.01	71.90	71.91
Mixed NN LM	78.92	71.34	70.63

Table 3. Test set perplexity of the best NN LMs and SRI

Language Model	Bigram	Trigram	4-gram
Mixed NN LM	88.72	80.94	79.90
SRI	88.29	83.19	87.05

the Mixed NN LM and conventional n-grams estimated with the SRI Toolkit using the modified Kneser-Ney backoff smoothing (see Table 3). As can be observed, for trigrams and 4-grams, the Mixed NN LM outperform the statistical LMs, but for bigrams the resulting NN LM and SRI models are the equivalent.

5 Evaluation of the Proposed Approach

We have tested the constants pre-computation and smoothing approach presented in Section 3. The procedure is as follows:

1. Choose a hierarchy of models, from higher order n-grams downto bigrams.
2. Pre-compute the softmax normalization constants for the K more frequent $(n-1)$-grams in the training set for each n-gram NN LM bigger than bigram. Store the constants in a table.
3. Pre-compute every softmax normalization constants for every word of the lexicon using the bigram NN LM. Store the constants in a $|\Omega|$-sized table.
4. During the test evaluation, for each token: search the softmax pre-computed constant associated to the $(n-1)$-length prefix of the token in the table. If the constant is in the table, calculate the probability; otherwise, switch to the inmediately inferior NN LM.

Figure 2 (left) shows the test set perplexities for different number of pre-computed softmax normalization constants. The test set perplexity of the 4-gram Mixed NN LM is 79.90 (Table 3) and the best smoothed perplexity obtained with our approach is 80.78 (Figure 2 (left)).

Afterwards, we have tested another smoothed model using the statistical bigram which gives better perplexity than the best bigram NN LM, and the resulting perplexity, varying the number of pre-computed softmax normalization constants, is shown in Figure 2 (right). As can be observed, a better perplexity, 79.02, is obtained with this new LM.

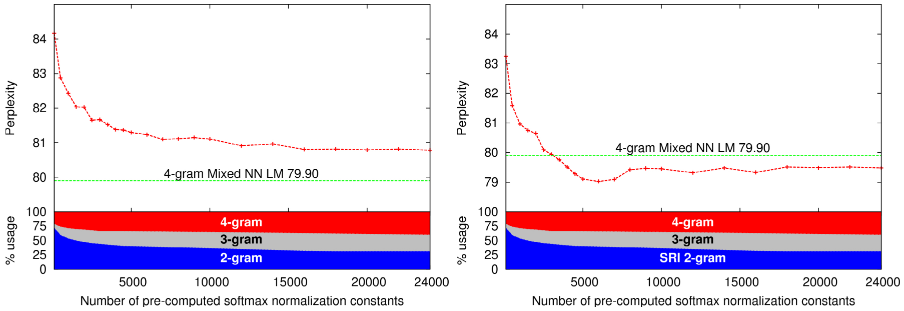


Fig. 2. Influence of the number of pre-computed softmax normalization constants in the test set perplexity of the proposed approach for the Mixed NN LMs (left), and the Mixed NN LMs and statistical bigram (right)

In both cases (using only NN LM or combining with statistical n-grams), the time needed to compute NN LM lookups is 1.94×10^{-4} seconds per token, versus the original NN LM which requires 6.43×10^{-3} seconds per token, that is, a speedup of 33 times faster has been obtained.

In order to study the efficience of the smoothed model, we have measured the time which is needed for each processed token if we only use the true NN LM. With this non-smoothing approach, when a softmax normalized constant is not in the pre-computed table, a full forward-pass of the NN LM is done in order to calculate it. The best time achieved, when 24 000 pre-computed softmax normalization constants are used, is 1.82×10^{-3} seconds per token for the 4-gram Mixed NN LM, one order of magnitude bigger than with the smoothed model. The non-smoothed 3-gram Mixed NN LM needs 9.68×10^{-4} seconds per token, five times more than the corresponding smoothed model. The effect of the number of pre-computed softmax normalization constants of this non-smoothing approach can be observed in Figure 3. Notice that in this case the perplexity of the NN LMs is not affected (see Table 3).

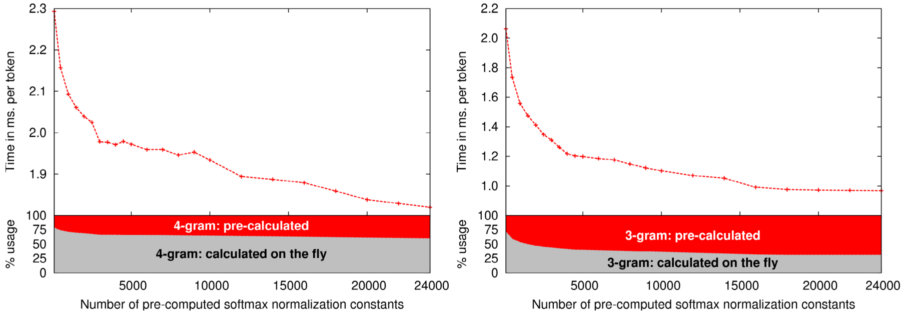


Fig. 3. Influence of the number of pre-computed softmax normalization constants in the evaluation time of the non-smoothed approach for the 4-gram Mixed NN LMs (left), and for the 3-gram Mixed NN LMs (right)

6 Discussion and Conclusions

A novel method to allow fast evaluation of connectionist language models has been presented. A set of normalization values associated to the most frequent n-grams is pre-computed and the model is smoothed with lower n-gram connectionist or statistical models. The proposed approach has been tested by training a 4-gram NN LM in a 3 000-vocabulary task from the LOB corpus. The best perplexity is obtained by combining the Mixed NN LM with a statistical bigram. A speedup of 33 times faster is achieved with this lexicon size. Nevertheless, higher speedups would be achieved for bigger lexicon sizes. Our next goal is to train more complex NN LMs and to integrate them in a recognition or translation systems.

Acknowledgements

This work has been partially supported by the Spanish Ministerio de Educación y Ciencia (TIN2006-12767 and TIN2008-06856-C05-02).

References

1. Bahl, L.R., Jelinek, F., Mercer, R.L.: A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Trans. on PAMI 5(2), 179–190 (1983)
2. Jelinek, F.: Statistical Methods for Speech Recognition. In: Language, Speech, and Communication. The MIT Press, Cambridge (1997)
3. Katz, S.: Estimation of Probabilities From Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Trans. on ASSP 34(3), 400–401 (1987)
4. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A Neural Probabilistic Language Model. Journal of Machine Learning Research 3(2), 1137–1155 (2003)
5. Castro, M.J., Prat, F.: New Directions in Connectionist Language Modeling. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2686, pp. 598–605. Springer, Heidelberg (2003)
6. Schwenk, H.: Continuous space language models. Comput. Speech Lang. 21(3), 492–518 (2007)
7. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Oxford (1995)
8. Johansson, S., Atwell, E., Garside, R., Leech, G.: The Tagged LOB Corpus: User's Manual. Technical report (1986)
9. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: PDP: Computational models of cognition and perception, vol. I, pp. 319–362. MIT Press, Cambridge (1986)
10. España-Boquera, S., Zamora-Martínez, F., Castro-Bleda, M., Gorbe-Moya, J.: Efficient BP Algorithms for General Feedforward Neural Networks. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4527, pp. 327–336. Springer, Heidelberg (2007)
11. Rosenfeld, R.: Adaptative statistical language modeling: A maximum entropy approach. PhD thesis, Carnegie Mellon University (1994)
12. Stolcke, A.: SRILM: an extensible language modeling toolkit. In: Proceedings of the Int. Conf. on Spoken Language Processing, pp. 901–904 (2002)

Improving the Consistency of AHP Matrices Using a Multi-layer Perceptron-Based Model

Jose Antonio Gomez-Ruiz[1], Marcelo Karanik[2], and José Ignacio Peláez[1]

[1] Department of Computer Science and Artificial Intelligence, University of Málaga,
29071 Málaga - Spain
[2] Artificial Intelligence Research Group, National Technological University,
3500 Resistencia - Argentina
{janto,jignacio}@lcc.uma.es, marcelo@frre.utn.edu.ar

Abstract. The Analytic Hierarchy Process (AHP) uses hierarchical structures to arrange comparing criteria and alternatives in order to give support in decision making tasks. The comparisons are realized using pairwise matrices which are filled according to the decision maker criterion. Then, matrix consistency is tested and priorities of alternatives are obtained. If a pairwise matrix is incomplete, two procedures must be realized: first, to complete the matrix with adequate values for missing entries and, second, to improve the consistency matrix to an acceptable level. In this paper a model based on Multi-layer Perceptron (MLP) neural networks is presented. This model is capable of completing missing values in AHP pairwise matrices and improving its consistency at the same time.

Keywords: AHP, decision support systems, pairwise matrix reconstruction, multi-layer perceptron, neural networks.

1 Introduction

In decision theory area, many methods are used to give support in order to select an alternative based on a set of preferences, points of view, viability aspects, etc. One of these methods is the Analytic Hierarchy Process (AHP) [1] which compares and ranks alternatives according to a set of criteria and sub-criteria arranged in a hierarchical structure. The preference between different criteria is expressed in natural language using a predetermined scale and then it is translated into numerical preference values. At each level of the hierarchy, the criteria are compared in pairs using pairwise matrices. The criteria are used to label rows and columns of the pairwise matrix and the intersection between them is filled with a numerical preference value. The pairwise matrix consistency is a relevant factor in AHP and, to calculate it, it is necessary that the pairwise matrix is complete. If a pairwise matrix is incomplete, the missing entries must be filled with values that maintain adequate levels in the matrix consistency. Finally, the principal eigenvector of the matrix is calculated and the sum of their elements is normalized, obtaining the relative preference for the alternatives.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 41–48, 2009.
© Springer-Verlag Berlin Heidelberg 2009

There are several methods that complete missing values in pairwise matrices [2-4] and others that improve the matrix consistency [2], [5-6], however, it is not possible to use them simultaneously because improving methods work with complete matrices.

If a pairwise matrix with missing elements is considered to be a pattern distorted to recognize, the correct pattern associated will be the most similar complete and consistent pairwise matrix that can be related using a generalization tool. In this sense, neural networks are a proven generalization method used in artificial intelligence and they can be trained to associate these incomplete patterns to more similar correct ones. A Multi-layer Perceptron (MLP) based model which completes missing values in pairwise matrices and, at the same time, is able to improve its consistency, is described and tested with matrices with different levels of inconsistency. In section 2, the Analytic Hierarchy Process, the calculation of priorities and the concept of matrix consistency are briefly explicated; in section 3, matrix reconstruction using MLP-based model is explained; simulations are shown in section 4; and, finally, conclusions are presented in section 5.

2 AHP Description

In AHP [1], the problem is represented using a hierarchy which is an abstraction of the whole problem. The main goal is placed at the root of the hierarchical tree where the alternatives derive from it. The basic idea of AHP is to arrange goals, attributes and issues in a hierarchic structure in order to raise two objectives: to obtain a complete view of relationships inherent in the situation and to provide a comparison mechanism at each level that uses the same order of magnitude [7]. Entities are arranged in levels in such way that entities of the same level do not have a connection with each other but they are thoroughly connected with entities of adjacent levels. These relations determine the hierarchy structure of criteria and alternatives to evaluate.

The comparison begins from the criteria levels to alternative levels using pairwise matrices in which the decision maker fills each upper diagonal element with a value obtained from a scale of verbal judgments [1]: (equal, moderately more, strongly more, very strongly more, extremely more). This scale has associated numerical judgments (1, 3, 5, 7, 9) and compromises (2, 4, 6, 8) between these values. The inferior triangle portion of the matrix is completed with the reciprocal judgment values in the top triangle portion, obtaining a reciprocal matrix (considering the axioms of reciprocal, homogeneity, dependence and expectation defined by Saaty [8]). The priorities from criteria pairwise matrices are obtained by calculating the principal eigenvector (using, for example, the power method [9-10]). Then, the eigenvector values must be normalized before using them in the next level.

The whole process refers to matrix algebra calculations but there is an important issue to consider: the consistency of pairwise matrices. For example, suppose that the decision maker says “I strongly prefer A over B and I very strongly prefer B over C”. If he also says “I strongly prefer C over A”, this last judgment is inconsistent and it must be corrected before using it. Clearly, the consistency of judgments in the decision making is a factor to consider carefully. For instance, Saaty [1] proposes a multiplicative reciprocity between the elements of pairwise matrices while other authors define a consistency ratio based on percentiles [5], or analyze the matrix consistency based on the characteristic polynomial of a positive reciprocal matrix [4]. In this paper the consistency ratio based on the multiplicative reciprocity is used (this ratio is frequently used for testing the consistency on matrix reconstruction [2], [11]).

In a reciprocal positive matrix, the largest eigenvalue is greater than or equal to its order. A reciprocal matrix **A** is perfectly consistent if its largest eigenvalue is equal to the matrix order [1]. The consistency index (CI) is defined as:

$$\mathrm{CI} = \frac{\lambda_{\max} - n}{n - 1} \tag{1}$$

where $\lambda_{\max}$ is the largest eigenvalue of matrix **A** and n its order.

Then, the consistency ratio (CR) is calculated in the following way:

$$\mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}} \tag{2}$$

where RI is the Random Index obtained from 500 positive reciprocal matrices randomly created with Saaty's scale values (see Table 1).

Table 1. Values of RI

n	3	4	5	6	7	8
RI	0.58	0.9	1.12	1.24	1.32	1.41

If the CR is small, about 10% or less, the eigenvector, **w**, associated to the largest eigenvalue of **A** is accepted. In other words, to determine if a pairwise matrix is consistent, its CR must be under 0.1 [1]. This ratio represents approximately 25% of matrices with order 3 but this percentage decreases drastically when the matrix order grows [12]. This implies that it is hard to obtain large order consistent matrices.

3 Matrix Reconstruction

Before using AHP, missing values in pairwise matrices must be completed and its consistency must be acceptable. Most reconstruction methods take an incomplete pairwise matrix and fill the missing values using diverse strategies [2] and the CR is used to evaluate the performance of methods because it synthesizes the matrix consistency in just one value. After that, if the matrix CR is not under 0.1, some values must be modified using improving consistency methods [6], [13]. This clearly implies that two kinds of methods must be required in order to obtain adequate pairwise matrices.

On the other hand, we propose a MLP-based model that is able to complete missing entries and to improve the matrix CR at the same time, without extra calculations. It was compared with the Revised Geometric Mean [3] and the Connecting Path [14] methods showing that it is able to fill missing values in AHP pairwise matrices with similar performance.

Basically, the proposed model uses a MLP neural network [15]. In the learning process, the MLP adapts the connection weights attempting to minimize the difference between the network outputs and the desired outputs. When an acceptable average training error is reached, the network is capable of recognizing incomplete patterns.

The MLP architecture receives the top triangular portion of a pairwise matrix, including missing elements (represented by -1), and returns the estimate missing and non missing elements, i.e. it returns all estimated values of the top triangular portion of the matrix. The number of input and output neurons is equal to the number of elements of the top triangular portion of the matrix. Therefore, if n is the pairwise matrix order, the MLP have $n(n - 1)/2$ inputs and outputs neurons. We propose to use only one hidden layer with thirty neurons. These parameters were selected after several tests whose results did not improve if more hidden layers or more hidden neurons were used. For example, the following reciprocal matrix of order 4, **A**, with two missing elements (represented with '–' character):

$$\mathbf{A} = \begin{bmatrix} 1 & - & a_{13} & a_{14} \\ - & 1 & - & a_{24} \\ 1/a_{13} & - & 1 & a_{34} \\ 1/a_{14} & 1/a_{24} & 1/a_{34} & 1 \end{bmatrix}$$

has associated an MLP architecture with six input neurons, thirty hidden neurons and six output neurons. This configuration is shown in Figure 1.

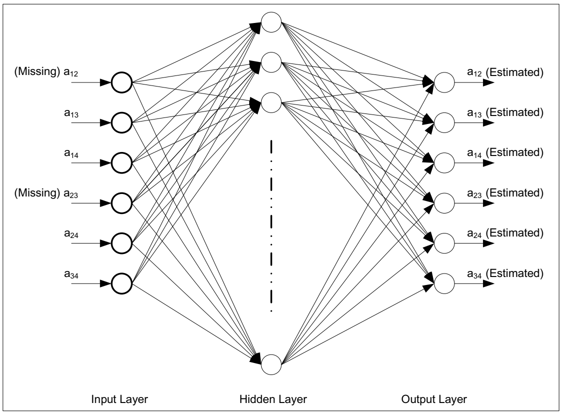


Fig. 1. MLP Architecture

The neurons in hidden and output layers use hyperbolic tangent transfer function because a better performance in the learning process is revealed. Due to the fact that hyperbolic tangent transfer function is defined in the interval [-1, 1], training patterns (inputs and desired outputs) must be normalized in that interval before being presented to network. To do that, the interval [-1, 1] is divided into seventeen uniform spaced values creating a correspondence between these values and the Saaty's scale values. In this way, the normalization process consists of replacing every matrix value for its corresponding normalized value.

The main idea is that the MLP learns distinct configurations of consistent matrices and it can relate an incomplete pairwise matrix with the most similar of them. To reach this objective, the MLP uses random reciprocal pairwise matrices with low CR (under 0.1) for training process.

When the training process has finished, the network is ready to recognize incomplete pairwise matrices. Nevertheless, it is not possible to use the pairwise matrix with Saaty's scale values directly because the network only recognizes normalized values into interval [-1, 1]. For this reason, inputs must be previously normalized following the same criteria used with training data. Finally, the obtained results are transformed into their original form and the missing value positions are filled with the corresponding transformed network outputs.

Once the values are obtained, there are two possibilities: (1) to complete only the missing entries, maintaining the non missing matrix values unchanged; (2) to complete the missing entries as well as modify some non missing matrix values in order to improve the matrix consistency.

In the first alternative, missing values positions are identified and replaced by the transformed network output values in those positions. The remaining elements of the matrix are not changed. In the second alternative, the same substitution previously mentioned is made, but also some non missing values are changed attempting to improve the CR. To realize this modification, it is necessary to define a measure capable of controlling the magnitude of these changes. In this sense, it is defined the Saaty's Scale Distance (*SSD*) as the absolute difference between two values position in that scale. Specifically, given the Saaty's scale [1/9, 1/8, 1/7, 1/6, 1/5, 1/4, 1/3, 1/2, 1, 2, 3, 4, 5, 6, 7, 8, 9] and the corresponding positions for those values defined from 1 to 17, the *SSD* between two values, v_1 and v_2, is defined by:

$$SSD_{v1,v2} = \left| pos(v_1) - pos(v_1) \right| \tag{3}$$

where v_1 and v_2 belong to the Saaty's scale.

Notice that the possible values for SSD_{v1v2} are integers between 0 and 16, being 0 if the network output coincides with the original value. This mechanism permits to modify original values (missing and non missing) having control over changes. If it is necessary to complete only the missing values in a pairwise matrix, then SSD_{v1v2} is set to zero. Otherwise, a fixed distance must be selected (between 1 and 16). In Figure 2, the summarized process is shown.

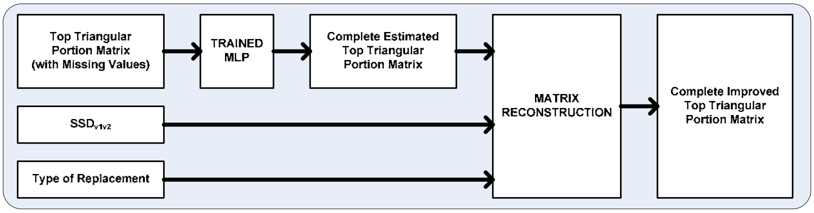

Top Triangular Portion Matrix (with Missing Values)
TRAINED MLP
Complete Estimated Top Triangular Portion Matrix
SSD$_{v1v2}$
Type of Replacement
MATRIX RECONSTRUCTION
Complete Improved Top Triangular Portion Matrix

Fig. 2. Matrix Reconstruction Process

4 Simulations

Simulations were realized using random reciprocal pairwise matrices with orders between 4 and 8, and between 1 and 4 missing values (1000 matrices for each matrix order and different sets for each quantity of missing values). Before training the MLP, some parameters must be set. The following parameters were selected after comparing the performance of different training simulations: Levenberg-Marquardt training algorithm (to improve the convergence of backpropagation algorithm); 100 training epochs; learning rate equal to 0.3 and minimum acceptable average error equal to 0.005.

After training, the proposed model was tested using the same type of random reciprocal pairwise matrices that had been used in the training process (1000 matrices for each matrix order and different sets for each quantity of missing values). All these matrices contained values belonging to Saaty's scale and they were created with low, medium and high levels of inconsistency in order to evaluate the reconstruction performance in all possible cases. In the simulations, the MLP proposed model was used in order to improve the CR with different values of SSD_{v1v2}.

The main idea is, first, to fill in missing entries and, second, to replace some of the non missing matrix values with values obtained by the neural network with an increasing SSD_{v1v2} (from one to sixteen) or with a decreasing SSD_{v1v2} (from sixteen to one) in order to evaluate the CR evolution (type of replacement, see Figure 2). For example, fixing SSD_{v1v2}=8, in an increasing way, means that an original non missing value is replaced if the distance between this and the value obtained by the neural network is smaller than or equal to 8. On the other hand when using decrements, non missing values are changed by their correspondent network outputs if the distance between them is bigger than or equal to 8. This test was realized for matrices with high, medium and low CR values.

Results of simulations are summarized in Figures 3, 4 and 5 in which median values of CR are plotted using matrix orders between 4 and 8.

For high level of inconsistency (Figures 3(a) and 3(b)), it is possible to reach acceptable CR values for all matrix orders, except for order 8. In Figure 3(a), it can be observed that, using increments of SSD_{v1v2} values, the average CR is gradually reduced for every increment and it tends to stabilize for distances further than 9. With

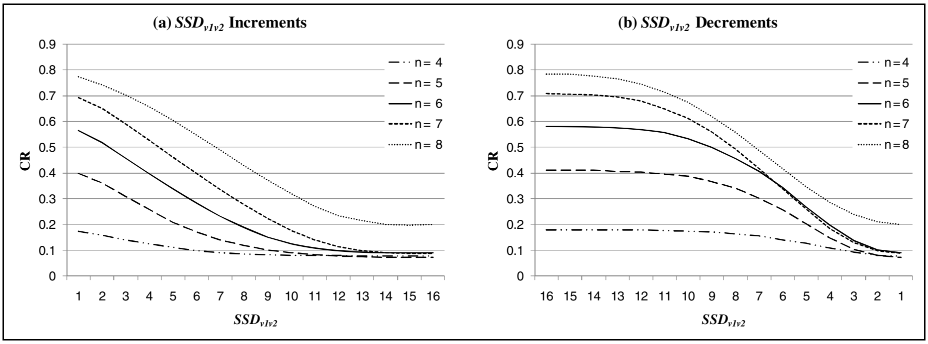


Fig. 3. CR Variation in Matrices with High Level of Inconsistency

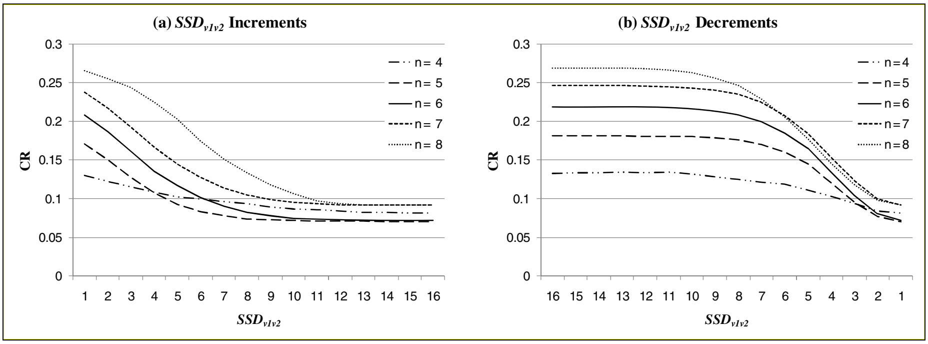


Fig. 4. CR Variation in Matrices with Medium Level of Inconsistency

decrements of SSD_{v1v2} (Figure 3(b)) there are not significant CR variations for bigger distance values but, from SSD_{v1v2} lower than 9, CR is considerably reduced to reach acceptable consistency levels.

For medium level of inconsistency (Figures 4(a) and 4(b)), the behavior is very similar to the observed for high levels, and the same tendency in maintained. In both cases, acceptable CR values are reached for all matrix orders (also for order 8).

For low level of inconsistency, a different behavior is observed (Figures 5(a) and 5(b)). When using SSD_{v1v2} increments (Figure 5(a)) the CR grows for small distance values and it is maintained for big ones. Consequently, in Figure 5(b), the CR does not change for big distances and it is increased using small SSD_{v1v2} values. Nevertheless, even if the MLP does not improve the matrix consistency for any matrix order, the obtained CR is sensibly under 0.1.

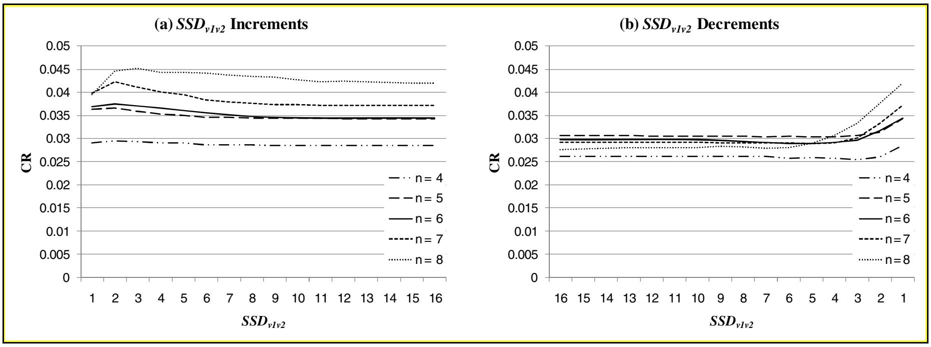


Fig. 5. CR Variation in Matrices with Low Level of Inconsistency

5 Conclusions

Using the MLP-based model, it is possible to improve the matrix consistency obtaining acceptable consistency ratios by changing some of the non missing values with their corresponding neural network outputs. This can be done using the Saaty's Scale Distance (SSD_{v1v2}) between the original and the estimated values. Since the network outputs are very similar to original values, a better performance in consistency improvement is observed if values of SSD_{v1v2} between 3 and 6 are used.

With the MLP-based model, the decision maker can complete those judgments which he considers relevant and let the neural network suggest the values for empty matrix positions. Also, if all pairwise matrix elements were completed beforehand, the proposed model could be used to indicate which values must be changed in order to improve the matrix consistency. In the latter case, the decision maker may take his experience or preferences into account to change specific values.

Acknowledgements

This work has been partially supported by the following projects: TIN2006-14285 (Ministry of Education of Spain), TIC-1615 (Proyec. Exc. Junta de Andalucia) and EIINRE571 (National Technological University, Argentina).

References

1. Saaty, T.L.: The Analytic Hierarchy Process. McGraw-Hill, New York (1980)
2. Chen, Q., Triantaphyllou, E.: Estimating Data for Multicriteria Decision Making Problems. Optimization Techniques. Encyclopedia of Optimization, 567–576 (2001)
3. Harker, P.T.: Alternative Modes of Questioning in the Analytic Hierarchy Process. Mathematical Modelling 9(3-5), 353–360 (1987)
4. Shiraishi, S., Obata, T., Daigo, M.: Properties of a Positive Reciprocal Matrix and their Application to AHP. Journal of the Operations Research Society of Japan 41(3), 404–414 (1998)
5. Pelaez, J.I., Lamata, M.T.: A Method for Improving the Consistency of Judgements. International Journal Of Uncertainty Fuzziness And Knowledge-Based Systems 10(6), 677–686 (2002)
6. Saaty, T.L.: Decision-making with the AHP: Why is the Principal Eigenvector Necessary. European Journal of Operational Research 145, 85–91 (2003)
7. Saaty, T.L.: How to Make a Decision - The Analytic Hierarchy Process. European Journal of Operational Research 48(1), 9–26 (1990)
8. Saaty, T.L.: Some mathematical concepts of the Analytic Hierarchy Process. Behaviormetrika 29, 1–9 (1991)
9. Burden, R.L., Faires, J.D.: Numerical Analysis. PWS-Kent, Boston (1993)
10. Chapra, S.C., Canale, R.P.: Numerical Methods for Engineers. McGraw Hill, New York (2006)
11. Hu, Y.C., Tsai, J.F.: Backpropagation Multi-layer Perceptron for Incomplete Pairwise Comparison Matrices in Analytic Hierarchy Process. Applied Mathematics and Computation 180(1), 53–62 (2006)
12. Pelaez, J.I., Lamata, M.T.: A New Measure of Consistency for Positive Reciprocal Matrices. Computers & Mathematics with Applications 46(12), 1839–1845 (2003)
13. Cao, D., Leung, L.C., Law, J.S.: Modifying Inconsistent Comparison Matrix in Analytic Hierarchy Process: A Heuristic Approach. Decision Support Systems 44(4), 944–953 (2008)
14. Harker, P.T.: Incomplete Pairwise Comparisons in the Analytic Hierarchy Process. Mathematical Modelling 9(11), 837–848 (1987)
15. Haykin, S.: Neural Networks: a Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1998)

Global and Local Modelling in Radial Basis Functions Networks

L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, G. Rubio, and J. Urquiza

Departamento de Arquitectura y Tecnología de los Computadores,
Universidad de Granada, 18071 Granada, Spain
`jherrera@atc.ugr.es`

Abstract. In the problem of modelling Input/Output data using neuro-fuzzy systems, the performance of the global model is normally the only objective optimized, and this might cause a misleading performance of the local models. This work presents a modified radial basis function network that maintains the optimization properties of the local sub-models whereas the model is globally optimized, thanks to a special partitioning of the input space in the hidden layer performed to carry out those objectives. The advantage of the methodology proposed is that due to those properties, the global and the local models are both directly optimized. A learning methodology adapted to the proposed model is used in the simulations, consisting of a clustering algorithm for the initialization of the centers and a local search technique.

1 Introduction

When modelling input/output data, it is intended that the obtained system faithfully represents the real phenomenon that is being addressed, that is, that the system is accurate, and at the same time that it expresses its behavior in a comprehensible way, that is, that the system is interpretable. Neuro-fuzzy systems then have to combine these two contradictory requirements to the obtained model, the first more inherent to neural networks, and the second more inherent to fuzzy systems [1] [2]. These systems can approximate a nonlinear mapping using a relatively low number of neurons, that we may call sub-models. Each one of these sub-models, that constitutes the neuro-fuzzy system, performs the approximation of a part of the input space and jointly collaborates to globally approximate the subjacent model from this set of I/O data.

Radial Basis Function (RBF) networks [3] [4] are a widely used neuro-fuzzy paradigm, whose main characteristic is their ability to provide a high accurate model. An additional advantage of this type of models is that there's the possibility to perform a rule extraction procedure from the RBF network, transforming it into a Takagi-Sugeno-Kang (TSK) set of interpretable fuzzy rules that can keep the accuracy of the original system [5], [6].

This work will consider RBF networks using linear weights, so-called RBF networks with regression weights [7]. This type of networks present the advantage that the flexibility of the local models is increased, leading to networks with a much lower number of neurons [8]. The output of a RBF network with regression weights can be expressed as:

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 49–56, 2009.
© Springer-Verlag Berlin Heidelberg 2009

$$F(\boldsymbol{x}) = \frac{\sum\limits_{k=1}^{K} \mu^k(\boldsymbol{x}) Y^k(\boldsymbol{x})}{\sum\limits_{k=1}^{K} \mu^k(\boldsymbol{x})} \tag{1}$$

where K is the number of neurons in the hidden layer, $Y^k(\boldsymbol{x}) = a^k + \sum\limits_{i=1}^{n} b_i^k * x_i$ are their linear output weights of the neurons, and the $\mu^k(\boldsymbol{x}) = \prod_{i=1}^{n} \mu_i^k(x_i)$ are the activations of the hidden neurons, that when using Gaussian kernels can be expressed as:

$$\mu_i^k(x_i) = e^{-\frac{(x_i - c_i^k)^2}{2{\sigma_i^k}^2}} , \tag{2}$$

being c_i^k and σ_i^k the center and radius of the Gaussian function of neuron k at dimension i.

When dealing with the optimization of RBF networks, global accuracy is usually the single objective to optimize, and the problem of local model optimization is barely addressed [9], [10], [11]. Nevertheless, some works have studied multiobjective optimization formulations of neuro-fuzzy systems, that deal with both local and global modelling [12]. Multi-modelling is another important area of research dealing with local and global optimization [13].

This work presents modified RBF model whose main characteristic is its ability to perform a global model approximation, whereas preserving the interpretation of the local sub-models. These characteristics are possible thanks to an special partitioning of the input space by means of a modified calculation of the final normalized activation of the neurons, which controls their overlapping and thus allows us to perform an unconstrained training. The learning methodology for this model used in the simulations section consists of a clustering algorithm especially suited to function approximation problems and a local search technique.

2 Modified Radial Basis Function Network for Local and Global Modelling

The way to obtain the desired characteristic is adapted from previous approaches for fuzzy systems [14] [15]. The local modelling will be achieved through the use of the Taylor series expansion of a function around a point. The intention is that the polynomial weights of the RBF network represent the Taylor series expansion of the system output around the neuron centres, thus verifying the local modelling properties of the neurons. Two properties will be required in the RBF network with regression weights so that these properties are obtained:

1. The weights have the general form of a centred polynomial around a point which resembles a truncated Taylor series expansion. The Taylor Theorem states that if a continuous function $f(\boldsymbol{x})$ defined in an interval $(c-r, c+r)$ is l times differentiable, it can be approximated around the point $\boldsymbol{x} = \boldsymbol{c}$ as its Taylor series expansion

$$f(\boldsymbol{x}) \simeq f(\boldsymbol{c}) + (\boldsymbol{x} - \boldsymbol{c})^T \left[\nabla(\boldsymbol{c})\right] + \tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{c})^T W (\boldsymbol{x} - \boldsymbol{c}) + \ldots \tag{3}$$

Given the general RBF output function formulation, the linear weights of the RBF network can be easily modified to resemble a truncated form of equation 3 (the constant and linear term in the summation).

2. The input partitioning caused by the Radial Basis Functions in the hidden layer follow a structure that allows such local modelling. This implies a twofold condition. First, the model output should be continuous and differentiable, so that the Taylor Theorem can be applied. Second, the overlapping degree of all the RBF nodes activations should vanish at each neuron centre. This way, every point of the n-dimensional space $\boldsymbol{c}^k$ identified by the centre of a neuron k, is only affected by its respective Radial Basis Function in the global system output function. Moreover, the linear weights $Y^k(\boldsymbol{x})$ will contain the information on the function $f(\boldsymbol{x})$ in $\boldsymbol{c}^k$, as well as information about its behavior around that point (its derivatives).

In [14] a special membership function type was presented for grid-partitioned fuzzy systems, that carries out those conditions about the overlapping of the input space partitioning and the continuity and differentiability of the global model output. However, the overlapping degree among the different local models in the neurons can not nevertheless be directly controlled through the use of a specific radial basis function type. The activation functions are not organized along each dimension separately, but rather they are "randomly" distributed (scattered) according to the distribution of the neuron centres along the n-dimensional input space.

Traditionally the output of a normalized RBF network is obtained through weighted average aggregation of the hidden neuron outputs, in which the final normalized activation of a neuron depends on the relative activation of the rest of the neurons as seen in equation 1. The proposal in this work is to modify this final neuron activation, so that it is adapted to the overlapping and continuity requirements imposed to the RBF network. For the sake of simplicity, we next first explain how to perform this modification in the process for a one-dimensional case. Later it will be extrapolated to the general case.

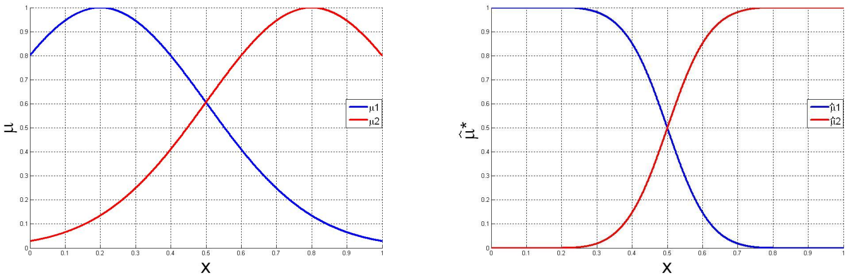
Let us assume the simple case of a one-dimensional space with domain [0,1] with two gaussian neurons centred for example in $c^1 = 0.2$ and $c^2 = 0.8$ with $\sigma = 0.3$ (see figure 1.(a)). In this case, there is a moderated overlapping between the two neurons' activations. In order to comply with the afore-mentioned overlapping conditions we will allow the domain of the first neuron activation $\mu^1(x)$ to be limited by the function $1-\mu^2(x)$. That is, when the activation value of the opposite neuron is 1, the first neuron activation will be forced to take the value 0. That is, the final activation value for any point in the network neurons using normalization would be

$$\mu^{1*}(x) = \mu^1(x)\left(1-\mu^2(x)\right), \hat{\mu}^{1*}(x) = \frac{\mu^{1*}(x)}{\mu^{1*}(x)+\mu^{2*}(x)} \tag{4}$$

$$\mu^{2*}(x) = \mu^2(x)\left(1-\mu^1(x)\right), \hat{\mu}^{2*}(x) = \frac{\mu^{2*}(x)}{\mu^{1*}(x)+\mu^{2*}(x)} \tag{5}$$

Generalizing to the n-dimensional case, the activation value of the k-th neuron is obtained by the following equation

$$\mu^{k*}(\boldsymbol{x}) = \mu^k(\boldsymbol{x}) \prod_{\substack{j=1;\\ j\neq k}}^{K} \left(1-\mu^j(\boldsymbol{x})\right) \tag{6}$$



(a) (b)

Fig. 1. a) Original μ^1 and μ^2 MFs for the one-dimensional example. b) Normalized final neurons activations using the modified calculation $\hat{\mu}^{1*}$ and $\hat{\mu}^{2*}$.

Therefore, for any given number of neurons K, the general expression for the RBF network output, using normalization (which forces the activation value of the neurons to sum up to one in every point) can be calculated as follows

$$F(\boldsymbol{x}) = \sum_{k=1}^{K} \hat{\mu}^{k*}(\boldsymbol{x}) Y^k(\boldsymbol{x}) = \frac{\sum\limits_{k=1}^{K} \mu^{k*}(\boldsymbol{x}) Y^k(\boldsymbol{x})}{\sum\limits_{k=1}^{K} \mu^{k*}(\boldsymbol{x})} = \frac{\sum\limits_{k=1}^{K} \left(\mu^k(\boldsymbol{x}) \prod\limits_{\substack{j=1;\\ j\neq k}}^{K} \left(1-\mu^j(\boldsymbol{x})\right) \right) Y^k(\boldsymbol{x})}{\sum\limits_{k=1}^{K} \left(\mu^k(\boldsymbol{x}) \prod\limits_{\substack{j=1;\\ j\neq k}}^{K} (1-\mu^j(\boldsymbol{x})) \right)} \tag{7}$$

where $\hat{\mu}^{k*}(\boldsymbol{x}) = \mu^{k*}(\boldsymbol{x}) / \sum\limits_{j=1}^{K} \mu^{j*}(\boldsymbol{x})$ is the normalized activation value for neuron k.

With this new formulation of the system's output given in equation 7, the final normalized neuron activations are modified so that they consider the relative positioning of each neuron with respect to the others. Moreover, the output function of the proposed model is continuous and differentiable, since it is a linear composition of continuous and differentiable functions.

It is immediate to deduce that the following properties hold, due to the continuity of the gaussian function:

$$\begin{array}{c} \hat{\mu}^{2*}(c^1) = 0; \hat{\mu}^{1*}(c^1) = 1; \Rightarrow F(c^1) = Y^1(c^1) = a^1 \\ \hat{\mu}^{1*}(c^2) = 0; \hat{\mu}^{2*}(c^2) = 1; \Rightarrow F(c^2) = Y^2(c^2) = a^2 \end{array} \tag{8}$$

moreover, due to the differentiability properties of those functions, it holds that

$$\begin{array}{c} \frac{\partial \hat{\mu}^{1*}}{\partial x}(c^1) = 0 \wedge \frac{\partial \hat{\mu}^{2*}}{\partial x}(c^1) = 0 \Rightarrow \frac{\partial F}{\partial x}(c^1) = \frac{\partial Y^1}{\partial x}(c^1) = b^1 \\ \frac{\partial \hat{\mu}^{2*}}{\partial x}(c^2) = 0 \wedge \frac{\partial \hat{\mu}^{1*}}{\partial x}(c^2) = 0 \Rightarrow \frac{\partial F}{\partial x}(c^2) = \frac{\partial Y^2}{\partial x}(c^2) = b^2 \end{array} \tag{9}$$

Note that those properties are hold thanks to the use of the designed partitioning and to the use of truncated Taylor series-shaped polynomial consequents. Thus, according to equations 8 and 9, it holds that the weights $Y^k(x)$ of the neurons, can be interpreted as the first-order truncated Taylor series expansion of the model output around the respective neuron centre. Those results can directly be extrapolated to the n-dimensional case, thanks to the continuity and differentiability of the compounded functions.

3 Learning Methodology

A typical learning procedure for RBF networks can be summarized as follows [16]

- initialization of the neuron centres and radii
- calculation of the optimal weights of every neuron using LSE
- local search minimization to optimize centres and other MF parameters

The learning algorithm we propose is based on this scheme due to the lower computational cost that it entails, and to its effectiveness in finding good solutions. In the simulations, the learning methodology will make use of the ICFA algorithm [16] (Improved Clustering for Function Approximation) for an appropriate initialization of the centres in order to obtain optimal accurate models. This algorithm use the information provided by the output values of the function to approximate, so that it places more centres where the output variability is higher, instead of where there is a higher number of input vectors.

Since the model output function (Eq. 1) is linear with respect to all the neuron weights parameters, given a set of M input/output data $D = (\boldsymbol{x}^1, y^1), ..., (\boldsymbol{x}^m, y^m)$ of a modelling problem, it is possible to optimally obtain these parameters through the use of a wide range of mathematical methods. In this work we will use the Least Square Error (LSE) approach for the optimization of the neuro-fuzzy linear coefficients, and Singular Value Decomposition (SVD) to solve the linear equation system constructed. The objective is to minimize the mean square error function $J = \sum_{m \in D} \left(F\left(\boldsymbol{x}^m\right) - y^m\right)^2$.

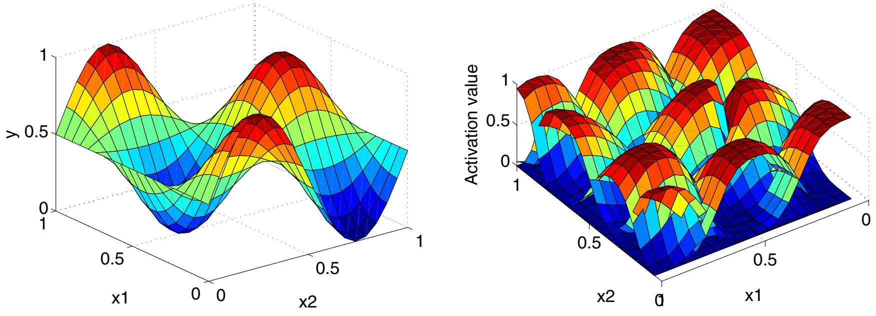
Once an initial configuration of the centres and radii is obtained, thanks to the availability of a direct method to obtain the optimal consequents coefficients and the continuity and differentiability properties of the proposed model with respect to any of its parameters, any local search procedure could be used in order to find an optimal configuration of the parameters defining every weight. Using a local search procedure, both centres and radii can be optimized to obtain a more accurate system according to the error function J. In this work we have optimized the neuron centres using the Levenberg-Marquardt algorithm, estimating for the sake of simplicity the radii according to the minimum distance among them [17].

4 Simulations

This section presents a significant example of the application of the model modified RBF network methodology to function approximation problems. Given the y function that is expressed as (see reference [18])

$$y(x_1, x_2) = 0.5 * (1 + \sin(2 * \pi * x_1) * \cos(2 * \pi * x_2) \qquad (10)$$

400 samples were generated randomly distributed as training dataset. Another set of 400 samples was generated as test set. As an example of execution, 10 centres were



(a) (b)

Fig. 2. a) Original function y. b) Activations of the 10 neurons of the modified RBF network obtained after the complete learning algorithm for the function example y.

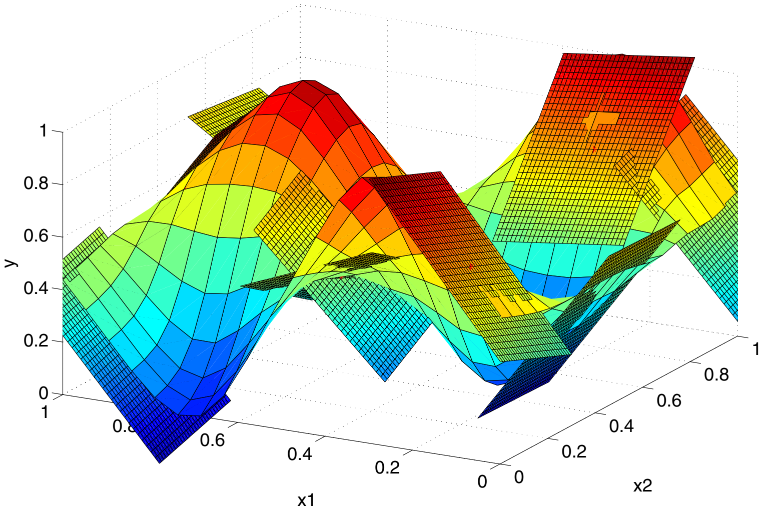


Fig. 3. Joint representation of the model output, together with the linear weights of the 10 network neurons in the area surrounding the corresponding centre

initialized using the ICFA algorithm [16]. The radius of every node of the model was automatically initialized according to the minimum distance among them [17]. Figure 2.(a) shows the shape of the original function y.

The local search procedure readjusts the radial basis functions parameters (centres and radius) according to the performance obtained in the optimization of the linear consequents using LSE, so that the error function J is minimized. The training Normalized Root Mean Square Error (NRMSE [14]) of the modified RBF network is 0.095, whereas the test NRMSE obtained is 0.1.

Figure 2.(b) shows the activations of the 10 neurons of the RBF network, according to the modified final normalized neuron activation designed. This figure shows the distribution of the centres obtained after the execution of the algorithm; it is obtained a distribution of the neurons activations that is intuitive and in which no neuron has effect in the rest of the neurons centres, being each neuron activation limited by the location of the rest of centres in the n-dimensional input space.

The effect of this partitioning in the model is observed in figure 3, which shows the representation of the local and global modelling performed by the proposed modified RBF model having ten neurons. It can be observed that the weights bring the information about the exact function value at the neuron centre, as well as the behavior of the model output function in the area around that neuron centre. Each polynomial defines a plane that indicates the model output slope value around the neuron centre. Moreover, the global model output approximates the subjacent function given in the training data set for this example. Given this modified RBF model, the extraction of the set of rules from the network would be immediate, according to the well-known equivalence of this model with a TSK fuzzy system [4] [19] [14].

5 Conclusions

In this paper we have presented a modified Radial Basis Function model, that due to a special partitioning of the input space provides a simultaneous optimization of the global model and of the local sub-models. In order to show the properties of the proposed modified RBF model, we have used a learning methodology consisting of a clustering algorithm especially suited to function approximation problems and a local search technique. The example performed has shown that it is able to obtain a good global approximation, and moreover the obtained local models provide the information about the behavior of the model in the areas of the input space around the neuron centres, and therefore of the behavior of the subjacent system being approximated. Moreover due to the equivalency between TSK fuzzy models and RBF networks, there is the possibility to perform a rule extraction procedure to obtain an equivalent TSK fuzzy system with rules having the same global-local approximation capabilities.

Acknowledgment

This work has been partially supported by the Spanish CICYT Project TIN2007-60587 and Junta Andalucia Project P07-TIC-02768.

References

1. Casillas, J., Cordon, O., del Jesus, M.J., Herrera, F.: Tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. IEEE Trans. Fuzzy Syst. 13(1), 12–29 (2005)
2. Herrera, L., Pomares, H., Rojas, I., Valenzuela, O., Awad, M.: MultiGrid-based fuzzy systems for function approximation. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS, vol. 2972, pp. 252–261. Springer, Heidelberg (2004)

3. Haykin, S.: Neural Networks. Prentice-Hall, Englewood Cliffs (1998)
4. Jang, J.S.R., Sun, C.T.: Functional equivalence between radial basis function networks and fuzzy inference systems. IEEE Transactions on Neural Networks 4(1), 156–159 (1993)
5. Jin, Y., Sendhoff, B.: Extracting interpretable fuzzy rules from rbf networks. Neural Processing Letters 17, 149–164 (2003)
6. Vachkov, G.L., Kiyota, Y., Komatsu, K.: Learning of rbf network models for prediction of unmeasured parameters by use of rules extraction algorithm. In: Proc. of NAFIPS 2005 (2005)
7. Langari, R., Wang, L., Yen, J.: R.b.f. networks, regression weights, and the expectation-maximization algorithm. IEEE T. Syst., Man, and Cyber. A 27, 613–623 (1997)
8. Rojas, I., Pomares, H., Bernier, J., Ortega, J., Pino, B., Pelayo, F., Prieto, A.: Time series analysis using normalized pg-rbf network with regression weights. NeuroComputing 42, 267–285 (2002)
9. Leng, G., McGinnity, T.M., Prasad, G.: Design for self-organizing fuzzy neural networks based on genetic algorithms. IEEE T. Fuzzy Systems 14(6), 755–766 (2006)
10. Liu, P., Li, H.: Efficient learning algorithms for three-layer regular feedforward fuzzy neural networks. IEEE T. Neural Networks 15(3), 545–558 (2004)
11. González, J., Rojas, I., Ortega, J., Pomares, H., Fernandez, F., Diaz, A.: Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. IEEE T. Neural Networks 14(6), 1478–1495 (2003)
12. Johansen, T., Babuska, R.: Multiobjective identification of takagi-sugeno fuzzy models. IEEE Trans. Fuz. Syst. 11(6), 847–860 (2003)
13. Madani, K., Thiaw, L.: Self-organizing multi-modeling: A different way to design intelligent predictors. Neurocomput 70(16-18), 2836–2852 (2007)
14. Herrera, L., Pomares, H., Rojas, I., Valenzuela, O., Prieto, A.: Tase, a taylor series based fuzzy system model that combines interpretability and accuracy. Fuzzy Sets and Systems 153, 403–427 (2005)
15. Herrera, L., Pomares, H., Rojas, I., Guillén, A., Awad, M., González, J.: Interpretable rule extraction and function approximation from numerical input/output data using the modified fuzzy tsk model: Tase model. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS, vol. 3641, pp. 402–411. Springer, Heidelberg (2005)
16. Guillén, A., González, J., Rojas, I., Pomares, H., Herrera, L.J., Valenzuela, O., Prieto, A.: Using fuzzy logic to improve a clustering technique for function approximation. Neurocomputing 70(16-18), 2853–2860 (2007)
17. Moody, J., Darken, C.: Fast learning in networks of locally-tuned processing units. Neural Computation 1(2), 281–294 (1989)
18. Rovatti, R., Guerrieri, R.: Fuzzy sets of rules for system identification. IEEE Trans. Fuzzy Syst. 4(2), 89–102 (1996)
19. Reyneri, L.M.: Unification of neural and wavelet networks and fuzzy systems. IEEE Transactions on Neural Networks 10(4), 801–814 (1999)

A Preliminar Analysis of CO^{2}RBFN in Imbalanced Problems

M.D. Pérez-Godoy[1], A.J. Rivera[1], A. Fernández[2], M.J. del Jesus[1],
and F. Herrera[2]

[1] Department of Computer Science, University of Jaén,
Campus Las Lagunillas, 23071 Jaén, Spain
[2] Department of Computer Science and Artificial Intelligence,
University of Granada, 18071 Granada, Spain

Abstract. In many real classification problems the data are imbalanced, i.e., the number of instances for some classes are much higher than that of the other classes. Solving a classification task using such an imbalanced data-set is difficult due to the bias of the training towards the majority classes. The aim of this contribution is to analyse the performance of CO^{2}RBFN, a cooperative-competitive evolutionary model for the design of RBFNs applied to classification problems on imbalanced domains and to study the cooperation of a well known preprocessing method, the "Synthetic Minority Over-sampling Technique" (SMOTE) with our algorithm. The good performance of CO^{2}RBFN is shown through an experimental study carried out over a large collection of imbalanced data-sets.

Keywords: Neural Networks, Radial Basis Functions, Genetic Algorithms, Imbalanced Data-Sets.

1 Introduction

Radial Basis Function Networks (RBFNs) are one of the most important Artificial Neural Network (ANN) paradigms in the machine design field. An RBFN is a feed-forward ANN with a single layer of hidden units, called radial basis functions (RBFs) [5,19]. The overall efficiency of RBFNs has been proved in many areas such as pattern classification [6], function approximation [24], time series prediction [28] and multiple specific applications such as credit assessment [16], process control [14], medical diagnosis [17], and time series forecasting [27] among others. In most of these areas, the data-sets have a common and very usual characteristic: they are imbalanced data-sets [7].

The imbalance data-set problem occurs when the number of instances of one class overwhelms the others. To solve it there are two main types of solutions: solutions at the data level which is achieved balancing the class distribution and solutions at the algorithmic level, for example adjusting the cost per class.

An important paradigm for RBFN design is the Evolutionary Computation [12]. There are different proposals in this area with different scheme representations: Pittsburgh [13], where each individual is a whole RBFN, and cooperative-competitive [28], where an individual represents an RBF.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 57–64, 2009.
© Springer-Verlag Berlin Heidelberg 2009

In this work we study the performance of CO^{2}RBFN [25] in the framework of imbalanced data-sets and analyse the cooperation of a well known preprocessing method, the “Synthetic Minority Over-sampling Technique” (SMOTE) [8] with our algorithm. To do so, in Section 2 a brief description of the imbalanced data-sets in classification and the solutions provided for them in the specialized bibliography is shown. The cooperative-competitive evolutionary model for the design of RBFNs applied to classification problems, CO^{2}RBNF, is described in Section 3. The analysis of the experiments and the conclusions are shown in Sections 4 and 5.

2 Imbalanced Data-Sets in Classification

The problem of imbalanced data-sets in classification [7] occurs when the number of instances of one class is much lower than the instances of the other classes. This problem is implicit in most real world applications including, but not limited to telecommunications, finance, biology and medicine.

In this work, we focus on binary imbalanced data-sets, where there is only one positive and one negative class. Furthermore, we use the imbalance ratio (IR) [22], defined as the ratio of the number of instances of the majority class and the minority class, to organize the different data-sets.

In classification, the class imbalance problem will cause a bias on the training of classifiers and will result in the lower sensitivity of detecting the minority class examples. For this reason, a large number of approaches have been previously proposed to deal with it. These approaches can be categorized into two groups: the internal approaches that create new algorithms or modify existing ones to take the class imbalance problem into consideration [3,30] and external approaches that preprocess the data in order to diminish the effect caused by their class imbalance [4,10].

For RBFNs there are some proposals for the use of a cost function in the training process to compensate imbalance class and strategies to reduce the impact of the cost function in the data probability distribution [1]. With respect to external approaches, different studies have been done to assess the effect of the size of the training data-set on the accuracy [2]. In [20], the authors study three different ANN architectures, multilayered back propagation, RBFNs and Fuzzy ARTMAP, with the use of random oversampling and the snowball training methods. In [23], the over-sampling method SMOTE [8], has been used with RBFNs showing a good behaviour. According to this, we employ in this contribution the SMOTE algorithm in order to deal with the problem of imbalanced data-sets.

The SMOTE method (detailed in [8]) tries to form new minority class examples by interpolating between several minority class examples that lie together. Thus, the overfitting problem is avoided and causes the decision boundaries for the minority class to spread further into the majority class space.

With respect to the evaluation in imbalanced domains the most used empirical measure, accuracy, does not distinguish between the number of correct labels of different classes, which in the framework of imbalanced problems may lead to erroneous conclusions. The metric used in this work is the geometric mean of the true rates [3], defined as

$$GM = \sqrt{\frac{TP}{TP + FN}\frac{T\dot{N}}{FP + TN}} \quad (1)$$

where TP,TN,FP and FN stand for True Positives, True Negatives, False Positives and False Negatives respectively. This metric attempts to maximize the accuracy of each one of the two classes with a good balance.

3 CO^{2}RBFN: An Evolutionary Cooperative-Competitive Hybrid Algorithm for RBFN Design

CO^{2}RBFN [25], is an evolutionary cooperative-competitive hybrid algorithm for the design of RBFNs. In this algorithm each individual of the population represents, with a real representation, an RBF and the entire population is responsible for the final solution. The individuals cooperate towards a definitive solution, but they must also compete for survival. In this environment, in which the solution depends on the behaviour of many components, the fitness of each individual is known as credit assignment. In order to measure the credit assignment of an individual, three factors have been proposed: the RBF contribution to the network output, the error in the basis function radius, and the degree of overlapping among RBFs.

The application of the operators is determined by a Fuzzy Rule-Based System. The inputs of this system are the three parameters used for credit assignment and the outputs are the operators' application probability.

The main steps of CO^{2}RBFN, explained in the following subsections, are shown in the pseudocode in Figure 1.

```
1. Initialize RBFN
2. Train RBFN
3. Evaluate RBFs
4. Apply operators to RBFs
5. Substitute the eliminated RBFs
6. Select the best RBFs
7. If the stop condition is not
     verified go to step 2
```

Fig. 1. Main steps of CO^{2}RBFN

RBFN initialization. To define the initial network a specified number m of neurons (i.e. the size of population) is randomly allocated among the different patterns of the training set. To do so, each RBF centre, $\boldsymbol{c}_i$, is randomly established to a pattern of the training set. The RBF widths, d_i, will be set to half the average distance between the centres. Finally, the RBF weights, w_{ij}, are set to zero.

RBFN training. The Least Mean Square algorithm [29] has been used to calculate the RBF weights. This technique exploits the local information that can be obtained from the behaviour of the RBFs.

RBF evaluation. A credit assignment mechanism is required in order to evaluate the role of each RBF ϕ_i in the cooperative-competitive environment. For an RBF, three parameters, a_i ,e_i ,o_i are defined:

- The contribution, a_i, of the RBF ϕ_i, $i = 1 \ldots m$, is determined by considering the weight, w_i, and the number of patterns of the training set inside its width, pi_i. An RBF with a low weight and few patterns inside its width will have a low contribution:

$$a_i = \begin{cases} |w_i| & if \quad pi_i > q \\ |w_i| * (pi_i/q) & otherwise \end{cases} \tag{2}$$

  where q is the average of the pi_i values minus the standard deviation of the pi_i values.
- The error measure, e_i, for each RBF ϕ_i, is obtained by counting the wrongly classified patterns inside its radius:

$$e_i = \frac{pibc_i}{pi_i} \tag{3}$$

  where $pibc_i$ and pi_i are the number of wrongly classified patterns and the number of all patterns inside the RBF width respectively. It must be noted that this error measure does not consider the imbalance among classes.
- The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i. This parameter is computed by taking into account the fitness sharing methodology [12], whose aim is to maintain the diversity in the population. This factor is expressed as:

$$o_i = \sum_{j=1}^{m} o_{ij} \tag{4}$$

  where o_{ij} measures the overlapping of the RBF ϕ_i y ϕ_j $j = 1 \ldots m$.

Applying operators to RBFs. In CO^{2}RBFN four operators have been defined in order to be applied to the RBFs:

- Operator Remove: eliminates an RBF.
- Operator Random Mutation: modifies the centre and width of an RBF in a percentage below 50% of the old width.
- Operator Biased Mutation: modifies the width and all coordinates of the centre using local information of the RBF environment. The centre is varied in order to approximate it to the average of the patterns belonging to the

Table 1. Fuzzy rule base representing expert knowledge in the design of RBFNs

	Antecedents			Consequents					Antecedents			Consequents			
	v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}		v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}
R1	L			M-H	M-H	L	L	R6		H		M-H	M-H	L	L
R2	M			M-L	M-H	M-L	M-L	R7			L	L	M-H	M-H	M-H
R3	H			L	M-H	M-H	M-H	R8			M	M-L	M-H	M-L	M-L
R4		L		L	M-H	M-H	M-H	R9			H	M-H	M-H	L	L
R5		M		M-L	M-H	M-L	M-L								

RBF class and inside its RBF width. The objective of the width training is that most of the patterns belonging to the RBF class will be inside the RBF width.
- Operator Null: in this case all the parameters of the RBF are maintained.

The operators are applied to the whole population of RBFs. The probability for choosing an operator is determined by means of a Mandani-type fuzzy rule based system [18] which represents expert knowledge about the operator application in order to obtain a simple and accurate RBFN. The inputs of this system are parameters a_i, e_i and o_i used for defining the credit assignment of the RBF ϕ_i. These inputs are considered as linguistic variables va_i, ve_i and vo_i. The outputs, p_{remove}, p_{rm}, p_{bm} and p_{null}, represent the probability of applying Remove, Random Mutation, Biased Mutation and Null operators, respectively.

Table 1 shows the rule base used to relate the described antecedents and consequents. In the table each row represents one rule. For example, the interpretation of the first rule is: If the contribution of an RBF is Low Then the probability of applying the operator Remove is Medium-High, the probability of applying the operator Random Mutation is Medium-High, the probability of applying the operator Biased Mutation is Low and the probability of applying the operator null is Low.

Introduction of new RBFs. In this step, the eliminated RBFs are substituted by new RBFs. The new RBF is located in the centre of the area with maximum error or in a randomly chosen pattern with a probability of 0.5 respectively.

The width of the new RBF will be set to the average of the RBFs in the population plus half of the minimum distance to the nearest RBF. Its weights are set to zero.

Replacement strategy. The replacement scheme determines which new RBFs (obtained before the mutation) will be included in the new population. To do so, the role of the mutated RBF in the net is compared with the original one to determine the RBF with the best behaviour in order to include it in the population.

4 Experimentation and Results

In this study our algorithm is applied to eight data-sets with two different degrees of imbalance and compared with four different soft-computing methods [11],

Table 2. Experimentation Results

				CO^{2}RBFN	**C4.5**	**Chi**	**Ishibuchi05**
IR	**data-set**	**CO^{2}RBFN**	**E-Algorithm**	**+SMOTE**	**+SMOTE**	**+SMOTE**	**+SMOTE**
3.19	Glass0123vs456	92.27 ± 3.27	82.09 ± 6.96	**93.78 ± 3.28**	90.13 ± 3.17	85.83 ± 3.04	88.56 ± 5.18
4.92	New-thyroid2	98.40 ± 3.72	88.57 ± 3.82	**98.46 ± 2.22**	96.51 ± 4.87	89.81 ± 10.77	94.21 ± 4.23
5.14	New-thyrid1	**98.02 ± 3.05**	88.52 ± 8.79	98.01 ±2.79	97.98 ± 3.79	87.44 ± 8.11	89.02 ± 13.52
5.46	Ecoli2	92.02 ± 3.40	70.35 ± 15.36	**93.14 ± 4.50**	91.60 ± 4.86	88.01 ± 5.45	87.00 ± 4.43
16.68	Abalone9vs18	**75.70 ± 9.52**	32.29 ± 20.61	75.34 ± 10.34	53.19 ± 8.25	63.93 ± 11.00	65.78 ± 9.23
23.10	Yeast2vs8	71.88 ± 14.13	72.83 ± 14.97	77.31 ± 12.23	**78.23 ± 13.05**	72.75 ± 14.99	72.83 ± 14.97
32.78	Yeast5	94.12 ± 4.40	88.17 ± 7.04	94.03 ± 4.58	92.04 ± 4.99	93.41 ± 5.35	**94.94 ± 0.38**
128.87	Abalone19	50.12 ± 21.81	0.00 ± 0.00	**70.18 ± 11.77**	15.58 ± 21.36	62.96 ± 8.27	66.09 ± 9.40

where SMOTE method has been used to pre-process the datasets. In order to estimate the precision we use five fold cross validation approach.

Table 2, columns 3 and 5 show the GM measure obtained with CO^{2}RBFN applied to data-sets with and without SMOTE pre-processing respectively. In column 4, the GM results for the E-algorithm [30] are described (which does not use pre-processed data due to it does not need to change the original class distribution). Columns 6, 7 and 8 describes the GM results for C4.5 [26], Chi [9] and Ishibuchi [15] with the data-sets preprocessed by SMOTE. The analysis of the results shows that when CO^{2}RBFN is applied to the data-sets with an IR lower than 20, there is no need for preprocessing since our algorithm outperforms all the other considered data mining methods, i.e., E-algorithm, C4.5 + SMOTE, Chi + SMOTE and Ishibuchi+SMOTE.

Regarding CO^{2}RBFN in data-sets with the higher degree of imbalance, we observe the necessity to preprocess the data, as the results are improved considering the application of SMOTE. Furthermore, CO^{2}RBFN is the method with the best GM results for five of the eight data-sets. For the other data-sets, CO^{2}RBFN + SMOTE is the second best choice.

In summary, CO^{2}RBFN presents a good behavior with the imbalanced data-sets, even when it is applied to data-sets without pre-processing method.

The good performance of CO^{2}RBFN in classification imbalanced tasks, is due to the definition proposed for the credit assignment of individuals. In this way and in order to evaluate an individual only one of the three parameters takes into account the accuracy of the RBF. The other ones are dedicated for distributing, exploring and optimizing the RBFs in the dataset space definition.

5 Concluding Remarks

In this contribution, we analyze CO^{2}RBFN, a hybrid evolutionary cooperative-competitive algorithm for the RBFN design applied to the classification of imbalanced data-sets with medium and high IR.

We study the effect of a preprocessing stage in the performance of CO^{2}RBFN by contrasting the results obtained using the original data-sets against the ones obtained with the SMOTE algorithm. Furthermore, we include in our experimental study some well-known data mining methods for comparison.

CO^{2}RBFN (without pre-processing) has a robust performance with imbalanced data-sets with IR less than 20, obtaining the best GM results. The performance of CO^{2}RBFN + SMOTE is suitable with a higher IR.

As future work, we will focus our studies in the framework of high imbalanced data-sets, analysing the improvement of CO^{2}RBFN considering a precision factor in the credit assignment.

Acknowledgment

This work had been supported by the Spanish Ministry of Science and Technology under Projects TIN2008-06681-C06-01 and TIC-3928.

References

1. Alejo, R., García, V., Sotoca, J.M., Mollineda, R.A., Sánchez, J.S.: Improving the performance of the RBF neural networks trained with imbalanced samples. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 162–169. Springer, Heidelberg (2007)
2. Al-Haddad, L., Morris, C.W., Boddy, L.: Training radial basis function neural networks: effects of training set size and imbalanced training sets. Journal of Microbiological Methods 43, 33–44 (2000)
3. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition 36(3), 849–851 (2003)
4. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behaviour of several methods for balancing machine learning training data. SIGKDD Explorations 6(1), 20–29 (2004)
5. Broomhead, D., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex System 2, 321–355 (1988)
6. Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. IEEE T. Syst. Man Cy. B 35(5), 928–947 (2005)
7. Chawla, N.V., Japkowicz, N., Kolcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations 6(1), 1–6 (2004)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligent Research 16, 321–357 (2002)
9. Chi, Z., Yan, H., Pham, T.: Fuzzy algorithms with applications to image processing and pattern recognition. World Scientific, Singapore (1996)
10. Estabrooks, A., Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data-sets. Computational Intelligence 20(1), 18–36 (2004)
11. Fernández, A., del Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification system with genetic rule selection for imbalanced data-set. International Journal of Approximate Reasoning (2009) doi:10.1016/j.ijar.2008.11.004
12. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
13. Harpham, C., Dawson, C.W., Brown, M.R.: A review of genetic algorithms applied to training radial basis function networks. Neural Computing and Applications 13, 193–201 (2004)

14. Huang, S.N., Tan, K.K., Lee, T.H.: Adaptive neural network algorithm for control design of rigid-link electrically driven robots. Neurocomputing 71(4-6), 885–894 (2008)
15. Ishibuchi, H., Yamamoto, T.: Rule weight specification in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 13, 428–435 (2005)
16. Lacerda, E., Carvalho, A., Braga, A., Ludermir, T.: Evolutionary Radial Functions for Credit Assessment. Appl. Intell. 22, 167–181 (2005)
17. Maglogiannis, I., Sarimveis, H., Kiranoudis, C.T., Chatziioannou, A.A., Oikonomou, N., Aidinis, V.: Radial basis function neural netwroks classification for the recognition of idiopathic pulmonary fibrosis in microscopic images. IEEE T. Inf. Technol B 12(1), 42–54 (2008)
18. Mandani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man Mach. Stud. 7(1), 1–13 (1975)
19. Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. Neural Comput. 1, 281–294 (1989)
20. Murhphey, Y.L., Guo, H.: Neural learning from umbalanced data. Applied Intelligence 21, 117–128 (2004)
21. Orr, M.J.L.: Regularization on the selection of radial basis function centers. Neural Comput. 7, 606–623 (1995)
22. Orriols-Puig, A., Bernadó-Mansilla, E.: Evolutionary rule-based systems for imbalanced data-sets. Soft Computing 13(3), 213–225 (2009)
23. Padmaja, T.M., Dhulipalla, N., Krishna, P.R., Bapi, R.S., Laha, A.: An unbalanced data classification model using hybrid sampling technique for fraud detection. In: Mueller, M.S., Chapman, B.M., de Supinski, B.R., Malony, A.D., Voss, M. (eds.) IWOMP 2005 and IWOMP 2006. LNCS, vol. 4315, pp. 341–438. Springer, Heidelberg (2008)
24. Park, J., Sandberg, I.: Universal approximation using radial-basis function networks. Neural Comput. 3, 246–257 (1991)
25. Pérez-Godoy, M., Rivera, A.J., del Jesus, M.J., Berlanga, F.J.: Utilización de un sistema basado en reglas difusas para la aplicacin de operadores en un algoritmo cooperativo-competitivo. In: ESTYLF 2008, pp. 689–694 (2008)
26. Quilan, J.R.: C4.5: Programs for Machine Learning. Morgan Kauffman Publishers, San Mateo (1993)
27. Sun, Y.F., Liang, Y.C., Zhang, W.L., Lee, H.P., Lin, W.Z., Cao, L.J.: Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting. Neural Comput. Appl. 143(1), 36–44 (2005)
28. Whitehead, B., Choate, T.: Cooperative-competitive genetic evolution of Radial Basis Function centers and widths for time series predictiont. IEEE Trans. on Neural Networks 7(4), 869–880 (1996)
29. Widrow, B., Lehr, M.A.: 30 Years of adaptive neural networks: perceptron, madaline and backpropagation. Proceedings of the IEEE 78(9), 1415–1444 (1990)
30. Xu, L., Chow, M.Y., Taylor, L.S.: Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-algorithm. IEEE Transactions on Power Systems 22(1), 164–171 (2007)

Feature Selection in Survival Least Squares Support Vector Machines with Maximal Variation Constraints

V. Van Belle*, K. Pelckmans, J.A.K. Suykens, and S. Van Huffel

Katholieke Universiteit Leuven, ESAT-SCD
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
{`vanya.vanbelle,kristiaan.pelckmans,johan.suykens,`
`sabine.vanhuffel`}`@esat.kuleuven.be`

Abstract. This work proposes the use of maximal variation analysis for feature selection within least squares support vector machines for survival analysis. Instead of selecting a subset of variables with forward or backward feature selection procedures, we modify the loss function in such a way that the maximal variation for each covariate is minimized, resulting in models which have sparse dependence on the features. Experiments on artificial data illustrate the ability of the maximal variation method to recover relevant variables from the given ones. A real life study concentrates on a breast cancer dataset containing clinical variables. The results indicate a better performance for the proposed method compared to Cox regression with an L_1 regularization scheme.

Keywords: failure time data, feature selection, LS-SVM.

1 Introduction

Survival analysis studies the time until a certain event is observed. A typical problem within these studies is the presence of censored data, or data for which the exact failure time is not observed exactly. The most common censoring scheme is right censoring. In this case the event time is known to be later than the last time for which information is available. A second type of censoring is left censoring, occurring in cases where it is known that the failure occurred before a certain time. Another type of censoring is interval censoring, a combination of left and right censoring. This type is often seen in clinical studies in which patients are scheduled for regular check-ups.

An important goal within the analysis of survival data is the construction of prognostic indices. A prognostic index is a scoring function indicating the risk of each patient (in medical studies) or component (in electrical studies). The construction of a prognostic index is associated with a financial cost and effort depending on the number of covariates used within the index and the difficulty of obtaining that covariate. By selecting the most relevant variables, this cost can be

---

* Research supported by GOA-AMBioRICS, CoE EF/05/006, FWO G.0407.02 and G.0302.07, IWT, IUAP P6/04, eTUMOUR (FP6-2002-LIFESCIHEALTH 503094).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 65–72, 2009.
© Springer-Verlag Berlin Heidelberg 2009

reduced. Due to this fact and the increasing availability of high dimensional data, as there are genomics and proteomics data [1], feature selection becomes more and more an issue in current data analysis. Different feature selection procedures have been proposed [2,3,4,5], and can be categorized as filter or wrapper methods [6]. Filter methods select variables independent to the predictor, whereas wrapper methods are methods for which the selection procedure is related to the predictor performance. Most feature selection methods are based on forward selection or backward elimination processes. Forward selection starts with the most relevant feature and adds the best feature in each step. The subset of features is no longer extended once the addition of an extra feature does not improve the performance. In backward selection the search is started with the largest set of variables and the least promising one is eliminated in each step. Disadvantages of these methods are the need to train the model for each subset of selected variables and their large variability: small changes in the data can result in the selection of different subsets of variables. An alternative approach to reveal the most relevant feature is found in LASSO [2], which shrinks some coefficients and sets others to 0. To obtain zeros for certain coefficients the sum of the absolute values of the coefficients is constrained to be less than a certain constant.

In this paper we select variables as proposed in [7,8] and investigate it for a model in survival analysis. Instead of constraining the sum of the coefficients, the maximal variation of each component is minimized. The reason for this is that non-relevant variables will result in very small maximal variations, whereas the variation for relevant variables will remain large. The goal of this paper is to combine the above feature selection procedure with least squares support vector machines (LS-SVM) [9,10] designed for survival analysis [11], resulting in a convex quadratical optimization (QP) problem.

This work is organized as follows. In Section 2 we summarize the principle of survival LS-SVM. Afterwards this model is adapted towards feature selection. Section 3 describes results obtained on artificial and clinical datasets. Section 4 concludes this work.

2 Additive Survival Least Squares LS-SVM

In previous work we presented a Least Squares Support Vector Machine (LS-SVM) as a flexible, non-linear, kernel based model for survival analysis [11], which builds further on the work presented in [12,13]. The model is built from the idea that in practice one is interested in the ranking of samples according to their risk on experiencing the event. The final goal is to create a model for which the predicted risk $u(x)$ correlates with the observed failure times t. To measure the concordance between the estimated risk and the observed failure time, the concordance index (c-index) [14] is used:

$$CI(u) = \frac{\sum_{i=1}^{n_t} \sum_{j \neq i} \varDelta(x_i, t_i, \delta_i; x_j, t_j, \delta_j) I[(u(x_j) - u(x_i))(t_j - t_i) > 0]}{\sum_{i=1}^{n_t} \sum_{j \neq i} \varDelta(x_i, t_i, \delta_i; x_j, t_j, \delta_j)} , \quad (1)$$

where n_t is the number of test points, x_i, t_i and δ_i are the covariate vector, failure time and censoring indicator (1 for an observed event and zero for right censored data) of sample i and $I[z] = 1$ if $z > 0$, and zero otherwise. $\Delta(x_i, t_i, \delta_i; x_j, t_j, \delta_j)$ indicates whether the observations i and j are comparable and depends on the censoring mechanism present in the data. Without censoring $\Delta(x_i, t_i, \delta_i; x_j, t_j, \delta_j)$ equals 1. For right censoring $\Delta(x_i, t_i, \delta_i; x_j, t_j, \delta_j)$ equals zero if $t_i < t_j$ & $\delta_i = 1$ or $t_j < t_i$ & $\delta_j = 1$, and zero otherwise.

2.1 Pairwise Kernel Machine

A possible approach for maximizing the c-index with regard to $u(x)$ is as follows [11]:

$$\begin{aligned} &\boxed{\text{P1}} \ \min_{w,\xi_{ij}} \ \tfrac{1}{2} w^T w + \tfrac{1}{2}\gamma \sum_{i=1}^{n}\sum_{j=1}^{n} \Delta_{ij}\xi_{ij}^2 \\ &\text{s.t. } \ w^T\varphi(x_j) - w^T\varphi(x_i) = 1 + \xi_{ij}, \ \forall\, i,j = \ 1,\ldots,n\,, \end{aligned} \tag{2}$$

where $x_i \in \mathbb{R}^d$ represents the feature vector for the ith datapoint, $w \in \mathbb{R}^{n_\varphi}$ is an unknown vector for the model $u(x) = w^T\varphi(x)$, with $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{n_\varphi}$. n is the number of datapoints, $\gamma > 0$ a regularization constant, ξ_{ij} slack variables allowing for incorrect rankings and $\Delta_{ij} = \Delta(x_i, t_i, \delta_i; x_j, t_j, \delta_j)$. Here it is assumed that $t_j > t_i$ for $j > i$ and the 1 at the right hand side is used as a target value. To visualize the effect of one covariate on the estimated risk, it can be shown how risk changes when varying a single covariate while keeping the rest constant. However, this only gives information on the effect of the covariate on the ranking of the failure time and not on the failure time itself. To obtain the latter information, an extra constraint is added and a componentwise kernel is used [11]:

$$\begin{aligned} &\boxed{\text{P2}} \ \min_{w,\xi,b,\chi} \ \tfrac{1}{2}\textstyle\sum_{p=1}^{d} w_p^T w_p + \tfrac{1}{2}\gamma \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} \Delta_{ij}\xi_{ij}^2 + \frac{1}{2}\mu \sum_{j=1}^{n} \chi_j^2 \\ &\text{s.t. } \begin{cases} \sum_{p=1}^{d} w_p^T\varphi_p(x_{j(p)}) - \sum_{p=1}^{d} w_p^T\varphi_p(x_{i(p)}) = 1 + \xi_{ij}, \forall\, i,j = 1,\ldots,n \\ \delta_j t_j = \delta_j(\sum_{p=1}^{d} w_p^T\varphi_p(x_{j(p)}) + b) - \chi_j, \qquad \forall\, j = 1,\ldots,n\,, \end{cases} \end{aligned} \tag{3}$$

where $x_{i(p)}$ represents the pth covariate of the ith datapoint, d is the number of covariates, $w_p \in R^{n_{\varphi_p}}$ represents the unknown vector of the pth covariate in the model and $\varphi_p(x_{(p)})(\cdot) : \mathbb{R} \to \mathbb{R}^{n_{\varphi_p}}$ represents the feature map corresponding to the pth covariate $x_{(p)}$. $\gamma > 0$ and $\mu > 0$ are two regularization constants and ξ_{ij} and χ_j are slack variables allowing for incorrect rankings and regression errors, respectively.

In [11] we proposed to use the above method to estimate functional forms of covariates in a univariate setting and to combine these terms linearly in order to obtain an interpretable utility function (Figure 1(a)). However, certain covariates

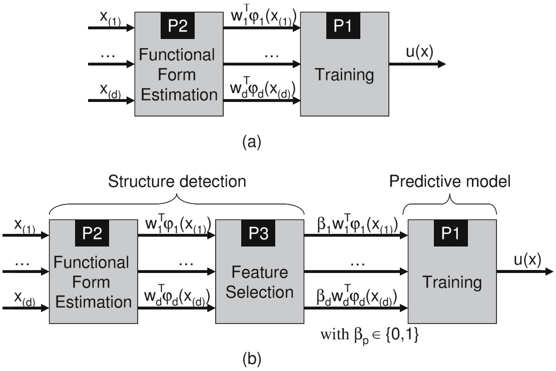


Fig. 1. Illustration of the training phase. **(a)** In previous work we proposed to use model P2 to estimate functional forms of covariates in a first layer and to combine these estimated linearly in a second layer with model P1. **(b)** To select relevant features a third layer is introduced, in which a new model (P3) is used for feature selection. The training of the final prognostic index $u(x)$ is then done only incorporating the selected features.

used in this model, are possibly of little or no importance to the development of an optimal utility function. The model is therefore adapted to incorporate the selection of relevant variables, as illustrated in Figure 1(b).

2.2 Feature Selection

The approach proposed in [7] uses a componentwise model (as we do, see equation (3)) and additionally penalizes large variations for each component. This results in variations which are very low for non-relevant variables and larger for relevant variables. In the methodology followed in our previous work [11] we use model P2 with a polynomial kernel $K(x,z) = (\nu + x^T z)^d, \nu \geq 0$, in a componentwise way, to estimate the functional form for each covariate in a first step (Figure 1). In a second step, model P1 with a linear kernel is used to create the prognostic index as a linear combination of the estimated functional forms. This work concentrates on the adaptation of this second layer, where we not only want to produce a prognostic index, but we want to do so with a small subset of covariates. Therefore the sum of the maximal variations of all components is added to the loss function. Additional constraints indicate the restriction on the componentwise maximal variations. The model formulation then becomes:

$$\boxed{\text{P3}} \quad \min_{w,\xi,m} \ \tfrac{1}{2}\textstyle\sum_{p=1}^{d} w_p^T w_p + \tfrac{1}{2}\gamma \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} \mathit{\Delta}_{ij}\xi_{ij}^2 + \mu \sum_{p=1}^{d} m_p$$

$$\text{s.t.} \quad \begin{cases} \sum_{p=1}^{d} w_p^T \left(\varphi_p(x_j) - \varphi_p(x_i)\right) = 1 + \xi_{ij}, \ \forall\ i,j = 1,\ldots,n \\ -m_p \leq w_p^T \varphi_p(x_{i(p)}) \leq m_p, \qquad\qquad \forall\ i = 1,\ldots,n,\ \forall\ p = 1,\ldots,d, \end{cases} \tag{4}$$

where m_p is the variation of the pth covariate. The prognostic index for a datapoint with covariates x is defined as $u(x) = \sum_{p=1}^{d} w_p^T \varphi_p(x_{(p)})$, where $\varphi_p(x_{(p)}) = x_{(p)}$ for a linear kernel.

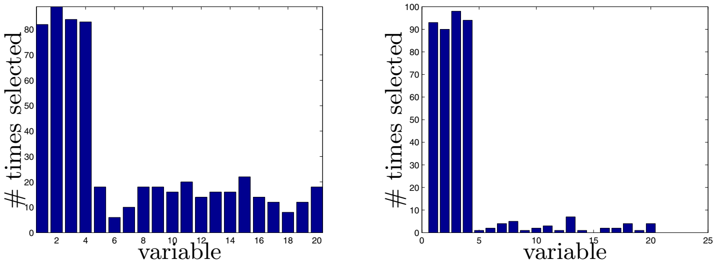
This problem is a quadratic programming problem and can therefore be solved efficiently. In our application, model P3 is used in a linear setting, since all non-linear effects are estimated in the first layer of the model. However, the formulation can be solved in the dual form for applications with other kernels.

3 Results

This section summarizes results on artificial and clinical datasets. The artificial data shows a clear difference in maximal variation for relevant versus non-relevant variables. Applying this approach to a dataset with clinical features [15] on breast cancer results in a better prognostic index then when using L_1 regularization with Cox' proportional hazard model [16]. Model selection was performed using 10-fold cross-validation. The model selection criterion was the concordance of new samples relative to the training samples, as defined in equation (1). The tuning parameters $\gamma(P1)$, $\gamma(P2)$, $\gamma(P3)$, $\mu(P2)$ and $\mu(P3)$ where found using 10-fold cross-validation combined with a grid search, where CI^u was used as model selection criterion.

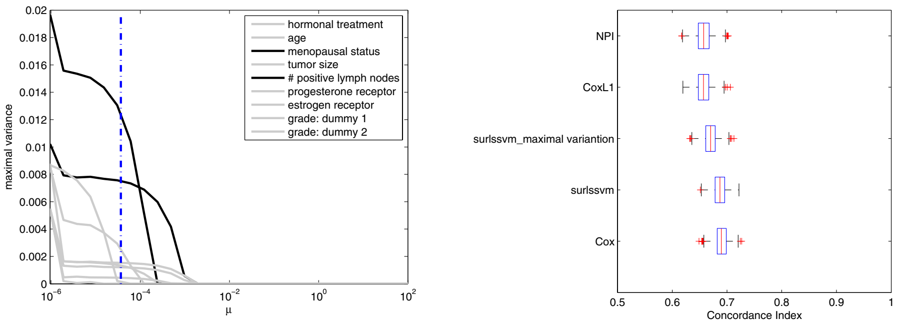
3.1 Artificial Data

In a first example we generated 100 datasets, each containing 100 training points, 100 test points and 20 variables of which only 4 contributed to the survival time.



(a) (b)

Fig. 2. **(a)** Frequency of selection for each variable in an artificial example where only the first 4 covariates contribute to the failure time (Weibull distributed). 100 models were trained with randomly chosen weights for the first 4 covariates. Variables with a maximal variation larger than one fifth of the largest maximal variation were selected. The relevant features are selected in most models, whereas the non-relevant features are only sporadically selected. **(b)** Additional 10-fold cross-validation: features selected in more than 8 folds were retained. On all 100 models, the relevant features were significantly more often selected.



(a) (b)

Fig. 3. Feature selection with SURLSSVM. **(a)** Influence of μ on the maximal variation for each component, for optimal γ. At the optimal value of μ (vertical line) hormonal treatment and the number of positive lymph nodes are selected as relevant variables. **(b)** Comparison of performances of Cox and SURLSSVM(P2-P1), CoxL1 and model SURLSSVM_maximal variation (P2-P3-P3). The models with all variables performs best. The LSSVM-based model with a subset of variables performs better than CoxL1.

All covariates were normally distributed and the survival time was Weibull distributed, depending on the covariates as $\sum_{p=1}^{20} w_p x_{(p)}$ where w_p was a randomly chosen value for $p = 1, \ldots, 4$ and 0 otherwise. Conditional independent censoring is added as follows: the censoring time is distributed exponentially, dependent on the first covariate. A point is considered to be censored in case the survival time was higher than the censoring time.

When only considering model P3, Figure 2(a) shows the frequency at which variables were selected for each of the 100 models. Features for which the maximal variation was larger than one fifth of the largest maximal variation were selected for each model. We clearly see that the relevant features are selected for nearly every model. The non-relevant variables are only sporadically selected. Results are further improved after 10-fold cross-validation (Figure 2(b)).

3.2 Breast Cancer Dataset

This example illustrates the selection of a subset of clinical variables on the German Breast Cancer Study Group data [15][1], containing information on 686 patients and 8 variables. Available variables are: hormone treatment, age, menopausal status, tumor size, tumor grade, the number of positive lymph nodes, the progesterone receptor (fmol) and the estrogen receptor (fmol). Two third of the data were used for training, the rest for testing the models. 299 (43.6%) patients had a breast cancer related event within the study time, leaving all other patients with a right censored failure time.

We compare the selected variables and the performances of our model (P1-P3-P3) (Figure 1(b)) with an L_1 regularization scheme with Cox regression (CoxL1)

---

[1] `http://www.blackwellpublishers.com/rss/Volumes/A162p1.htm`

and the Nottingham Prognostic Index (NPI), a linear prognostic model used in clinical practice. Figure 3(a) illustrates how the variation of the parameter μ influences the maximal variation for the different components. The vertical line indicates the optimal value of this parameter. A clear difference in variation between the black and gray variables is noted at this point. According to our model the number of positive lymph nodes and whether the patient received hormonal treatment or not, are relevant features. When using CoxL1, the variables with a non-zero coefficient are the number of positive lymph nodes, the progesterone receptor and the grade of the tumor. The NPI on the other hand considers the tumor size, grade and the number of positive lymph nodes as relevant variables. All kernel-based models used a polynomial kernel to fit non-linearities. Figure 3(b) compares performance of different models: Cox (Cox) and a two layer model (P2-P1) (Figure 1(a)) (SURLSSVM) using all covariates, CoxL1 and our presented model (P2-P3-P1) (SURLSSVM_maximal variation). The models with all variables performs best. The LSSVM-based model with a subset of variables performs better than CoxL1.

4 Conclusions

In this work we presented a method to select relevant features in survival analysis within an LS-SVM based model. Results on an artificial dataset show the selection of relevant variables and the rejection of non-relevant variables. In the clinical dataset used in this paper, the proposed method performs better than Cox regression with L_1 norm and the the NPI, which is used in clinical practice. Both the NPI and Cox-L1 use three different variables, whereas our method selected only two variables. Although further research is necessary, these preliminary results are promising.

References

1. Bϕvelstad, H.M.M., Nygård, S., Stϕrvold, H.L.L., Aldrin, M., Borgan, O., Frigessi, A., Lingjærde, O.C.C.: Bioinformatics 23(16), 2080–2087
2. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. Roy. Statist. Soc. Ser. B 58(1), 267–288 (1996)
3. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
4. Rakotomamonjy, A.: Variable selection using SVM-based criteria. Journal of Machine Learning Research 3, 1357–1370 (2003)
5. Ojeda, F., Suykens, J.A.K., De Moor, B.: Low rank updated LS-SVM classifiers for fast variable selection. Neural Networks 21(2-3), 437–449 (2008)
6. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. 97(1-2), 273–324 (1997)
7. Pelckmans, K., Suykens, J.A.K., De Moor, B.: Componentwise support vector machines for structure detection. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 643–648. Springer, Heidelberg (2005)

8. Pelckmans, K., Goethals, I., De Brabanter, J., Suykens, J.A.K., De Moor, B.: Support Vector Machines: Theory and Applications. In: Wang, L. (ed.) Componentwise Least Squares Support Vector Machines, pp. 77–98. Springer, Heidelberg (2005)
9. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters 9(3), 293–300 (1999)
10. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)
11. Van Belle, V., Pelckmans, K., Suykens, J.A.K., Van Huffel, S.: Additive survival least squares support vector machines. Technical report, ESAT-SISTA, K.U. Leuven (Leuven, Belgium) (submitted for publication) (2008)
12. Van Belle, V., Pelckmans, K., Suykens, J.A.K., Van Huffel, S.: Support Vector Machines for Survival Analysis. In: Proceedings of the Third International Conference on Computational Intelligence in Medicine and Healthcare (CIMED 2007), Plymouth, UK (July 2007)
13. Van Belle, V., Pelckmans, K., Suykens, J.A.K., Van Huffel, S.: Survival SVM: a Practical Scalable Algorithm. In: Proceedings of the 16th European Symposium on Artificial Neural Networks (ESANN 2008), Bruges, Belgium, April 2008, pp. 89–94 (2008)
14. Harrell, F.E., Lee, K.L., Pollock, B.G.: Regression models in clinical studies: Determining relationships between predictors and response. Journal of the National Cancer Institute 80 (1988)
15. Schumacher, M., Basert, G., Bojar, H., Huebner, K., Olschewski, M., Sauerbrei, W., Schmoor, C., Beyerle, C., Neumann, R.L.A., Rauschecker, H.F.: Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. Journal of Clinical Oncology, 12 (1994)
16. Cox, D.R.: Regression models and life-tables (with discussion). Journal of the Royal Statistical Society 34, 197–220 (1972)

A Simple Maximum Gain Algorithm for Support Vector Regression

Álvaro Barbero and José R. Dorronsoro[⋆]

Dpto. de Ingeniería Informática and Instituto de Ingeniería del Conocimiento
Universidad Autónoma de Madrid, 28049 Madrid, Spain

Abstract. Shevade's *et al.* Modification 2 is one of the most widely used algorithms to build Support Vector Regression (SVR) models. It selects as a size 2 working set the index pair giving the maximum KKT violation and combines it with the updating heuristics of Smola and Schölkopf enforcing at each training iteration a $\alpha_i \alpha_i^* = 0$ condition. In this work we shall present an alternative, much simpler procedure that selects the updating indices as those giving a maximum gain in the SVR dual function. While we do not try to enforce the $\alpha_i \alpha_i^* = 0$ condition, we show that it will hold at each iteration provided it does so at the starting multipliers. We will numerically show that the proposed procedure requires essentially the same number of iterations than Modification 2 having thus the same time performance while being much simpler to code.

1 Introduction

Given a sample $\{(X_i, t_i) : 1 \leq i \leq N\}$ and a tolerance ϵ, standard ϵ–insensitive Support Vector Regression (SVR) [1,2] tries to adjust a linear model $W \cdot X + b$ so that $|W \cdot X_i + b - t_i| \leq \epsilon$ and $\|W\|$ is minimal. In other words, the problem SVR tries to solve is

$$\min \frac{1}{2}\|W\|^2 \; s.t. \; -\epsilon \leq W \cdot X_i + b - t_i \leq \epsilon.$$

However, in practical problems, the previous restrictions have to be relaxed introducing extra slack terms and become $W \cdot X_i + b - t_i \geq -\epsilon - \xi_i$, $W \cdot X_i + b - t_i \leq \epsilon + \xi_i^*$, $\xi_i, \xi_j^* \geq 0$. The function to be minimized is now $J(W, b, \xi, \xi^*) = \|W\|^2/2 + C \sum (\xi_i + \xi_i^*)$ where C is a properly chosen penalty factor. Obviously, minimizing $J(W, b, \xi, \xi^*)$ is equivalent to minimizing $\sum_i [t_i - W \cdot X_i - b]_\epsilon + \lambda \|W\|^2$, for $\lambda = 1/2C$ and where $[z]_\epsilon = \max(0, |z| - \epsilon)$. Thus, SVR can be seen as a modeling problem where errors are measured in terms of the $[\cdot]_\epsilon$ function and a regularization term $\lambda\|W\|^2$ is added. In practice, however, the problem actually solved is the simpler dual of the previous one, namely, to minimize

---

[⋆] With partial support of Spain's TIN 2007–66862 and Cátedra UAM–IIC en Modelado y Predicción. The first author is supported by the FPU–MEC grant AP2006–02285.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 73–80, 2009.
© Springer-Verlag Berlin Heidelberg 2009

$$f(\alpha, \alpha^*) = \frac{1}{2} \sum (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) X_i \cdot X_j - \sum (\alpha_i - \alpha_i^*) t_i + \epsilon \sum (\alpha_i + \alpha_i^*) \quad (1)$$

subject to $0 \leq \alpha_i, \alpha_i^* \leq C$ and $\sum \alpha_i = \sum \alpha_i^*$. We point out that it is easy to prove that the optimal α_i^o, $(\alpha^*)_i^o$ coefficients must verify $\alpha_i^o (\alpha^*)_i^o = 0$.

SVR is a powerful modeling and prediction tool and good implementations are available such as LibSVM [3] or Weka. On the other hand, the same usefulness of SVR makes it often necessary to provide independent, stand–alone implementations because of functional specifications or intellectual property rights considerations. This makes desirable the availability of simple, easy to program algorithms. Since (1) is a quadratic programming (QP) problem, using a general QP solver could be an option. However, their coding can be quite complicated as they may involve expensive matrix computation techniques or require that the quadratic matrix is stored in memory, something that may not be possible in SVR, as the matrix dimension coincides with the sample size.

Among the algorithmic approaches to SVR giving a best performance is the one proposed of Shevade *et al.* [4] that has a SMO–like character. It first selects two updating indices l, u that most violate the Karush–Kuhn–Tucker (KKT) conditions at the present multipliers α, leading to four possible multiplier pairs, (α_l, α_u), (α_l^*, α_u), (α_l, α_u^*), (α_l^*, α_u^*). The final pair is chosen along the lines proposed by Smola and Schölkopf in [1] that take into account the pair's effect in the dual function and ensure that $\alpha_i \alpha_i^* = 0$ also hold after the update.

Here we shall propose a very simple alternative to SVR training in which a two multiplier working set is directly selected in such a way that it gives an approximately maximum gain to the objective function. Moreover, while we do not explicitly try to enforce the extra constraints $\alpha_i \alpha_i^* = 0$, it turns out that they will automatically hold provided they do so for the initial multipliers. As a consequence, our procedure essentially arrives at the same optimal multipliers than the one in [4] and requires basically the same number of iterations and time performance. On the other hand, it is much simpler and, therefore, easier to code.

The paper is organized as follows. In section 2 we shall briefly review Shevade's *et al.* approach, while in section 3 we will present our maximum gain algorithm. We shall compare both procedures in section 4 and the paper will end with a short discussion section.

2 Shevade et al. SVR Algorithm

We shall give a simplified description of the approach in [4]. As said before, it combines the selection of two updating indices based on an analysis of the KKT conditions for SVR and a second choice of the concrete multiplier pair to be updated according to the heuristics suggested in [1] Starting with the index selection, the following sets are considered first:

Table 1. Boundaries for the new $\alpha_u^{(*)}$ value, so that the SVR dual constraints still hold after the update if $\alpha_u^{(*)} \in [L, U]$

	α_l	α_l^*
α_u	$L = \max(0, \gamma - C)$, $H = \min(C, \gamma)$	$L = \max(0, \gamma)$, $H = \min(C, C + \gamma)$
α_u^*	$L = \max(0, -\gamma)$, $H = \min(C, -\gamma + C)$	$L = \max(0, -\gamma - C)$, $H = \min(C, -\gamma)$

$$\begin{aligned} I_{0a} &= \{i : 0 < \alpha_i < C\}, \;\; I_{0b} = \{i : 0 < \alpha_i^* < C\}, \;\; I_0 = I_{0a} \cup I_{0b}, \\ I_1 &= \{i : \alpha_i = 0, \alpha_i^* = 0\}, \\ I_2 &= \{i : \alpha_i = 0, \alpha_i^* = C\}, \;\; I_3 = \{i : \alpha_i = C, \alpha_i^* = 0\}. \end{aligned} \tag{2}$$

Writing $F_i = t_i - W \cdot X_i$, we define next

$$\begin{aligned} \tilde{F}_i &= F_i + \epsilon \text{ if } i \in I_{0b} \cup I_2, \\ &= F_i - \epsilon \text{ if } i \in I_{0a} \cup I_1, \\ \bar{F}_i &= F_i - \epsilon \text{ if } i \in I_{0a} \cup I_3, \\ &= F_i + \epsilon \text{ if } i \in I_{0b} \cup I_1. \end{aligned}$$

With this notation, a careful analysis of the KKT conditions for SVR implies [4] that if they hold at a given (W, b), the following inequalities must also be true:

$$b \geq \tilde{F}_i \;\forall\, i \in I_0 \cup I_1 \cup I_2, \;\; b \leq \bar{F}_i \;\forall\, i \in I_0 \cup I_1 \cup I_3.$$

Whether this holds can be very easily checked, for if we define

$$b_{up} = \min \left\{\bar{F}_i : i \in I_0 \cup I_1 \cup I_3\right\}, \;\; b_{low} = \max \left\{\tilde{F}_i : i \in I_0 \cup I_1 \cup I_2\right\}, \tag{3}$$

we should have $b_{low} \leq b_{up}$ if we are at an optimum. Conversely, if $b_{low} > b_{up}$ the optimum has not been reached yet. In this second case, the indices i_{low}, i_{up} used to compute b_{low} and b_{up} define a maximal KKT–violating pair and the so–called Modification 2 in [4] suggests to use them as a first step in the updating procedure. For convenience, we shall use the notations l, u instead of i_{low}, i_{up} in what follows.

Once these indices have been chosen, we have to decide which one of the four possible combinations of the $\alpha_l, \alpha_u, \alpha_l^*, \alpha_u^*$ has to be updated. Since we also want to enforce the $\alpha_j \alpha_j^* = 0$ restriction, we have to decide among the following four cases:

$$\begin{aligned} &1 : \text{update } \alpha_u, \alpha_l; \text{ set } \alpha_u^* = \alpha_l^* = 0; \quad 2 : \text{update } \alpha_u, \alpha_l^*, \text{ set } \alpha_u^* = \alpha_l = 0; \\ &3 : \text{update } \alpha_u^*, \alpha_l, \text{ set } \alpha_u = \alpha_l^* = 0; \quad 4 : \text{update } \alpha_u^*, \alpha_l^*, \text{ set } \alpha_u = \alpha_l = 0. \end{aligned}$$

Algorithm 1. Shevade *et al.* SMO Modification 2 for SVR

```
initialize $\alpha$ ;
while (stopping condition == FALSE) do
    find $(b_{low}, b_{up})$, $u = i_{up}$ and $l = i_{low}$ using (3) ;
    pick a valid case;
    repeat
        find the optimum value for $\alpha_u^{(*)}$;
        constrain the value found using the boundaries in table 1;
        if $\alpha^{(*)} = 0$, select adjacent case;
    until optimum case found
    update $\alpha_u^{(*)}$, $\alpha_l^{(*)}$ with their computed values ;
    set the two complementary coefficients to 0 ;
end while
```

To select the best pair, Smola and Schölkopf use the dual constraint $\sum \alpha_i = \sum \alpha_i^*$ to write down explicitly the gains $f(\alpha', (\alpha^*)') - f(\alpha, \alpha^*)$ that correspond to each pair and the updates that lead to them, which are the following:

$$\begin{aligned}
\text{Case 1}: &\quad \alpha'_u = \alpha_u + (\psi_u - \psi_l)/\|Z_{l,u}\|^2;\\
\text{Case 2}: &\quad \alpha'_u = \alpha_u + (\psi_u - \psi_l - 2\epsilon)/\|Z_{l,u}\|^2;\\
\text{Case 3}: &\quad (\alpha^*)'_u = \alpha^*_u - (\psi_u - \psi_l + 2\epsilon)/\|Z_{l,u}\|^2;\\
\text{Case 4}: &\quad (\alpha^*)'_u = \alpha^*_u - (\psi_u - \psi_l)/\|Z_{l,u}\|^2.
\end{aligned}$$

We write $\psi_i = t_i - W \cdot X_i - b$ and $Z_{i,j} = X_i - X_j$. Note that the b term cancels out in the $\psi_u - \psi_l$ differences and, hence, there is no need to compute it. Once the desired α'_u or $(\alpha^*)'_u$ has been computed, the new value for the corresponding α'_l or $(\alpha^*)'_u$ can be then obtained by observing that we must have $\alpha'_u - (\alpha^*)'_u + \alpha'_l - (\alpha^*)'_l = \alpha_u - \alpha^*_u + \alpha_l - \alpha^*_l$; we shall denote this common quantity as γ.

We have to take care of the box constraints $0 \leq \alpha'_i, (\alpha^*)'_i \leq C$, which may result in having to clip some of the previous choices. It turns out that the new $\alpha'_l, \alpha'_u, (\alpha^*)'_l, (\alpha^*)'_u$ must lie between a high bound H and a low one L; table 1 gives these H, L values for the four updating cases. Finally, we explore the four feasible cases settling in the pair giving a maximum gain overall. As usual in SVM training, starting at some feasible multiplier vector, one iteratively selects the working sets and performs the preceding steps to obtain the new multipliers. Moreover, the procedure is easily kernelizable, as dot products are the only vector operations.

While sensible and fast, the full procedure is slightly cumbersome. We show next how to derive a much simpler alternative which, as we shall see, requires essentially the same number of iterations.

3 Maximum Gain SVR

The dual formulation of SVR can be expressed in a more compact form if we "double" the number of vectors X and targets t by defining $X_i = X_{i-N}$ and

$t_i = t_{i-N}$ for $i = N+1, \dots, 2N$; we shall slightly change our multiplier notation, writing now α_i, $i = N+1, \dots, 2N$ for the α^*_{i-N} multipliers. We also introduce a set of $y_i = \pm 1$ labels, with $y_i = 1$ for $i = 1, \dots, N$, $y_i = -1$ for $i = N+1, \dots, 2N$ and write $\tilde{t}_i = t_i - y_i \epsilon$. The dual function can be now written as

$$f(\alpha) = \frac{1}{2} \sum_1^{2N} \alpha_i \alpha_j y_i y_j X_i \cdot X_j - \sum_1^{2N} \alpha_y y_i \tilde{t}_i = \frac{1}{2} \|W\|^2 - \sum_1^{2N} \alpha_i y_i \tilde{t}_i \tag{4}$$

subject to $0 \le \alpha_i \le C$ and $\sum y_i \alpha_i = 0$ and where we write $W = \sum \alpha_i y_i X_i$.

Assume now an W update of the form $\alpha'_L = \alpha_L + \delta_L$, $\alpha'_U = \alpha_U + \delta_U$, $\alpha'_i = \alpha_i$ for other i. From $\sum y_i \alpha'_i = 0$ it follows that $\delta_L y_L + \delta_U y_U = 0$; that is, $\delta_U = -y_U y_L \delta_L$. Therefore, if $W' = \sum \alpha'_i y_i X_i$, we have $W' = W + \delta_L y_L X_L + \delta_U y_U X_U = W + \delta_L y_L (X_L - X_U) = W + \delta_L y_L Z$, with $Z = Z_{L,U} = X_L - X_U$. Writing $\delta = \delta_L$, we have

$$\begin{aligned} f(\alpha') &= \frac{1}{2} \|W'\|^2 - \sum_1^{2N} \alpha'_i y_i \tilde{t}_i \\ &= \frac{1}{2} \|W\|^2 + \delta_L y_L W \cdot Z + \frac{\delta^2}{2} \|Z\|^2 - \sum_1^{2N} \alpha_i y_i \tilde{t}_i - y_L \delta_L \tilde{t}_L - y_U \delta_U \tilde{t}_U \\ &= f(\alpha) + \delta y_L W \cdot Z + \frac{\delta^2}{2} \|Z\|^2 - y_L \delta (\tilde{t}_L - \tilde{t}_U) = \Phi(\delta). \end{aligned}$$

Now we have $\Phi'(\delta) = y_L (W \cdot Z - (\tilde{t}_L - \tilde{t}_U)) + \delta \|Z\|^2$ and solving $\Phi'(\delta) = 0$ we arrive at an extreme δ given by

$$\delta = -\frac{y_L (W \cdot (X_L - X_U) - (\tilde{t}_L - \tilde{t}_U))}{\|Z\|^2} = y_L \frac{\Delta}{\|Z\|^2} = y_L \lambda, \tag{5}$$

where we write $\Delta = W \cdot X_U - \tilde{t}_U - (W \cdot X_L - \tilde{t}_L)$ and $\lambda = \frac{\Delta}{\|Z\|^2}$. The gain $f(\alpha') - f(\alpha)$ achieved by this δ value is then

$$f(\alpha') - f(\alpha) = -\frac{(y_L \Delta)^2}{\|Z\|^2} = -\frac{\Delta^2}{\|Z\|^2},$$

which in principle can be approximately minimized not paying attention to the $\|Z\|^2$ denominator and choosing

$$U = \arg\max_i \{W \cdot X_i - \tilde{t}_i\}; \quad L = \arg\min_i \{W \cdot X_i - \tilde{t}_i\}.$$

Notice then that we have $\Delta \ge 0$ and also $\lambda \ge 0$. On the other hand, we must met the restrictions $0 \le \alpha'_i \le C$. They are automatic for $i \ne L, U$. Since $\alpha'_L = \alpha_L + \delta_L = \alpha_L + y_L \lambda$, we must have $\alpha_L < C$ if $y_L = 1$ and $\alpha_L > 0$ if $y_L = -1$. Similarly, since $\alpha'_U = \alpha_U + \delta_U = \alpha_U - y_U \lambda$, we must have $\alpha_U < C$ if $y_U = -1$ and $\alpha_U > 0$ if $y_U = 1$. Thus, we must refine our previous L, U choices to

$$\begin{aligned} L &= \arg\min_i \{W \cdot X_i - \tilde{t}_i : (y_i = 1, \alpha_i < C) \vee (y_i = -1, \alpha_i > 0)\}. \\ U &= \arg\max_i \{W \cdot X_i - \tilde{t}_i : (y_i = 1, \alpha_i > 0) \vee (y_i = -1, \alpha_i < C)\}. \end{aligned} \tag{6}$$

Algorithm 2. Maximum Gain Algorithm

```
initialize $\alpha$ ;
while (stopping condition == FALSE) do
    select $L$, $U$ by (6) ;
    compute $\delta^*$ by (5) and clip it as in (9) ;
    set $\alpha_L += y_L\delta^*$, $\alpha_U -= y_L\delta^*$ ;
    update $d_i = W \cdot X_i$ using (10) ;
end while
```

Reverting to the notation in section 2, we show next that if $\alpha_i \alpha_i^* = 0$, then $\alpha_i'(\alpha^*)_i' = 0$ for $1 \le i \le N$. We will consider the following index sets $\mathcal{L}_+ = \{i : \alpha_i < C\}$, $\mathcal{L}_- = \{i : \alpha_i^* > 0\}$, $\mathcal{U}_+ = \{i : \alpha_i > 0\}$, $\mathcal{U}_- = \{i : \alpha_i^* < C\}$. Then, the L, U choices in (6) can be written as

$$L = \arg\min_i \{W \cdot X_i - t_i + \epsilon : i \in \mathcal{L}_+\} \cup \{W \cdot X_i - t_i - \epsilon : i \in \mathcal{L}_-\}, \quad (7)$$

$$U = \arg\max_i \{W \cdot X_i - t_i + \epsilon : i \in \mathcal{U}_+\} \cup \{W \cdot X_i - t_i - \epsilon : i \in \mathcal{U}_-\}. \quad (8)$$

By our assumption, $\alpha_i^* > 0$ implies $\alpha_i = 0$ and also $\alpha_i > 0$ implies $\alpha_i^* = 0$. Thus, if L is taken from $\mathcal{L}_-$ we will automatically have $\alpha_L' = \alpha_L = 0$. Similarly, if U is taken from $\mathcal{U}_+$ we will have $(\alpha^*)_U' = \alpha_U^* = 0$. On the other hand, since $W \cdot X_i - t_i + \epsilon > W \cdot X_i - t_i - \epsilon$, the only option to choose L from the $\mathcal{L}_+$ subset is that $\alpha_L^* = 0$. Similarly, the only option to choose U from the $\mathcal{U}_-$ subset is that $\alpha_U = 0$. Therefore, if these other subsets provide the choices of L and U, we will also have either $(\alpha^*)_L' = \alpha_L^* = 0$ or $\alpha_U' = \alpha_U = 0$ automatically. As a consequence, in all cases, the constraints $\alpha_i'(\alpha^*)_i' = 0$ will also hold for the new α'. Finally, to keep the box constraints, we must restrict the λ choices as follows

$$\begin{aligned} &\text{if } y_L = 1 : \lambda^* = \min\{\lambda, C - \alpha_L\}; \ \text{if } y_L = -1 : \lambda^* = \min\{\lambda, \alpha_L\}; \\ &\text{if } y_U = 1 : \lambda^* = \min\{\lambda, \alpha_U\}; \ \text{if } y_U = -1 : \lambda^* = \min\{\lambda, C - \alpha_U\}. \end{aligned} \quad (9)$$

We shall call this procedure the Maximum Gain (MG) algorithm. Since its only vector operations are dot products, it can be easily applied in a kernel setting. To speed up the L, U choices we may cache the values $d_i = W \cdot X_i$. Clearly

$$d_i' = W \cdot X_i + \lambda^* y_L (X_L - X_U) \cdot X_i = d_i + \lambda^* y_L (X_L \cdot X_i - X_U \cdot X_i), \quad (10)$$

and it is clear that MG will require $2N$ kernel operations per iteration, as it is also the case of Shevade's *et al.* method.

4 Numerical Experiments

To assess the performance of our proposed Maximum Gain algorithm we have benchmarked its fitting ability (measured in mean squared error, mse) and number of training iterations over six regression datasets. We have also run Shevade's *et al.* Modification 2 in the same scenarios for comparison purposes. All the

Table 2. Total number of patterns N, training set sizes n and input dimensions d of the studied datasets

Name	abalone	bodyfat	housing	mg	mpg	space
N	2846	252	506	1385	297	3107
n	1000	100	200	600	200	1000
d	8	14	13	6	7	6

Table 3. Averages and standard deviations MSE values for both models and number of iterations (in thousands) for Modification 2 and MG procedures

Dataset	MSE	# Iters. Modif. 2	# Iters. Max. Gain
abalone	0.404 ± 0.034	27.84 ± 9.03	27.87 ± 9.07
bodyfat	0.068 ± 0.024	7.27 ± 1.17	7.27 ± 1.18
housing	0.190 ± 0.047	11.43 ± 3.00	11.43 ± 3.00
mg	0.341 ± 0.023	11.33 ± 3.46	11.33 ± 3.45
mpg	0.116 ± 0.028	10.00 ± 3.67	10.01 ± 3.68
space	0.365 ± 0.045	22.49 ± 10.03	22.51 ± 10.00

datasets used were obtained from the LIBSVM online repository [5], and their characteristics are presented in table 2. Note that missing values are present in some of the patterns of several datasets: those patterns were discarded from our experiments. For each of the six datasets we have randomly generated 100 independent train–test splits of the complete sample to test the models obtained. The size of the training subsets are also given in table 2.

In all these cases we have used the Gaussian kernel $K(x, x') = \exp \frac{\|x-x'\|^2}{2\sigma^2}$ with $2\sigma^2 = 50$, penalty parameter $C = 10$ and insensibility parameter $\epsilon = 0.025$. As very often done in SVM settings, we will stop both algorithms when $\Delta = W \cdot X_U - \tilde{t}_U - (W \cdot X_L - \tilde{t}_L) < \tau$ for some tolerance value τ; we shall use $\tau = 10^{-5}$ in all our experiments. Table 3 shows for each dataset the average and standard deviation of the MSE values (column 2), the number of iterations (in thousands) of Modification 2 (column 3) and that of our MG algorithm (column 4). It is readily seen that both algorithms behave in a similar fashion. The mse errors are exactly the same on all datasets up to the fifth significant digit, and the number of iterations (and, hence, kernel operations) are also similar; in fact, a Wilcoxon rank test shows no significant difference at the 10% level.

5 Discussion

Shevade's *et al.* Modification 2 is one of the fastest first order procedures for SVR. While conceptually simple, the explicit enforcing of the $\alpha_i \alpha_i^* = 0$ constraint results in having to consider separately four possible multiplier choices and to settle in a single one. This makes the algorithm slightly cumbersome. Here we have proposed the alternative Maximum Gain procedure, where the updating multipliers are directly selected as those giving an approximately maximum gain to the SVR dual function. The procedure is simpler than Modification 2 as it does not have to consider several subproblems for the working set selection. Moreover, while it does not try to explicitly enforce the $\alpha_i \alpha_i^* = 0$ constraints, it turns our that MG ensures it holds in all iterations provided it does so for the initial multipliers.

As we have numerically illustrated, the MG algorithm essentially requires the same number of iterations and, hence, kernel operations than Modification 2. It thus provides an algorithm easier to understand and program than Modification 2, one of the state of the art SVR procedures. Finally, if done along the lines of, for instance, [6], the MG algorithm allows naturally the consideration of possible second order gains. We are currently studying these and other related questions.

References

1. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing 14, 199–222 (2004)
2. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
3. Hsu, C.-W., Chang, C.-C., Lin C.-J.: A practical guide to support vector classification, `www.csie.ntu.edu.tw/~cjlin/libsvmtools`
4. Shevade, S.K., Keerthi, S.S., Bhattacharyya, C., Murthy, K.R.K.: Improvements to the smo algorithm for svm regression. IEEE Transactions on Neural Networks 11, 1188–1193 (2000)
5. Chang, C.-C., Lin C.-J.: Libsvm regression dataset repository, `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html`
6. Glasmachers, T., Igel, C.: Second order smo improves svm online and active learning. Neural Computation 20(2), 374–382 (2008)

Domains of Competence of Artificial Neural Networks Using Measures of Separability of Classes*

Julián Luengo and Francisco Herrera

Dept. of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, Granada, 18071, Spain
`{julianlm,herrera}@decsai.ugr.es`

Abstract. In this work we want to analyse the behaviour of two classic Artificial Neural Network models respect to a data complexity measures. In particular, we consider a Radial Basis Function Network and a Multi-Layer Perceptron. We examine the metrics of data complexity known as *Measures of Separability of Classes* over a wide range of data sets built from real data, and try to extract behaviour patterns from the results. We obtain rules that describe both good or bad behaviours of the Artificial Neural Networks mentioned.

With the obtained rules, we try to predict the behaviour of the methods from the data set complexity metrics prior to its application, and therefore establish their domains of competence.

Keywords: Artificial Neural Networks, Classification, Data complexity, Multilayer Perceptron, Radial Basis Function Networks.

1 Introduction

The use of Artificial Neural Networks (ANNs) is very common nowadays, and they have been applied in several tasks and fields. Due their excellent adjusting capabilities, they have been successfully applied in the Data Mining ambit and many others [15], becoming a referent. Particularly, there exist recent contribution in the field for all the considered models in this work [6,14].

The prediction capabilities of classifiers are strongly dependent on the problem's characteristics. Recently has arisen an emergent field that uses a set of complexity measures applied to the problem to describe its difficulty. These measures quantify particular aspects of the problem which are considered complicated to the classification task [9]. Studies of data complexity metrics applied to particular classifications algorithms can be found in [2,3,10,16].

We are interested in analysing the relationship between ANNs and the data complexity measures based on the separability of classes. We consider two models

---

[*] This work has been supported by the Spanish Ministry of Science and Technology under Project TIN2008-06681-C06-01. J. Luengo holds a FPU scholarship from Spanish Ministry of Innovation and Science.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 81–88, 2009.
© Springer-Verlag Berlin Heidelberg 2009

of ANNs, Radial Basis Function Networks (RBFN) and Multi-Layer Preceptron (MLP).

To perform this study, we have created several binary classification data sets from real world problems, 438 ones, and computed the value of 2 metrics proposed by Ho and Basu [9]. We have analysed the intervals of the separability of classes values related to the created data sets, in which ANN methods performs well or bad, and then formulated a rule for such intervals. The rules try to describe the ranges where some information and conclusions about the behaviour of ANN methods can be stated.

This contribution is organised as follows. In Section 2 we describe the ANNs we have used. In Section 3 the considered complexity measures are described. Next, in Section 4 we include the experimental setup, the obtained results and the rules extracted, along their analysis. Finally, in Section 5 some concluding remarks are pointed out.

2 Preliminaries: Artificial Neural Networks

In this section, we will briefly describe the algorithms used. We have used the following models of ANNs:

- Multi-Layer Perceptron (MLP) with Backpropagation[12]: Classical model of a Multi-Layer Perceptron, with its weights adjusted with backpropagation. This class of networks consists of multiple layers of neurons, interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. The units of these networks apply a sigmoid function as an activation function. The parameters are:
  - Hidden Layers: 1 hidden layer.
  - Number of Neurons: 10 neurons.
- Radial Basis Function Network (RBFN)[5]: A RBF is a function which has built into a distance criterion with respect to a center. RBF networks have 2 layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. RBF networks have the advantage of not suffering from local minima in the same way as multi-layer perceptrons. The parameters are:
  - Number of Neurons: 50 neurons.

3 Data Complexity Measures Based on the Separability of Classes

In this section we describe the two metrics we have used in this contribution, with their correspondent acronym.

For our study, we will examine two measures of separability of classes from [9] which offer information for the ANN methods. They are described next.

- **N1:** fraction of points on class boundary. This method constructs a class-blind minimum spanning tree over the entire data set, and counts the number

of points incident to an edge going across the two classes. The fraction of such points over all points in the data set is used as a measure. For two heavily interleaved classes, a majority of points are located next to the class boundary. However, the same can be true for a sparsely sampled linearly separable problem with margins narrower than the distances between points of the same class.
- **N3:** error rate of 1-NN classifier. This is simply the error rate of a nearest-neighbour classifier measured with the training set. The error rate is estimated by the leave-one-out method. The measure denotes how close the examples of different classes are. Low values of this metric indicate that there is a large gap in the class boundary.

4 Experimental Study: Analysis of the ANNs with Data Complexity Measures

In this Section we analyse the obtained results for the ANN methods. First, in Subsection 4.1 we present the experimental framework, with the data sets generation method, and the global average accuracy of the ANN methods. Next we determine several rules based on ANNs behaviour in Subsection 4.2. Finally we analyse the collective evaluation of the set of rules in Subsection 4.3.

4.1 Experimental Framework: Data Sets Generation

We evaluate the ANN methods on a set of 438 binary classification problems. These problems are generated from pairwise combinations of the classes of 21 problems from the University of California, Irvine (UCI) repository [1]. These are *iris, wine, new-thyroid, solar-flare, led7digit, zoo, yeast, tae, balanced, car, contraceptive, ecoli, hayes-roth, shuttle, australian, pima, monks, bupa, glass, haberman* and *vehicle*.

In order to do that, we construct several new data sets with the combination of the examples from two different classes. This will result in a new data set with only 2 classes and with the original examples which had two such classes as output. We perform this process for every possible pairwise combination of classes. If an obtained data set with this procedure proves to be linearly-separable, we discard it. The complexity measure L1 from [9] indicates if a problem is linearly-separable.

This method for generating binary data sets is limited by the proper combinatorics, and we can only obtain over 200 new data sets with the original 20 data sets with this first approach. In order to obtain more data sets, we group the classes two by two, that is, we create a new binary data set, and each of its two classes are the combination of two original classes each. For this second approach we have used *ecoli*, *glass* and *flare* data sets, since they have a high number of class labels. Again, those data sets with a L1 value of zero are discarded.

In order to measure the ANNs performance, we have applied a 10 fold-cross validation scheme. In Table 1 we have summarized the global average Training and Test accuracy and standard deviation obtained by the ANN methods.

Table 1. Global Average Training and Test Accuracy for RBFN and MLP

	Global % Accuracy Training	Global % Accuracy Test
RBFN	93.12% ± 7.99	90.65% ± 10.42
MLP	95.98% ± 5.25	88.39% ± 10.41

4.2 Determination of Rules Based on the ANNs Behaviour

In the following we present the results of the runs over the 438 data sets summarized in Figures 1 to 4.

For each complexity measure, the data sets are sorted by the ascending value of the corresponding complexity measure, and put altogether in a Figure. In the X axis we represent the data sets, not the complexity measure value, and the Y axis depicts the accuracy obtained both in training and test. The reason to do so is to give each data set the same space in the graphic representation. For those measures where we can find different *ad-hoc* intervals which present *good* or *bad behaviour* of the ANNs, we use a vertical line to delimit the interval of the region of interest.

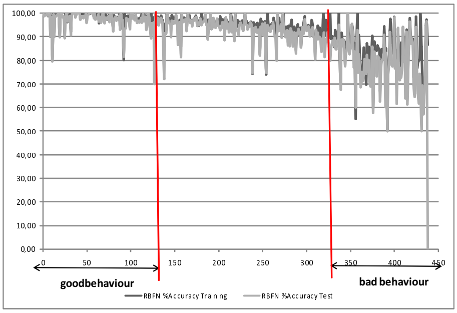


Fig. 1. RBFN accuracy in Training/Test sorted by N1

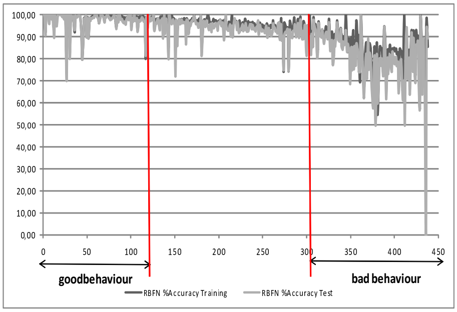


Fig. 2. RBFN accuracy in Training/Test sorted by N3

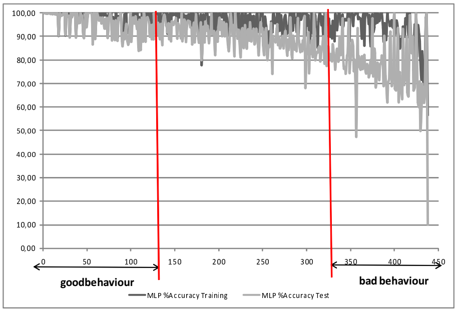


Fig. 3. MLP accuracy in Training/Test sorted by N1

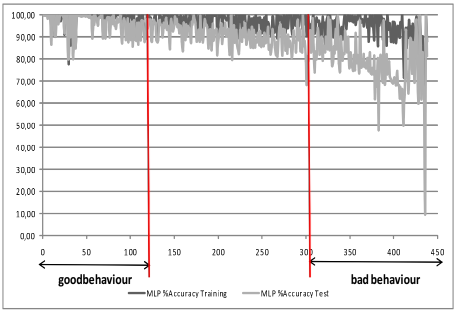


Fig. 4. MLP accuracy in Training/Test sorted by N3

Table 2. Significant intervals

Interval	ANNs Behaviour
$N1 < 0.08$	*good behaviour*
$N3 < 0.029$	*good behaviour*
$N1 > 0.25$	*bad behaviour*
$N3 > 0.108$	*bad behaviour*

- We understand for *good behaviour* an average high test accuracy in the interval, as well as the absence of over-fitting.
- By *bad behaviour* we refer to the presence of over-fitting and/or average low test accuracy in the interval.

In Table 2 we have summarized the intervals found ad-hoc from Figures 1 to 4.

From these ad-hoc intervals we construct several rules that model the performance of the ANNs we have used. In Table 3 we have summarized the rules derived from Table 2. Given a particular data set X, we get the complexity measure of X with the notation $CM[X]$. Table 3 is organised with the following columns.

- The first column corresponds to the identifier of the rule for further references.
- The "Rule'" column presents the rule itself.
- The third column "Support" presents the percentage of data sets which verifies the antecedent of the rule.
- The column "Neural Network" identifies the ANN to which this row refers to.
- The column "% Training" shows the average accuracy in training of all the examples which are covered by the rule.
- The column "Training Diff." contains the difference between the training accuracy of the rule and the training accuracy across all 438 data sets.
- The column "% Test" shows the average accuracy in test of all the examples which are covered by the rule.
- The column "Test Diff." contains the difference between the test accuracy of the rule and the test accuracy across all 438 data sets.

The positive rules (denoted with a "+" symbol in their identifier) always show a positive difference with the global average, both in training and test accuracy. The negative ones (with a "-" symbol in their identifier) verify the opposite case. The support of the rules shows us that we can characterize a wide range of data sets and obtain significant differences in accuracy.

From this set of rules we can state that a low N1 value results in a good behaviour of the ANN methods. A low N3 value obtains the same good results. In the other hand, a high value in the N1 metric produces a bad behaviour of the ANNs considered in our analysis. A high N3 value will also produce a bad behaviour of both ANN methods.

Although we have obtained some interesting rules, we can extend our study by considering the combination of these complexity metrics in order to obtain more precise and descriptive rules.

Table 3. Rules with one metric obtained from the intervals

Id.	Rule	Support	Neural Network	%Training	Training Diff.	% Test	Test Diff.
R1+	If N1[X] < 0.08 then *good behaviour*	29.22%	RBFN	98.03%	4.91%	97.20%	6.55%
			MLP	97.67%	1.69%	95.95%	7.56%
R2+	If N3[X] < 0.029 then *good behaviour*	26.71%	RBFN	98.18%	5.06%	97.28%	6.63%
			MLP	97.31%	1.33%	96.02%	7.63%
R1-	If N1[X] > 0.25 then *bad behaviour*	24.43%	RBFN	83.64%	-9.48%	78.22%	-12.43%
			MLP	93.68%	-2.30%	76.74%	-11.65%
R2-	If N3[X] > 0.108 then *bad behaviour*	31.51%	RBFN	85.33%	-7.79%	80.64%	-10.01%
			MLP	93.80%	-2.18%	78.47%	-9.92%

Table 4. Disjunction Rules from all simple rules

Id.	Rule	Support	Neural Network	%Training	Training Diff.	% Test	Test Diff.
PRD	If R1+ or R2+ then *good behaviour*	32.65%	RBFN	98.16%	5.04%	97.11%	6.46%
			MLP	97.17%	1.19%	95.52%	7.13%
NRD	If R1- or R2- then *bad behaviour*	31.96%	RBFN	85.50%	-7.62%	80.81%	-9.84%
			MLP	93.86%	-2.12%	78.57%	-9.82%
uncovered	If not PRD and not NRD then *good behaviour*	35.39%	RBFN	95.35%	2.23%	93.59%	2.94%
			MLP	96.80%	0.82%	90.70%	2.31%

4.3 Collective Evaluation of the Set of Rules

The objective of this section is to analyse the good and bad rules jointly. Thus we can arrive at a more general description and with wider support of the behaviour of the ANNs. We perform the disjunctive combination of all the positive rules to obtain a single rule, and all the negative ones. The new disjunctive rule will be activated if any of the component rules' antecedents are verified.

In Table 4 we summarize both disjunctions, and a third rule representing those data sets which are not charaterised by either disjunction rules.

From the collective rules we can observe that the support has been increased from the single rules both for the Positive Rule Disjunction (PRD) and Negative Rule Disjunction (NRD). In the other hand, the Test and Training Accuracy Differences are similar to the single rules from Table 3. Since there are no data sets in PRD and NRD simultaneously, we can consider three blocks of data sets with their respective support, as depicted in Figures 5 and 6(with no particular data set order within each block):

- The first block (the left-side one) represents the data sets covered by the PRD rule. They are the data sets recognized as being those in which the ANNs have good accuracy.
- The second block (the middle one) plots the data sets for the rule NRD, which are bad data sets for the ANNs methods considered.

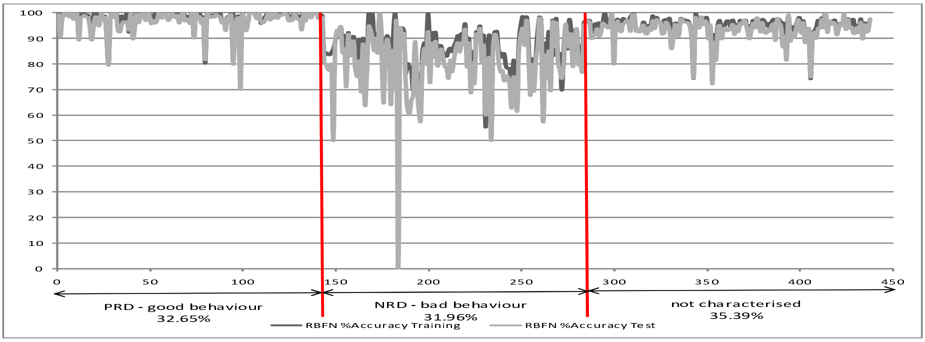


Fig. 5. Three blocks representation for PRD, NRD and not covered data sets for RBFN

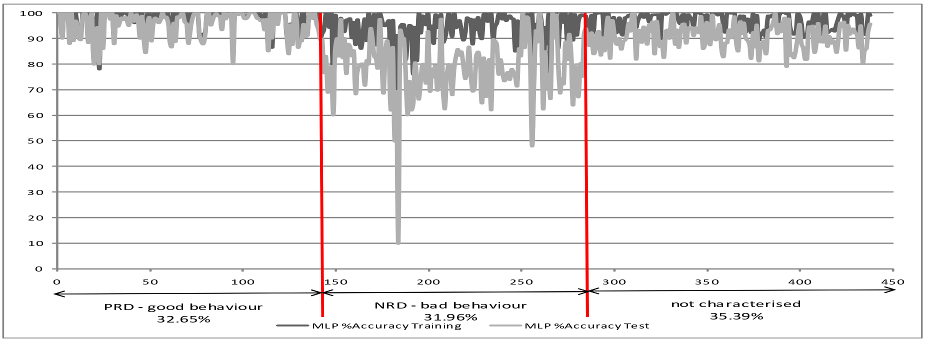


Fig. 6. Three blocks representation for PRD, NRD and not covered data sets for MLP

– The third and last block (the right-side one) contains the unclassified data sets by the previous two rules.

We can see that almost the 65% of the analysed data sets are covered by these two rules, and hence the *good behaviour* and *bad behaviour* consequents represent well the accuracy of ANN methods.

5 Concluding Remarks

We have performed a study over a set of binary data sets with two ANN methods. We have computed two data complexity measures of separability of classes for the data sets in order to obtain intervals of such metrics in which the method's performance is significantly good or bad. We have constructed descriptive rules, and studied the interaction between the intervals and the proper rules.

We have obtained two rules which are simple and precise to describe both good and bad performance of the ANNs considered in this work. Furthermore, we present the possibility of determining which data sets RBFN and MLP would perform well or badly prior to their execution, using the Data Complexity measures.

We must point out that this is a particular study for two specific methods. On the other hand, this work presents a new challenge that could be extended to other ANNs models, to analyse their domains of competence, and to develop new measures which could give more information on the behaviours of ANNs for pattern recognition.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. Baumgartner, R., Somorjai, R.L.: Data complexity assessment in undersampled classification. Pattern Recognition Letters 27, 1383–1389 (2006)
3. Bernadó-Mansilla, E., Ho, T.K.: Domain of Competence of XCS Classifier System in Complexity Measurement Space. IEEE Transactions on Evolutionary Computation 9(1), 82–104 (2005)
4. Bezdek, J.C., Kuncheva, L.I.: Nearest prototype classifier designs: An experimental study. International Journal of Intelligent Systems 16(12), 1445–1473 (2001)
5. Broomhead, D.S., Lowe, D.: Multivariable Functional Interpolation and Adaptive Networks. Complex Systems 11, 321–355 (1988)
6. Daqi, G., Chunxia, L., Yunfan, Y.: Task decomposition and modular single-hidden-layer perceptron classifiers for multi-class learning problems. Pattern Recognition 40(8), 2226–2236 (2007)
7. Dong, M., Kothari, R.: Feature subset selection using a new definition of classificabilty. Pattern Recognition Letters 24, 1215–1225 (2003)
8. Ho, T.K., Baird, H.S.: Pattern classification with compact distribution maps. Computer Vision and Image Understanding 70(1), 101–110 (1998)
9. Ho, T.K., Basu, M.: Complexity Measures of Supervised Classification Problems. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 289–300 (2002)
10. Basu, M., Ho, T.K. (eds.): Data Complexity in Pattern Recognition. Springer, Heidelberg (2006)
11. Li, Y., Dong, M., Kothari, R.: Classifiability-Based Omnivariate Decision Trees. IEEE Transactions on Neural Networks 16(6), 1547–1560 (2005)
12. Moller, F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533 (1993)
13. Mollineda, R.A., Sánchez, J.S., Sotoca, J.M.: Data Characterization for Effective Prototype Selection. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3523, pp. 27–34. Springer, Heidelberg (2005)
14. Renjifo, C., Barsic, D., Carmen, C., Norman, K., Peacock, G.S.: Improving radial basis function kernel classification through incremental learning and automatic parameter selection. Neurocomputing 72(1-3), 3–14 (2008)
15. Rojas, R., Feldman, J.: Neural Networks: A Systematic Introduction. Springer, Heidelberg (1996)
16. Sánchez, J.S., Mollineda, R.A., Sotoca, J.M.: An analysis of how training data complexity affects the nearest neighbor classifiers. Pattern Analysis and Applications 10(3), 189–201 (2007)
17. Singh, S.: Multiresolution Estimates of Classification Complexity. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1534–1539 (2003)

Self-estimation of Data and Approximation Reliability through Neural Networks

Leonardo M. Reyneri[1], Valentina Colla[2], Mirko Sgarbi[2], and Marco Vannucci[2]

[1] Politecnico di Torino, Electronic Department
Corso Duca degli Abruzzi 24, 10129, Torino, Italy
[2] Scuola Superiore Sant'Anna, PERCRO laboratory
Viale Rinaldo Piaggio, 56025 Pontedera, Italy
`leonardo.reyneri@polito.it`,{`colla`,`sgarbi`,`mvannucci`}`@sssup.it`

Abstract. This paper presents a method to estimate the reliability of the output of a (possibly neuro-fuzzy) model by means of an additional neural network. The proposed technique is most effective when the reliability of the model significantly varies in different areas of input space, as it often happens in many real-world problems, allowing the user to predict how reliable is a given model for each specific situation. Alternatively, the proposed technique can be used to analyze particular anomalies of input data set such as the outliers.

Keywords: Data reliability, Industrial databases.

1 Introduction

Neural Networks and Fuzzy Systems are known to be "good approximators" of any given problem. Unfortunately, the concept of "good' is necessarily rather vague as in many cases an error of, say, 1% is negligible, while in other applications the same error might be tremendously high.

A commonly used estimate of "goodness' is represented by a number of *global error functions* like, for instance, the *Root Mean Square Error* (RMS) [1]. These error functions are global in the sense that they give an *average* or *integral* estimate of the error of the neuro-fuzzy (NF) model with respect to the *ideal problem.*

In many cases, the amount of approximation error is approximately constant and independent of the actual input data, therefore the RMS is a good indication of how reliable is the NF approximation of the given problem in any point of the universe of discourse.

Yet there is a large amount of problems where either the problem or the NF system have some criticality concentrated in limited areas of the universe of discourse. A few examples are:

- strongly non-linear systems, where certain input conditions trigger unstable, noisy or chaotic conditions such that the output becomes either unpredictable or affected by a strong approximation error;

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 89–96, 2009.
© Springer-Verlag Berlin Heidelberg 2009

- clusters of data points which are partially overlapped, therefore similar points should in theory produce very different outputs, which is not possible with most NF systems;
- data sets with many outliers, that is, data points which have erroneously been classified and thus produce unpredictable results;
- complex systems which are approximated by NF systems with an insufficient number of neurons, therefore they cannot learn all the features of the problem. The NF system tends to be attracted by a subset of features (which are properly trained) at the expense of other features (associated with different areas of universe of discourse) which are neglected;
- classifiers trained from a data set with many more negative than positive cases (or the other way around), in which the NF system tends to learn nearly only the negative cases (which are usually less relevant) and to have a large error on the rare positive cases (which are usually the most important ones).
- problems with one-to-many (therefore, non bijective) input/output characteristic.

In all these cases, the global indication given by the RMS error is meaningless as it states that a given NF system is, in the average, very good but gives no indication whether, for instance, the error is concentrated all around the few relevant and most critical conditions, situation which completely impairs the validity and reliability of the NF system.

2 The Problem of Evaluating Reliability

Evaluating *reliability* of a NF system is surely not a trivial problem. This situation is made even worse by the lack of appropriate and general definitions of reliability.

The solution we propose is to define reliability by means of a measure of approximation error which is not global (as RMS error) but local, that is, a function $\sigma(\boldsymbol{x})$ which measures the standard deviation of the NF approximation error for each point $\boldsymbol{x}$ in the input space.

From this measure the user can infer how reliable is the NF system for each input condition and, consequently, infer where are the criticalities of either the problem or the data set(s) and take actions accordingly. For instance, if the error is concentrated in a specific area of the universe of discourse, the user can analyze the possible causes of that and, maybe, identify possible outliers (in data set) or instability (of the problem), or other particular situations.

3 Self-estimation of Reliability through Neural Networks

When applying any kind of model (both neural networks and other kind of mathematical or statistical models) to a function approximation problem, it is of utmost importance to evaluate if and how much such model is reliable, i.e. how much one can trust in the prediction the model provides. In the case of neural networks, the goodness and robustness of the prediction is strictly related to the "quality" of the data that have been used for the training: if such data are strongly affected by noise and other kind of errors, obviously the model performance will be poor. Frequently in real-world applications the data quality is a function of the same input variables that are fed to the model itself. For instance the sensitivity of a sensor that measures one of the input variables can vary with the measured quantity; moreover it is well known that the quantization error affects measurements of low values more than those of great values. Other problems may be more "difficult" in certain input conditions.

The proposed approach exploits an additional neural network to estimate the reliability of the prediction provided by a neural network–based model and of any kind of model whose parameters are tuned by exploiting a database. The basic idea of this method is illustrated by the block diagram reported in Fig. 1.

3.1 Absolute Model Errors and Variance

The rationale behind the proposed idea is to have a user-defined *model* of the system (possibly a NF system, yet it can be any other system), which approximates/predicts a given function $t_i(\boldsymbol{x_i})$ by providing an estimate y_i as a function of an input vector $\boldsymbol{x_i}$. For all those points $\boldsymbol{x_i}$ in which the model achieves a small error, the absolute error $\delta_i = (y_i - t_i)^2$ is small, while for those other points in which the model output is unreliable (for any of the reasons explained earlier), the absolute error $\delta_i = (y_i - t_i)^2$ becomes, in the average, large. The average (mathematical expectation)

$$\sigma^2(A) = \mathrm{E}\left(\delta_i\right) \quad \forall \boldsymbol{x_i} \in A$$

represents the statistical variance of NF approximation error for any given region A of the input space.

If we use an additional NF network to approximate $\sigma^2(\boldsymbol{x})$ and we train it with the values $\delta_i(\boldsymbol{x})$ and, if the size of this additional network is not too large, this additional network will provide a "smooth" approximation of $\delta_i(\boldsymbol{x})$, where the "smoothing" implies that the network will average the values δ_i for each region of input space, therefore providing mathematical expectation $\sigma^2 = \mathrm{E}\left(\delta_i\right)$.

In practice, the model produces an output vector $y(\boldsymbol{x})$, which is a function of the input $\boldsymbol{x}$: any supervised training algorithm exploits a training data set composed of N pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ with $i = 1 \dots N$ to make the output $\boldsymbol{y}(x_i)$ of the model as "similar" as possible to $\boldsymbol{t}_i$ $\forall i$. Different "similarity" criteria can be adopted (for instance, mean square error or normalized square root mean square error [2]).

Once the model is trained, for each input pattern in the training set, the absolute error $\delta_i = (\boldsymbol{y}(\boldsymbol{x}_i) - \boldsymbol{t}_i)^2$ can be measured. A new training database is therefore built, which is composed by the pairs $(\boldsymbol{x}_i, \delta_i)$ with $i = 1 \dots N$. A second feedforward neural networks (which is indicated in Fig. 1 as standard deviation predictor) is trained by exploiting this latter database, whose aim is

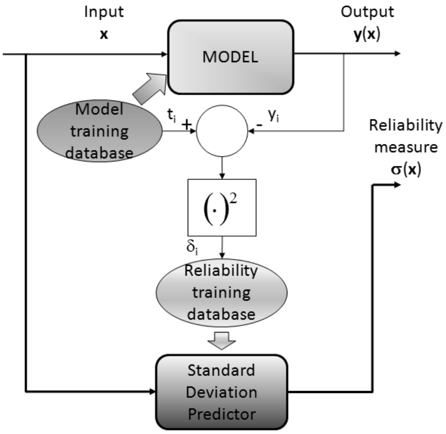


Fig. 1. Block diagram depicting the proposed approach to reliability estimate

to predict the function $\sigma^2(\boldsymbol{x})$, which is a measure of the model reliability as a function of the same input variables which are fed into the model. At the end, the whole system composed by the model and the reliability predictor is able to provide not only an estimate of the output vector $\boldsymbol{y}$ but also an indication of the reliability of such estimate. On the other hand, such reliability can depend on the kind of model that has been adopted (whose accuracy can be different in different areas of the input spaces, but also on the accuracy of the data that were used for training the model.

3.2 Identifying Potential Outliers

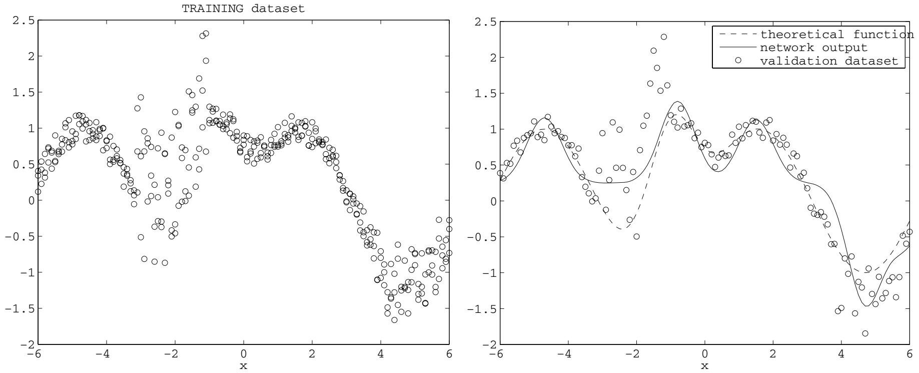
A side effect of the proposed technique is to identify potential outliers, by "flagging" all those patterns whose absolute error is larger than a multiple of the variance (as in the Grubbs test [3]):

$$\mathcal{O} = \left\{ \boldsymbol{x}_i \middle| \delta_i > k\sigma^2(\boldsymbol{x}_i) \right\}$$

where $\mathcal{O}$ is the set of potential outliers, while $k > 1$ is an appropriate parameter defined according to the application. The smaller, the more pattern are flagged as potential outliers.

4 Numerical Results

The above approach has been tested on two cases study: the first one, that is described in Sec. 4.1, has been created through simulation and concerns the approximation of a monodimensional function $f(x)$ of a scalar variable x by means of a neural network, while the second example is extracted from an industrial application, where a particular property of metal sheet must be predicted from a set of input variables related to the metal chemical composition as well as to the basic parameters of the manufacturing process.



a) b)

Fig. 2. a) Training data set that was used for the optimization of RBF network–based. b) Comparison between the function in eq. (2), the output of the trained RBF network–based model and the points in the validation data set.

4.1 A Theoretical Case Study

A synthetic database has been created, composed by couples $(t_i, y_i) \in \Re^2$ where the $t_i \in [-6, 6]$ while the output value is obtained according to the following formula:

$$t_i = f(x_i) + \nu_i + \mu_i \tag{1}$$

where

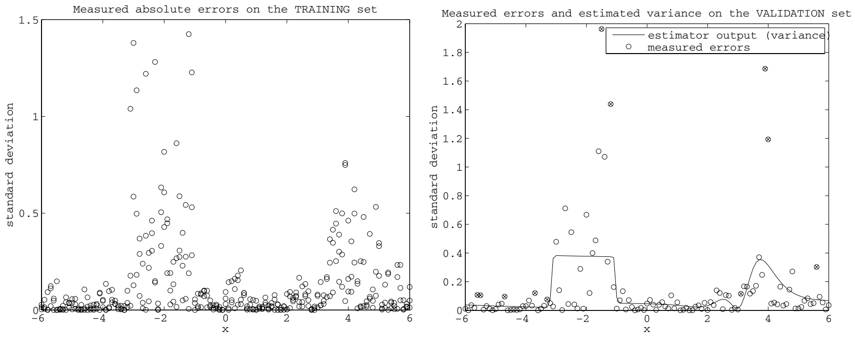
$$f(x) = 2e^{-(x+1)^2} + sin(x) \tag{2}$$

ν_i is a uniform random variable lying in the range $[-a, a]$, while μ_i is a uniform random variable whose range $[\mu_{min}(x), \mu_{max}(x)]$ is a function of x. Figure 2.a depicts an example of training data set composed by about 350 points, that has been obtained with $a = 0.2$ and the following assumptions are made:

$$\mu_{min}(x) = \begin{cases} -0.5\ for\ -3 < x < 1 \\ -0.7\ for\ 4 \leq x \leq 6 \\ 0 \qquad otherwise \end{cases} \qquad \mu_{max}(x) = \begin{cases} 1.5\ for\ -3 < x < 1 \\ 0.3\ for\ 4 \leq x \leq 6 \\ 0 \qquad otherwise \end{cases} \tag{3}$$

A validation data set composed by 120 points is also built under the same hypotheses, to test the trained model performance. The model adopted for the function approximation task is a Radial Basis Function (RBF) network [4] with 10 hidden neurons all characterized by a spreading factor equal to 0.1. Figure 2.b depicts the performance of the trained model, by comparing the theoretical function to approximate, that is given by eq.(2), the output provided by the trained RBF network and the validation data set.

Once the RBF network was trained, the training and validation data sets to be used in the design of the standard deviation predictor are built. Figure 3.a depicts an example of such training data set that was built by exploiting the data



a) b)

Fig. 3. a) Example of training data set for the standard deviation predictor (δ_i). b) Absolute errors (δ_i; circles), estimated variance ($\sigma^2(\boldsymbol{x})$; continuous line), potential outliers (crosses) for the validation set.

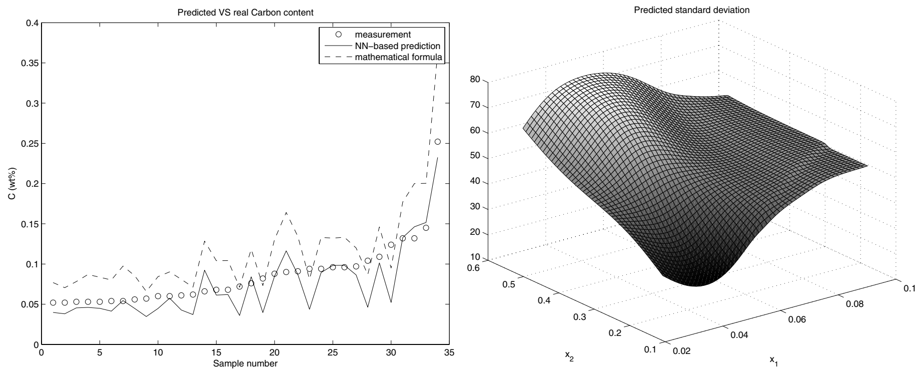
set depicted in Fig. 2.a and above-described trained RBF network–based model: i.e. the depicted samples are obtained as the absolute value of the difference between model prediction and actual data for each sample point of the training dataset reported in Fig. 2.a. The standard deviation predictor is realized with a standard Multi-Layer Perceptron (MLP) [5] with sigmoidal activation functions characterizing the hidden neurons and one linear output neuron. The dimension of such MLP has been heuristically determined. Figure 3.b illustrates the output of the best performing standard deviation predictor (which is characterized by 6 neurons in the hidden layer and was trained through the Levenberg-Marquardt method [6]), by comparing it with the points of the validation data set. Potential outliers are also identified with the method proposed above, by having $k = 3$; these are identified as crosses in Fig. 3.b.

4.2 A Practical Case Study

The aim of this work, is the prediction of the so-called *end–point* of an OBM (from the German *Oxygen Bodenblasen Maxhuette*) converter in steel making industry by exploiting the estimates of the Oxygen content and of the temperature in order to predict the final Carbon content. These three variables are usually linked by the following approximate mathematical equation:

$$Log\frac{P_c}{[C]^n[O]} = \frac{A}{T} + B \tag{4}$$

where $[C]$ and $[O]$ are, respectively, the Carbon and Oxygen content expressed in wt%, T is the temperature expressed in °K, while P_c is a constant pressure value in the range $[1, 1.5]$ atm, A, and B are constants whose values are, respectively, $A = 1895°$ K and $B = 1.6$. As far as the n value is concerned, it is commonly assumed equal to unity, but some literature results provide $n \simeq 0.5$ for $[C] < 0.08\%$ and $n \simeq 1$ otherwise.



a) b)

Fig. 4. a) Comparison between the validation data and the estimates provided by the NN-based model and the mathematical model of eq.(4). Samples are ordered for an increasing content of C; abscissa is sample index. b) Standard deviation prediction as a function of the two input variables (O and T).

Eq. (4) can easily be inverted in order to predict $[C]$ from $[O]$ and T, but the prediction obtained with the nominal values of the constant parameters is affected by a considerable error with respect to the $[C]$ measurements contained in a dataset that has been provided by the steel manufacturer for the particular steel grades that are produced in the considered plant. On the other hand, a reliable prediction of the final Carbon content at the end of the refining process is very important, as it allows to evaluate the process parameters (such as the amount of inflated oxygen and the duration of the refining process) in order to achieve some desired results by optimizing the time and cost of the production process itself. A neural network has been successfully applied to this problem [7]: an industrial database of about 200 data was exploited to train a two-layer MLP with 3 neurons in the hidden layer, a quite simple model which however was sufficient in order to reduce the prediction error of 66 % with respect to the prediction obtained through eq.(4), as it is shown also in Fig.4.a.

The above described approach for reliability estimate was applied by exploiting a two–layers MLP for standard deviation prediction: the dimension of such predictor was chosen through experiments, which showed that the best performance were obtained with 3 neurons in the hidden layer. The standard deviation strongly depends on the two input variables, as depicted in Fig. 4.b.

The proposed technique thus showed its capability of pointing out important features of the model (and of the associated training database) which, at least in the proposed example, could also be found with a targeted statistical analysis, but the proposed approach is indeed very simple and can be performed in an easy and automatic way, without the need for specific measurements and analyses, as a "by-product" of the model design phase. Moreover, it allowed to identify the area(s) in the input space where the neural model could be exploited reliably.

An additional application (not yet tested) is to identify in which input conditions one should choose either of the neural or the mathematical models, by applying the proposed technique also to the latter and to choose in which condition(s) one performs better than the other.

5 Conclusions and Future Work

A method for estimating the reliability of a model through neural networks has been presented and discussed. The model whose performance are analyzed is not necessarily based on neural networks, but surely if neural networks are exploited also for modeling purposes the effort for designing the additional standard deviation estimator is negligible with respect to its potential advantages. Through this approach, a considerable saving in the pre-processing and analysis phase of the training data can be achieved. Moreover, it is possible to better understand the model behavior as well as to exploit models whose average performance is not particularly good at least in the input areas where they provide reliable estimates. On the other hand, the user is advised when the model response is not very trustable, despite of a good average behavior. The method is also useful in order to evaluate the overall reliability of the developed predictor, for instance by computing a performance index that takes into account the relative width of the portions of the input space where the model reliability is not sufficient and an eventual weighting mechanism on the input space that takes into account the possibly different "importance" of each subset of the input space with respect to the proposed problem.

Future work will concern further tests of the developed technique on other industrial databases and models, included non-neural models, and a refinement of the developed software, in order to make the analysis easier and semi-automatic.

References

1. Casella, G., Lehmann, E.L.: Theory of Point Estimation. Springer, Heidelberg (1999)
2. Zhang, Q., Benveniste, A.: Wavelet Networks. IEEE Transactions on Neural Networks 3(6), 889–898 (1992)
3. Grubbs, F.E.: Procedures for detecting outlying observations in samples. Technometrics 11, 1–21 (1969)
4. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Computation 3(2), 246–257 (1991)
5. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, McClelland (eds.) Parallel data processing, ch. 8, vol. 1, MIT press, Cambridge (1986)
6. Marquardt, D.: An algorithm for least–square estimation of nonlinear parameters. SIAM Journal of applied mathematics (11), 164–168 (1963)
7. Valentini, R., Colla, V., Vannucci, M.: Neural predictor of the end point in a converter. Revista de Metalurgia 40(6), 416–419 (2004)

FPGA Implementations Comparison of Neuro-cortical Inspired Convolution Processors for Spiking Systems

A. Linares-Barranco, R. Paz, F. Gómez-Rodríguez, A. Jiménez, M. Rivas,
G. Jiménez, and A. Civit*

Robotic and Technology of Computers group, University of Seville,
Av. Reina Mercedes s/n, 41012-Sevilla, Spain
alinares@atc.us.es

Abstract. Image convolution operations in digital computer systems are usually very expensive operations in terms of resource consumption (processor resources and processing time) for an efficient Real-Time application. In these scenarios the visual information is divided in frames and each one has to be completely processed before the next frame arrives. Recently a new method for computing convolutions based on the neuro-inspired philosophy of spiking systems (Address-Event-Representation systems, AER) is achieving high performances. In this paper we present two FPGA implementations of AER-based convolution processors that are able to work with 64x64 images and programmable kernels of up to 11x11 elements. The main difference is the use of RAM for integrators in one solution and the absence of integrators in the second solution that is based on mapping operations. The maximum equivalent operation rate is 163.51 MOPS for 11x11 kernels, in a Xilinx Spartan 3 400 FPGA with a 50MHz clock. Formulations, hardware architecture, operation examples and performance comparison with frame-based convolution processors are presented and discussed.

1 Introduction

Digital vision systems process sequences of frames from conventional video sources, like cameras. For performing complex object recognition algorithms, sequences of computational operations are performed for each frame. The computational power and speed required makes it difficult to develop a real-time autonomous system. But brains perform powerful and fast vision processing using small and slow cells working in parallel in a totally different way. Vision sensing and object recognition in brains is not processed frame by frame; it is processed in a continuous way, spike by spike, in the brain-cortex.

The visual cortex is composed by a set of layers ([1][2]), starting from the retina. The processing starts beginning at the time the information is captured by the retina. Although cortex has feedback connections, it is known that a very fast and purely feed-forward recognition path exists in the visual cortex [1][3].

---

[*] This work was supported by Spanish grant TEC2006-11730-C03-02 (SAMANTA 2) and by the Andalussian Council project BrainSystem (P06-TIC-01417).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 97–105, 2009.
© Springer-Verlag Berlin Heidelberg 2009

In recent years significant progress has been made towards the understanding of the computational principles exploited by visual cortex. Many artificial systems that implement bio-inspired software models use biological-like (convolution based) processing that outperform more conventionally engineered machines [4]. However, these systems generally run at extremely low speeds because the models are implemented as software programs on conventional computers. For real-time solutions direct hardware implementations of these models are required. However, hardware engineers face a large hurdle when trying to mimic the bio-inspired layered structure and the massive connectivity within and between layers. A growing number of research groups world-wide are mapping some of these computational principles onto real-time spiking hardware through the development and exploitation of the so-called AER (Address Event Representation) technology.

AER was proposed by the Mead lab in 1991 [5] for communicating between neuromorphic chips with spikes (Fig. 1). Each time a cell on a sender device generates a spike, it communicates with the array periphery and a digital word representing a code or address for that pixel is placed on the external inter-chip digital bus (the AER bus). Additional handshaking lines (Acknowledge and Request) are used for completing the asynchronous communication. In the receiver chip the spikes are directed to the pixels whose code or address was on the bus. In this way, cells with the same address in the emitter and receiver chips are virtually connected by streams of spikes. Arbitration circuits ensure that cells do not access the bus simultaneously. Usually, these AER circuits are built using self-timed asynchronous logic [6].

There is a growing community of AER protocol users for bio-inspired applications in vision, audition systems and robot control, as demonstrated by the success in the last years of the AER group at the Neuromorphic Engineering Workshop series [7].

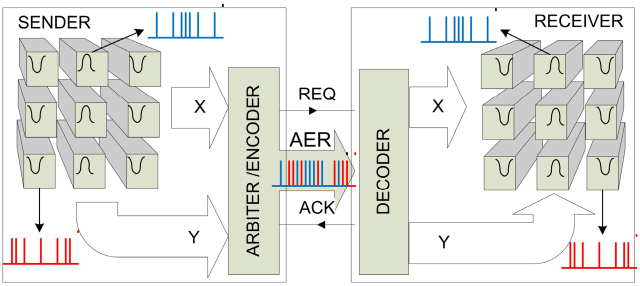


Fig. 1. Rate-coded AER inter-chip communication scheme

Table 1. AER Convolution processors comparison

Integrator based convolutions	**Mapper based convolutions**	**IF based convolutions [15]**
Poisson output	Poisson output	Arbitrated output
From 3x3 to 11x11 kernels	2x2 and 3x3 kernels	Up to 32x32 kernels
64x64 images	64x64 images	32x32 images per chip. Chips are scalables
50MHz clock	50MHz clock	200MHz clock
Fully digital 8-bit cells on RAM	No cells. Results on-the-fly by mappings	Analog integrators (capacitors)

In this paper we present two FPGA implementations of neuro-cortical inspired convolution processors. These circuits are similar from a previous work implemented in a VLSI chip [13], but with some differences, presented in table 1.

Section two explains how to convolve using spiking neurons. A VLSI chip [15] uses this mechanism. This paper presents an FPGA implementation with Poisson output distribution in section 3. In section 4 a new way of implementing convolutions not based on integrators is presented. Then in section 5 implementation details are presented. Section 6 discusses the performance of these spike-based convolution processors compared to frame-based digital convolution processors implemented on FPGA.

2 Convolutions with Spikes

a) Description. Complex filtering processing based on AER convolution chips already exists [13]. These chips are based on IF neurons. Each time an event is received a kernel of convolution is copied in the neighbourhood of the targeted IF neuron. When a neuron reaches its threshold, a spike is produced and the neuron is reset.

Bi-dimensional image convolution is defined mathematically by the following equation, being *K* an *nxm* kernel matrix of the convolution, *X* the input image and *Y* the convolved image.

$$\underset{i,j}{\forall} \rightarrow Y(i,j) = \sum_{a=-n/2}^{n/2} \sum_{b=-m/2}^{m/2} K(a,b) \cdot X(a+i, b+j)$$

Each convolved image pixel *Y(i,j)* is defined by the corresponding input pixel *X(i,j)* and weighted adjacent pixels, scaled by *K* coefficients. Therefore an input pixel *X(i,j)* contributes to the value of the output pixel *Y(i,j)* and their neighbours, multiplied by the corresponding kernel coefficients *K*.

For implementing convolutions using spikes let's suppose *Y* a matrix of integrators (capacitors for analog circuits or registers for digital circuits) to store the result of applying a kernel of convolution to an input image *X* that is coded into a stream of events through the AER protocol. Each pixel *X(i,j)* represents a gray value *G* in the original image. Let's suppose that in the AER bus will be represented, for a fixed period of time, an amount of events *P·G* (proportional to the gray level of the pixel). Each event identifies the source pixel by the address on the AER bus. The value of the pixel is coded by the frequency of appearance of this address on the bus. For each event coming from the continuous visual source (e.g. an AER retina or a synthetic AER generator), the neighbourhood of the corresponding pixel address in *Y* is modified by adding the convolution kernel *K*, stored in a RAM memory and previously configured. Thus, each element of *Y* is modified when an event with the address *(i,j)* arrives with the following equation:

$$Y(i+a, j+b) = Y(i+a, j+b) + K(a,b), \ \forall a,b \in \dim(K)$$

Once all the events of the pixel *X(i,j)* have been received and calculated, the integrator value of the corresponding address *Y(i,j)* has accumulated *X(i+a,j+b)*, $\forall a,b$ (the gray values) times the value of the kernel, so the multiplication operation has been reached.

The output of the convolution operation, at this point is stored in the matrix of integrators *Y*. This result can be sent out in several ways: (a) scanning all the integrators values and sending out an analog or digital signal. Each integrator or register is reset after reading. The system is converted into a frame-based output, so the neuro-inspired nature is lost. (b) Based on IF neuron, when an integrator reaches a threshold a spike is produced and the corresponding AER event is produced. The system is totally spike-based, but the output cannot follow a Poisson distribution due to the IF neuron [8]. Every time a spike is produced by a neuron, this is reset. This solution is used by analog convolution chips [13]. (c) Generate synthetically a stream of spikes. Having the result in the *Y* matrix, a method for synthetic AER generation can be used to warranty the Poisson distribution of spikes [10]. Since the AER Poisson generation method is accessing randomly *Y* matrix, this cannot be reset. A periodic and parameterized mechanism of forgetting is used instead of reseting cells when they spike. Periodically, the convolved matrix *Y* is accessed for subtracting a constant value to each element.

In [13] the convolution chip is able to receive positive and negative events to process signed kernels and therefore, it is also able to produce signed output events. This is done by duplicating the number of integrators, having a positive and a negative integrator per cell. If the positive integrator reaches the cero and additional negative values arrive to the cell, the negative integrator starts to work. The output event produced will be signed depending on the integrator used to produce that event.

b) FPGA Implementation I: Integrators on RAM. Figure 2 shows the block diagram of the first architecture. It is composed basically by two devices: a FPGA and a microcontroller. The FPGA is in charge of the AER-based convolution processor and the microcontroller is responsible of the PC interface for configuring all the parameters (Kernel matrix, kernel size, forgetting period and forgetting quantity). The interface between the microcontroller and the FPGA uses a Serial Peripheral Interface (SPI). The circuit in the FPGA can be divided into four parallel blocks:

- *AER event reception and processing.* The input traffic AER protocol is managed by the "AER input" block. Each received event *(i,j)* is used to address the *Y* matrix (64x64 cells). Centered on *(i,j)*, the kernel is copied in the *Y* matrix. The neighborhood of the cell *(i,j)* will modify their values by adding the corresponding kernel value. Therefore, *Y* matrix is always up-to-date with the convolution result. *Y* matrix is implemented using a block of dual-port RAM in the FPGA. Each position of the RAM is 8-bit length. The kernel is stored in an 11x11 matrix with 8-bit resolution. Each kernel value is in the range of -127 to 127. When the kernel is added the result stored in the matrix is limited between 0 and 255, because no signed events are implemented.
- *Forgetting mechanism.* For high AER input bandwidth and / or weighted kernels, maximum cell value could be reached quickly and thus, next events are lost. Let's call this situation saturation effect. For this reason, to avoid errors, a configurable forgetting circuitry is present in the architecture. The forgetting is based on a programmable counter that accesses the *Y* matrix periodically in order to decrease its values by a constant. Forgetting period and quantity are configured through USB. Since the saturation effect depends on the AER input bandwidth and the kernel values (weight), the forgetting parameters depend on these two.

- *Poisson like AER output. Y* always has the result of the convolution. Thanks to the forgetting mechanism, this matrix can be captured at any time with a valid convolved result. To warranty the Poisson distribution of output events, the *Y* matrix is accessed by a Random AER synthetic generator [9][10].
- *Configuration.* The controller is in charge of receiving kernel size and values, forgetting period and amount to forget. An SPI interface connects a USB microcontroller and the FPGA. A computer with MATLAB controls the system.

The system has been developed in hardware in a Spartan 3 FPGA. The testing scenario consists on a USB-AER working as Uniform AER generator [9], the output is connected to an AER-Robot working as the AER convolution processor; and its output is connected to a second USB-AER working as an AER datalogger or framegrabber. A laptop is connected to the three boards, to manage all of them from MATLAB, through USB.

Figure 2 (left, top row) shows a source image and its negative version. These images are converted into AER using an AER Uniform generator to feed the convolution processor. When applying a kernel for edge detection (ON to OFF, see figure 2, matrix K), opposite responses are expected. Bottom row shows the reconstructed normalized histograms for 512Kevents obtained with the AER datalogger [11], from convolution processor output. The middle column images are the result of applying a MATLAB *conv2()* function with the same kernel to the source bitmap used by AER generator.

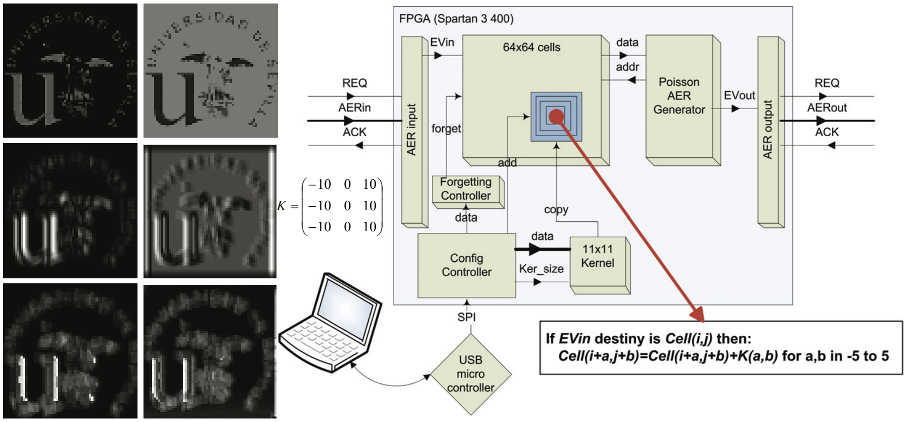


Fig. 2. Right: block diagram of the FPGA AER-based convolution processor. Left: source images (top), MATLAB simulation images result (center) and FPGA AER convolution output histograms (bottom).

3 Spike-Based Convolutions with a Probabilistic Mapper

a) Description. Let's explain how to implement this using a probabilistic multi-event mapper [15]. This mapper is able to send *M* different events per input event. Let's suppose that M is the number of elements of a convolution kernel K. Each of these mapped events has associated a repetition factor (R) and a probability (P) to be sent or not. For each input event (In_i), up to *dim(K)* output events are generated.

$$Out_{a,b} = R_{a,b} \cdot P_{a,b} \cdot In_i \quad \forall a,b \in \dim(K)$$

Thus a projective field of events is generated for each input event. From the receiver point of view, the number of events for the same address depends on probability and repetition factor of the input events in the neighbourhood. This neighbourhood is defined by the kernel size. Therefore, the number of events for a fixed output address follows the expression:

$$Nev\_Out_i = \sum_{a,b} In_{a,b} \cdot R_{a,b} \cdot P_{a,b} = \sum_{a,b} In_{a,b} \cdot K_{a,b} \Rightarrow K_{a,b} = R_{a,b} \cdot P_{a,b} \quad \forall a,b \in \dim(K)$$

So, to calculate the repetition and probability, the following equation can be applied:

$$R_i = \lceil K_i \rceil \text{ ; } and \ P_i = K_i / R_i$$

For example, for a desired kernel coefficient $K_i=1.2$, repetition factor is $R_i=2$ and probability is $P_i=0.6$.

b) Implementation II: Probabilistic Mapper. The identification or address of each emitter neuron that is present in the AER bus, is used to index a mapping table in the AER mapper. Then, a list of mapped addresses is sent replacing the original one. Each mapped address is stored in the mapping table with two parameters: a repetitious factor (R) and an output probability (P). A FSM is in charge of sending the list of events if the probability function allows it and repeating each mapped event according to R (see figure 3, left).

With this mapper, sophisticated operations can be performed in the AER traffic during the transmission time, like small kernel convolutions. As the probability can be modified per each mapped event, it is possible to implement a *one-to-dim(K)*

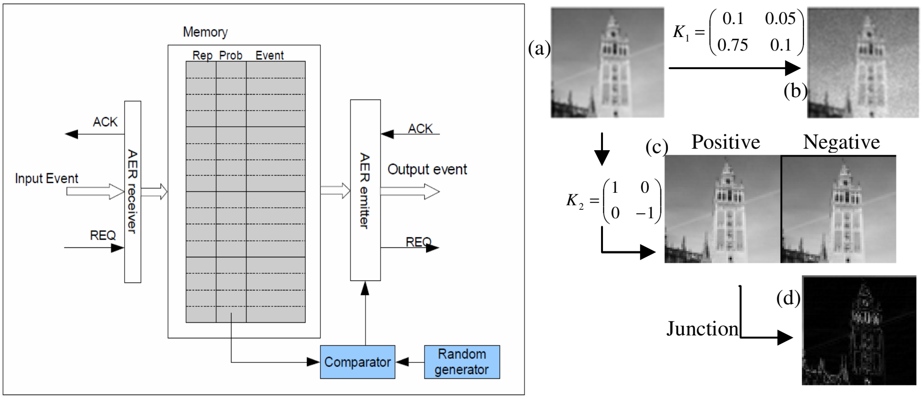


Fig. 3. Left: Probabilistic Multi-event mapper block diagram. Right: (a) source image, (b) K1 convolution histogram, (c) K2 postive output traffic and negative traffic, (d) abs(positve+negative) K2 histogram.

mapping, where each of these *dim(K)* events can be modulated with *P* and *R*, as expressed in previous section. Thus, this architecture is able to send weighted traffic to a neighbourhood per each input event. Furthermore, mapped events can be positive and negative, so signed events appears in the output AER bus. A receiver must take the average between positive and negative events per each address.

An internal LFSR (Left Feedback Shift Register) is used to generate pseudorandom numbers which are compared with the probability in order to decide if the mapped event is sent. In this way a Poisson output distribution is obtained [11][12].

This architecture has been tested in a USB-AER board that is composed by a Spartan II FPGA and 2Mb of SRAM memory (mapping table) [11]. It is able to process a *one-to-one* mapping in *120ns*, and a *one-to-M* effective mapping in *(60+M·60) ns*. If a mapped event is not sent the time consumption decreases in 20ns.

Figure 3 right shows two examples. The same source image (64x64) in AER format, generated by the Uniform AER generator [9], has been used (a). The second image is the normalized histogram of collecting AER traffic for 100ms, after applying a simple 2x2 kernel convolution K1 (b). The resulting image should be the initial one with soft-edges. An added random noise, due to Poisson distribution of mapped events, is introduced.

Images (c) and (d) belong to a second example, where an edge detection kernel (K2) is applied. The mapper sends a positive and a negative event per each input event with different addresses. Positive and negative (c) normalized histograms are shown. The last image (d) shows the absolute value of the combination of positive and negative traffic, using an up-down counter per address.

4 Comparison with Digital Frame-Based Convolution Processors

Digital frame-based convolution processors implemented in FPGA, GPUs or CPUs usually measure their performance by calculating the number of operations per second (MOPS). In this way, a frame-based convolution processor has to calculate ADD and MULT operations done to apply a kernel of convolution to an input frame. Once all the operations are executed for the whole frame, a result frame is obtained and sent to another stage. In [16][17] a performance study was presented for different kernel sizes and platforms (see table 2).

A spike-based convolution processor with integrators cells is performing a number of ADD operations per each input event, but no MULTs are done because these are replaced by an explosion in the number of ADD operations due to the spike-based representation of the information. In this case, the number of ADD per input event is equal to the kernel size, since the kernel is added to the neighborhood of the pixel address of the incoming event. Measuring AER protocol time in the input, it is feasible to calculate the MOPS.

There exist previous works related to convolution solving based on AER, like [13], where a 32x32 pixels convolution chip is based on analog Integrate and Fire neurons. This chip can process images with a 31x31 kernel. Each input spike needs 330 ns to be processed. Therefore, this system yields 31x31/330=2910MOPS.

FPGA implementation explained in section 3 is accessing sequentially all the kernel elements and adding their values to the corresponding cells. As cells are stored in RAM memory and this is read cell by cell, time required per each input event grows linearly with the kernel size. In this case the hardware implements only one

ADD. For example, for 3x3 kernel, the system consumes one cycle for reading the cell value and the corresponding kernel position, a second cycle is used for the add operation and for writing the result in the internal RAM. Thus 3x3x2 cycles are needed per each input event, apart from those needed for the handshake. The architecture can be easily improved by having a number of adders equivalent to the kernel row size, and allowing the access to the RAM by blocks. Table 2 shows the real performance in MOPS for a one ADD architecture, and the simulated performance for the N ADD architecture of the NxN kernel convolution processor.

A spike-based convolution processor based on the probabilistic mapper is not calculating neither MULT nor ADD operations. It works by projecting the input traffic following a 1 to N^2 mapping in a first stage, and then by cancelling negative and positive events of the same address in a short period of time. In this way one possible equivalent number of operations can be the number of mappings per each input event. Table 2 shows the results.

Table 2. Mega Operations per Second for frame and spike-based convolution processors

MOPS	Digital frame based 2D convolutions [16][17]				Spike-based 2D Convolutions			
					By mapping	By RAM integrators		By analog caps [13]
Kernel size (NxN)	Pentium 4 3GHz CPU	6800 ultra GPU	Spartan 3	Virtex II-Pro	Spartan II (100MHz)	Spartan 3 (real) 1 sum (50MHz)	Spartan 3 (sim) N sum (50 MHz)	VLSI 0,35um (200MHz)
2x2	14	1070	190	221	18,2	-	-	-
3x3	9,7	278	139	202	22,5	20,45	34,61	-
5x5	5,1	54	112	162	-	23,14	65,78	-
7x7	2,6	22	90	110	-	24,01	98,00	-
9x9	1,6	9	73	61	-	24,39	130,00	-
11x11	1,2	4,7	23	48	-	24,59	163,51	-
31x31	-	-	-	-	-	-	495,00	2910

5 Conclusions

Two neuro-cortical layers of 64x64 cells with NxN convolution kernel hardware implementation on FPGA have been presented. One can work with kernels of up to N=11 (~500Ksynapses), with 8-bit integrator cells, and the other with N=3 (~37Ksynapses), with no integrator cells. The AER output of both follows a Poisson distribution since LFSR are used. Both are compared with VLSI analog implementation and tested for image edge detection filtering. Poisson output distribution of events and low cost are the main differences and strongest contributions of this approach. A performance study is presented comparing frame-based digital convolution processors and VLSI analog-integrators spike-based convolution processors. Digital frame-based architectures are limited by the number of parallel ADDs and MULTs so their performance decreases while the kernel size increases. Digital RAM-cell spike-based architecture is limited by RAM access. When parallelizing the number of ADDs, the performance increases with the kernel size. Spike-based Mapper architecture avoids ADD operations since the operations consist in projecting inputs events with weighted and signed events and then averaging them in short periods of time. The performance of these systems depends on SRAM access time. Kernel sizes are limited by the mapping table capacity.

References

1. Serre, T.: Learning a Dictionary of Shape-Components in Visual Cortex:Comparison with Neurons, Humans and Machines, PhD dissertation, MIT. Comp. Sci. & AI Lab Technical Report, MIT-CSAIL-TR-2006-028 CBCL-260 (April 2006)
2. Shepherd, G.: The Synaptic Organization of the Brain, 3rd edn. Oxford University Press, Oxford (1990)
3. Thorpe, S., et al.: Speed of processing in the human visual system. Nature 381, 520–522 (1996)
4. Neubauer, C.: Evaluation of Convolution Neural Networks for Visual Recognition. IEEE Trans. on Neural Networks 9(4), 685–696 (1998)
5. Sivilotti, M.: Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks. Ph.D. Thesis, California Institute of Technology, Pasadena CA (1991)
6. Boahen, K.: Communicating Neuronal Ensembles between Neuromorphic Chips. Neuromorphic Systems. Kluwer Academic Publishers, Boston (1998)
7. Cohen, A., et al.: Report to the National Science Foundation: Workshop on Neuromorphic Engineering, Telluride, Colorado, USA (June-July 2004), `http://www.ini.unizh.ch/telluride`
8. Softky, W.R., Koch, C.: The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. J. Neurosci. 13(1), 334–350 (1993)
9. Linares-Barranco, A., Jimenez-Moreno, G., Linares-Barranco, B., Civit-Ballcels, A.: On Algorithmic Rate-Coded AER Generation. IEEE Trans. On Neural Networks 17(3) (May 2006)
10. Linares-Barranco, A., Oster, M., Cascado, D., Jimenez, G., Civit-Balcells, A., Linares-Barranco, B.: Inter-Spike-Intervals analysis of AER-Poisson-like generator hardware. Neurocomputing 70(16-18), 2692–2700 (2007)
11. Paz, R., et al.: Test Infrastructure for Address-Event-Representation Communications. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 518–526. Springer, Heidelberg (2005)
12. Linares-Barranco, A., Paz, R., Jiménez, A., Varona, S., Jiménez, G.: An AER-based Actuator Interface for Controlling an Antrophomorphic Robotic Hand. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4528, pp. 479–489. Springer, Heidelberg (2007)
13. Serrano-Gotarredona, R., et al.: A Neuromorphic Cortical-Layer Microchip for Spike-Based Event Processing Vision Systems. IEEE Trans. on Circuits and Systems I. 53(12) (December 2006)
14. Serrano-Gotarredona, T., Andreou, A.G., Linares-Barranco, B.: AER image filtering architecture for vision processing systems. IEEE Trans.Circuits and Systems (Part II): Analog and Digital Signal Processing 46(9), 1064–1071 (1999)
15. Linares-Barranco, A., et al.: Implementation of a time-warping AER mapper. In: ISCAS 2009, Taiwan (2009)
16. Cope, B., et al.: Implementation of 2D Convolution on FPGA, GPU and CPU. Imperial College Report, `http://cas.ee.ic.ac.uk/people/btc00/index_files/Convolution_filter.pdf`
17. Cope, B., Cheung, P.Y.K., Luk, W., Witt, S.: Have GPUs made FPGAs redundant in the field of video processing? In: Proc. IEEE International Conference on Field-Programmable Technology 2005, pp. 111–118 (December 2005)

Nonparametric Location Estimation for Probability Density Function Learning

Ezequiel López-Rubio, Juan Miguel Ortiz-de-Lazcano-Lobato,
and María Carmen Vargas-González

Department of Computer Languages and Computer Science
University of Málaga
Bulevar Louis Pasteur, 35. 29071 Málaga, Spain
`{ezeqlr,jmortiz}@lcc.uma.es`

Abstract. We present a method to estimate the probability density function of multivariate distributions. Standard Parzen window approaches use the sample mean and the sample covariance matrix around every input vector. This choice yields poor robustness for real input datasets. We propose to use the L1-median to estimate the local mean and covariance matrix with a low sensitivity to outliers. In addition to this, a smoothing phase is considered, which improves the estimation by integrating the information from several local clusters. Hence, a specific mixture component is learned for each local cluster. This leads to outperform other proposals where the local kernel is not as robust and/or there are no smoothing strategies, like the manifold Parzen windows.

Keywords: Probability density estimation, nonparametric modeling, median estimation, soft clustering, Parzen windows, robust statistics.

1 Introduction

The estimation of the unknown probability density function (PDF) of a continuous distribution from a set of input data forming a representative sample drawn from the underlying density is a problem of fundamental importance to all aspects of machine learning and pattern recognition (see [2] and [14]).

Parametric methods make a priori assumptions about the unknown distribution. They consider a particular functional form for the PDF and reduce the problem to the estimation of the required functional parameters. On the other hand, nonparametric approaches make less rigid assumptions. Popular nonparametric methods include the histogram, kernel estimation, nearest neighbor methods and restricted maximum likelihood methods, as can be found in [8], [9] and [6].

The kernel density estimator, also commonly referred as the Parzen window estimator [12], places a local Gaussian kernel on each data point of the training set. Parzen windows estimates are usually built using a 'spherical Gaussian' with a single scalar variance parameter σ^2. This drawback is partially solved in Manifold Parzen Windows algorithm [15], where a different covariance matrix is calculated for each component. The mean and the covariance matrix are estimated by considering a hard neighborhood of each input sample, and the sample mean is used for this purpose.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 106–113, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The sample mean is known to be very sensitive to outliers [3]. More robust location estimators have been developed [4]. In particular, the L1-median has an efficient learning algorithm [7] and has been proven to experience little degradation when outliers are present [5]. We propose in Section 2 to use the L1-median as an estimator of the mean vector, which is also used to estimate the covariance matrix.

In Section 3 we apply the smoothing technique presented in [10] to build soft clusters which share the information among local neighborhoods. This leads to further filtering of the input noise by smoothing the estimated parameters.

The asymptotical convergence of the proposed method is formally proven in Section 4. We show some experimental results in section 5, where our method produces more precise density estimations than the Manifold Parzen Windows and other approaches. Finally, Section 6 is devoted to conclusions.

2 Robust Location Estimation

Let **x** be an *D*-dimensional random variable and *p()* an arbitrary probability density function over **x** which is unknown and we want to estimate. The training set of the algorithm is formed by *N* samples of the random variable. For each training sample $\mathbf{x}_i$ we build a hard *Q*-neighborhood H_i with the *Q* nearest neighbours of $\mathbf{x}_i$, including itself. Hence H_i is interpreted as a random event which happens iff the input belongs to that neighborhood. The knowledge about the local structure of the distribution around $\mathbf{x}_i$ is obtained when we calculate the mean vector $\boldsymbol{\mu}$, the covariance matrix **C** and the correlation matrix **R**. The mean vector is estimated as the L1-median of H_i:

$$\boldsymbol{\mu}(H_i) = E[\mathbf{x} \mid H_i] = \arg\min_{\mathbf{z}} \sum_{\mathbf{x}_j \in H_i} \left\| \mathbf{z} - \mathbf{x}_j \right\| \tag{1}$$

where the minimization is carried out by the algorithm considered in [7]. This multivariate median has a breakdown point of 1/2, i.e. when just under 50% of the data is moved to infinity, the median remains in the vicinity of the majority of the data. Note that the sample mean has a breakdown point of $1/Q$, because only one input sample at infinity makes it go to infinity too. The correlation matrix **R** and the covariance matrix **C** are also estimated with the help of the L1-median:

$$\mathbf{C}(H_i) = E\left[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T \mid H_i\right] = \frac{1}{Q} \sum_{\mathbf{x}_j \in H_i} (\mathbf{x}_j - \boldsymbol{\mu}(H_i))(\mathbf{x}_j - \boldsymbol{\mu}(H_i))^T \tag{2}$$

$$\mathbf{R}(H_i) = E\left[\mathbf{x}\mathbf{x}^T \mid H_i\right] = \mathbf{C}(H_i) + \boldsymbol{\mu}(H_i)\boldsymbol{\mu}(H_i)^T \tag{3}$$

3 Smoothing

3.1 Soft Clustering

Now we present a smoothing procedure to merge the information from different hard neighborhoods. We define a soft cluster *i* by a random event S_i which verifies when the input belongs to cluster i. Each hard neighborhood H_j contributes to S_i with a weight w_{ij}:

$$w_{ij} = P[H_j \mid S_i] \tag{4}$$

$$\forall i \in \{1,2,\ldots,M\}, \quad \sum_{j=1}^{N} w_{ij} = 1 \tag{5}$$

where the number of soft clusters M may be different from the number of hard neighborhoods N. We can infer the structure of the soft cluster by merging the information from the hard neighborhoods:

$$\boldsymbol{\mu}(S_i) = E[\mathbf{x} \mid S_i] = \sum_j P[H_j \mid S_i] E[\mathbf{x} \mid H_j] = \sum_{j=1}^{N} w_{ij}\boldsymbol{\mu}(H_j) \tag{6}$$

$$\mathbf{R}(S_i) = E[\mathbf{x}\mathbf{x}^T \mid S_i] = \sum_j P[H_j \mid S_i] E[\mathbf{x}\mathbf{x}^T \mid H_j] = \sum_{j=1}^{N} w_{ij}\mathbf{R}(H_j) \tag{7}$$

$$\mathbf{C}(S_i) = E[(\mathbf{x} - \boldsymbol{\mu}(S_i))(\mathbf{x} - \boldsymbol{\mu}(S_i))^T \mid S_i] = \mathbf{R}(S_i) - \boldsymbol{\mu}(S_i)\boldsymbol{\mu}(S_i)^T \tag{8}$$

Finally, we need a method to determine the merging weights w_{ij}. We use the two approaches considered in [10]:

a) If M=N, we can perform the smoothing by replacing the 'hard' model at the data sample $\mathbf{x}_i$ by a weighted average of its neighbours ranked by their distance to $\mathbf{x}_i$.

b) We may use the fuzzy c-means algorithm [1] to perform a soft clustering. This algorithm partitions the set of training data into M clusters so it minimizes the distance within the cluster.

3.2 Probability Density Function Estimation

Once we have the estimations of the mean vectors $\boldsymbol{\mu}(S_i)$ and covariance matrices $\mathbf{C}(S_i)$ for each soft cluster S_i, it is needed to obtain a Gaussian distribution from them which is both accurate and small sized. A variable number, K_i, of eigenvalues of $\mathbf{C}(S_i)$ and their corresponding eigenvectors to be kept which is computed independently for each cluster S_i. The method ensures that a minimum amount of variance is conserved in order to satisfy the level of accuracy, $\alpha \in [0,1]$, chosen by the user:

$$K_i = \min\left\{ Z \in \{0,1,\ldots,D\} \mid \sum_{p=1}^{Z} \lambda_i^p \geq \alpha \; trace(\mathbf{C}(S_i)) \right\} \tag{9}$$

A second level of automated adaptation to the data will be added by means of a parameter γ. This parameter enables the method to select the right noise level. We propose to select the noise variance parameter σ^2 as follows:

$$\sigma^2(S_i) = \gamma \cdot \lambda_i^{K_i} \tag{10}$$

where $\gamma \in [0,1]$ and $\lambda_i^{K_i}$ is the last of the preserved eigenvalues of $\mathbf{C}(S_i)$, i.e. the smallest of the first K_i largest eigenvalues. The estimated noise variance $\sigma^2(S_i)$ is added to the K_i largest eigenvalues to yield the estimated variances of the K_i principal directions, while the D–K_i trailing directions have estimated variances of $\sigma^2(S_i)$.

The generated estimator will be formed by a mixture of M Gaussians, one for each soft cluster:

$$\hat{p}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} \exp\left(-\frac{1}{2}(a_i + b_i)\right) \tag{11}$$

$$a_i = D\log(2\pi) + (D - K_i)\log\left(\lambda_i^p + \sigma^2(S_i)\right) + \sum_{p=1}^{K_i}\log\left(\lambda_i^p + \sigma^2(S_i)\right) \tag{12}$$

$$b_i = \frac{1}{\sigma^2(S_i)}\left\|\mathbf{x} - \boldsymbol{\mu}(S_i)\right\|^2 + \sum_{p=1}^{K_i}\left(\frac{1}{\lambda_i^p} - \frac{1}{\sigma^2(S_i)}\right)\left\|\left(\mathbf{u}_i^p\right)^T\left(\mathbf{x} - \boldsymbol{\mu}(S_i)\right)\right\|^2 \tag{13}$$

where $\mathbf{u}_i^p$ is the eigenvector corresponding to the p-th largest eigenvalue of $\mathbf{C}(S_i)$.

3.3 Summary

The training algorithm can be summarized as follows:

1. For each training sample, compute the mean vector $\boldsymbol{\mu}(H_i)$, the covariance matrix $\mathbf{C}(H_i)$, and the correlation matrix $\mathbf{R}(H_i)$ of its hard neighborhood H_i with equations (**1**), (**2**) and (**3**).
2. Estimate the merging weights w_{ij} either by the distance method or the fuzzy c-means algorithm (see Subsection 3.1).
3. Compute the mean vectors $\boldsymbol{\mu}(S_i)$ and covariance matrices $\mathbf{C}(S_i)$ of each soft cluster S_i following (**6**) and (**8**).
4. Extract the eigenvalues and eigenvectors from $\mathbf{C}(S_i)$ and estimate the dimensionality of the underlying manifold K_i, by means of (**9**).
5. Use (**10**) to calculate $\sigma^2(S_i)$, the noise variance for the discarded directions.

4 Convergence Proof

In this section we prove that our estimator $\hat{p}(\ )$ converges to the true density function *p()* in the limit $N \to \infty$ and $M \to \infty$.

Lemma 1. Every local Gaussian kernel $K_i(\mathbf{x})$ tends to the D-dimensional Dirac delta function $\delta(\mathbf{x} - \boldsymbol{\mu}(S_i))$ as $N \to \infty$ and $M \to \infty$.

Proof. In the limit $N\rightarrow\infty$ and $M\rightarrow\infty$ the clusters H_i and S_i reduce their volume to zero, since the L1-median tends to the mean. This means that $\lambda_i^p \rightarrow 0$ for all i and p, where λ_i^p is the p-th eigenvalue of the covariance matrix $\mathbf{C}(S_i)$. Hence the kernels $K_i(\mathbf{x})$ are confined to a shrinking volume centered at $\mu(S_i)$, because the variances in each direction are λ_i^p, but they continue to integrate to 1. So, we have $K_i(\mathbf{x}) \rightarrow \delta(\mathbf{x}-\mu(S_i))$.

Theorem 1. The expected value of the proposed estimation tends to the true probability density function as $N\rightarrow\infty$ and $M\rightarrow\infty$.

Proof. This theorem is proven from Lemma 1 by following the reasoning in [10].

Theorem 2. The variance of the proposed estimation tends to zero as $N\rightarrow\infty$ and $M\rightarrow\infty$.

Proof. This theorem is proven from Theorem 1 by following the reasoning in [10].

5 Experimental Results

This section shows some experiments we have designed in order to study the quality of the density estimation achieved by our method. We call it MedianDist when the distance weighting is used, and MedianFuzzy when we use fuzzy c-means. Vincent and Bengio's method [15] is referred as MParzen, the original Parzen windows method [12] (with isotropic Gaussian kernels) is called OParzen, and finally the Mixtures of Probabilistic PCA model of Tipping and Bishop [13] is called MPPCA. A standard performance measure is the average negative log likelihood

$$ANLL = -\frac{1}{T}\sum_{i=1}^{T}\log \hat{p}(\mathbf{x}_i) \tag{14}$$

where $\hat{p}()$ is the estimator, and the test dataset is formed by T samples $\mathbf{x}_i$.

5.1 Experiment on 2D Artificial Data

We have considered two artificial 2D datasets. The first dataset consists of a training set of 100 points, a validation set of 100 points and a test set of 10000 points, which are generated from the following distribution of two dimensional *(x,y)* points:

$$x = 0.04t\sin(t) + \varepsilon_x, \quad y = 0.04t\cos(t) + \varepsilon_y \tag{15}$$

where $t \sim U(3,15), \varepsilon_x \sim N(0,0.01), \varepsilon_y \sim N(0,0.01), U(a,b)$ is uniform in the interval *(a,b)* and $N(\mu,\sigma)$ is a normal density. The second dataset is a capital letter 'S'.

We have optimized separately all the parameters of the five competing models with disjoint training and validation sets. The tuning of our parameters α and γ of MedianDist and MedianFuzzy has not needed much computation time, since they are robust against small changes in their values. The performance of the optimized models has been computed by 10-fold cross-validation, and the results are shown in Table 1, with the best result marked in bold. It can be seen that our models outperform the other three in density distribution estimation (note that lower is better).

Table 1. Quantitative results on the artificial datasets (standard deviations in parentheses)

Method	ANLL on test set (spiral)	ANLL on test set (capital 'S')
MedianDist	**–1.7781 (1.1521)**	**–2.0452 (0.4281)**
MedianFuzzy	–1.6891 (0.8712)	–1.9950 (0.5781)
OParzen	1.0817 (1.3357)	–0.6929 (0.4003)
MParzen	–0.9505 (0.3301)	–1.0956 (0.0657)
MPPCA	0.2473 (0.0818)	–0.1751 (0.6204)

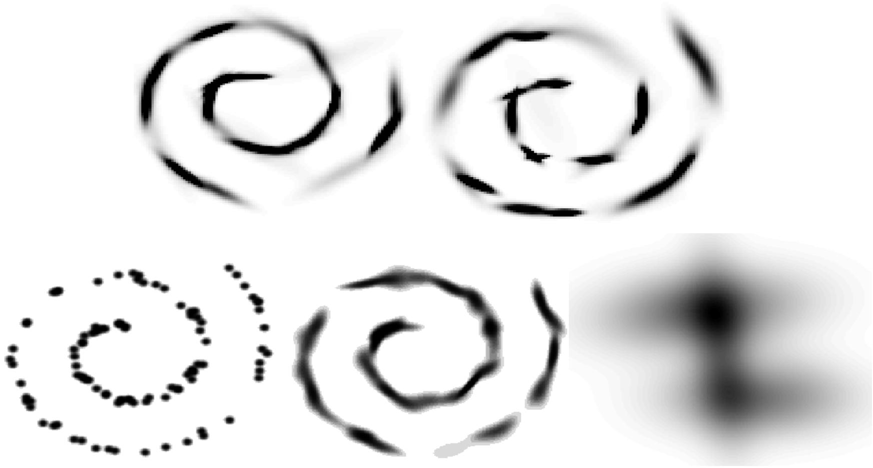


Fig. 1. Density estimation for the 2D spiral dataset. From left to right and from top to bottom: MedianDist, MedianFuzzy, OParzen, MParzen and MPPCA.

Figures 1 and 2 show density distribution plots corresponding to the five models. Darker areas represent zones with high density mass and lighter ones indicate the estimator has detected a low density area.

We can see in the plots that our models have less density holes (light areas) and less 'bumpiness'. This means that our model represents more accurately the true distribution, which has no holes and is completely smooth. We can see that the quantitative ANLL results agree with the plots, because the lowest values of ANLL match the best-looking plots. So, our model outperforms clearly the other three considered approaches.

5.2 Classification Experiment

We have selected 7 classification benchmarks from the UCI Repository of Machine Learning Databases [11] to perform a classification experiment. We have considered Bayesian classifiers which are built by estimating the probability density function of each class separately. Then, a test pattern is assigned to the class which yields the largest probability density (converts a soft classification to crisp). We have optimized the model parameters separately, as in the previous experiment. In this case we have optimized the models independently for each class of each database. The results of the 10-fold cross-validation are shown in Table 2, and the winning model for each database is shown in bold. We can see that our two approaches outperform the other three in most databases. As before, we have optimized all the parameters of the five competing models with disjoint training and validation sets. The parameters for the density estimator of each dataset have been optimized separately.

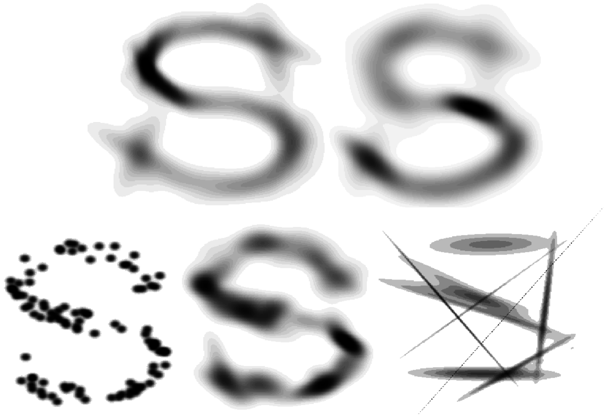


Fig. 2. Density estimation for the 2D capital 'S' dataset. From left to right and from top to bottom: MedianDist, MedianFuzzy, OParzen, MParzen and MPPCA.

Table 2. Probability of successful classification. The standard deviations are shown in parentheses.

Database	MedianDist	MedianFuzzy	OrigParzen	ManifParzen	MPPCA
BalanceScale	**0.892 (0.043)**	0.887 (0.043)	0.730 (0.040)	0.773 (0.046)	0.735 (0.040)
Contraceptive	0.337 (0.039)	**0.497 (0.037)**	0.453 (0.029)	0.481 (0.039)	0.488 (0.029)
Ionosphere	0.881 (0.070)	**0.914 (0.036)**	0.899 (0.056)	0.895 (0.053)	0.897 (0.044)
Liver	**0.686 (0.044)**	0.590 (0.056)	0.606 (0.079)	0.626 (0.078)	0.592 (0.109)
Pendigits	0.957 (0.014)	**0.980 (0.003)**	0.975 (0.004)	0.973 (0.006)	0.977 (0.002)
Pima	0.651 (0.035)	**0.672 (0.033)**	0.660 (0.039)	0.620 (0.049)	0.629 (0.056)
Wine	0.831 (0.077)	0.791 (0.073)	0.656 (0.117)	0.668 (0.079)	**0.863 (0.104)**

6 Conclusions

We have presented a probability density estimation model. It is based in the Parzen window approach. Our proposal uses the L1-median to build local models for a hard neighborhood of each training sample. Then soft clusters are obtained by merging these local models. Computational results show the superior performance of our method in density estimation and classification applications.

Acknowledgements

This work is partially supported by Junta de Andalucía (Spain) under contract TIC-01615, project name Intelligent Remote Sensing Systems.

References

1. Bezdek, J.C.: Numerical taxonomy with fuzzysets. J. Math. Biol. 1, 57–71 (1974)
2. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
3. Dang, X., Serfling, R., Zhou, W.: Influence functions of some depth functions and application to depth-weighted L-statistics. Journal of Nonparametric Statistics 21(1), 49–66 (2009)
4. Gervini, D.: Robust functional estimation using the median and spherical principal components. Biometrika 95(3), 587–600 (2008)
5. Hobza, T., Pardo, L., Vajda, I.: Robust median estimator in logistic regression. Journal of Statistical Planning and Inference 138(12), 3822–3840 (2008)
6. Hjort, N.L., Jones, M.C.: Locally Parametric Nonparametric Density Estimation. Annals of Statistics 24(4), 1619–1647 (1996)
7. Hössjer, O., Croux, C.: Generalizing Univariate Signed Rank Statistics for Testing and Estimating a Multivariate Location Parameter. Non-parametric Statistics 4, 293–308 (1995)
8. Izenman, A.J.: Recent developments in nonparametric density estimation. Journal of the American Statistical Association 86(413), 205–224 (1991)
9. Lejeune, M., Sarda, P.: Smooth estimators of distribution and density functions. Computational Statistics & Data Analysis 14, 457–471 (1992)
10. López-Rubio, E., Ortiz-de-Lazcano-Lobato, J.M.: Soft clustering for nonparametric probability density function estimation. Pattern Recognition Letters 29, 2085–2091 (2008)
11. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Department of Information and Computer Science, Irvine, CA (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
12. Parzen, E.: On the Estimation of a Probability Density Function and Mode. Annals of Mathematical Statistics 33, 1065–1076 (1962)
13. Tipping, M.E., Bishop, C.M.: Mixtures of Probabilistic Principal Components Analyzers. Neural Computation 11, 443–482 (1999)
14. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
15. Vincent, P., Bengio, Y.: Manifold Parzen Windows. Advances in Neural Information Processing Systems 15, 825–832 (2003)
16. Zaman, A., Rousseeuw, P.J., Orhan, M.: Econometric applications of high-breakdown robust regression techniques. Economics Letters 71, 1–8 (2001)

An Awareness-Based Artificial Neural Network for Cooperative Distributed Environments

Mauricio Paletta[1] and Pilar Herrero[2]

[1] Centro de Investigación en Informática y Tecnología de la Computación (CITEC).
Universidad Nacional de Guayana (UNEG), Av. Atlántico, Ciudad Guayana, 8050, Venezuela
mpaletta@uneg.edu.ve
[2] Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo S/N.
28.660 Boadilla del Monte, Madrid, Spain
pherrero@fi.upm.es

Abstract. Cooperative tasks to be successful in Collaborative Distributed Environments (CDE) require from users to be known. Before hard, which users are more suitable in the system to cooperate with, as well as which tools are needed to achieve the common goal in the system in a cooperative way. On the other hand, awareness allows users to be aware of others' activities each and every moment. Information about others' activities combined with their intentions and purposes could be used to improve cooperation in CDEs. This paper focuses on associating the concept of awareness with vector quantization techniques. It presents a novel non-supervised Artificial Neural Network (ANN) based model for CDE named CAwANN, which uses the information of awareness collaborations occurring in the environment in order to achieve the most suitable awareness-based collaboration / cooperation in the environment.

Keywords: Awareness, collaboration, distributed environment, artificial neural network.

1 Introduction

Collaborative Distributed Environments (CDEs) are those in which multiple users in remote locations, usually agents, participate in shared activity aiming to achieve a common goal. The success of achieving this goal in proper time and/or to obtain the higher quality of results in these dynamic and distributed environments depends on implementing an appropriate collaboration by means of the most suitable mechanism.

On the other hand, according to CSCW (Computer Supported Cooperative Work) awareness is a useful concept used to achieve cooperation and collaboration in CDE as it increases communication opportunities [13]. In this matter, a collaborative process is leaded by five processes [9, 11]: co-presence, awareness, communication, collaboration, and coordination. Moreover, in CSCL (Computer Supported Collaborative Learning), awareness plays an important role as it promotes collaboration opportunities in a natural and efficient way [19] and it improves effectiveness of collaborative learning. In this matter, Gutwin et al identified the following types of awareness [5]: social, task, concept, workspace, and knowledge.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 114–121, 2009.
© Springer-Verlag Berlin Heidelberg 2009

By considering the awareness definition previously given as well as associating this concept with Vector Quantization (VQ) techniques [8, 10, 16, 17], this paper presents a novel non-supervised Artificial Neural Network (ANN) model for CDEs named CAwANN. It takes into account the information of awareness collaborations occurring in the environment for achieving the most appropriate future awareness situations. The paper also presents the way in which a Neural-Gas network (NGAS) [12] is used to learn different types of awareness and therefore collaborations.

The rest of the paper is organized as follows. Section 2 presents some existing related work in the area. Background aspects of VQ and NGAS are showed in section 3. Section 4 presents the CAwANN model proposed in this paper. Details of implementation and results of experimental tests are showed in section 4. Finally, section 5 exposes paper conclusions and the future work related with this research.

2 Related Work

Researches in CSCW have already proposed awareness to: 1) give information on the surroundings of the target user [2]; 2) provide common or public space where users can gather and meet [3]; 3) simulate informal communicative opportunities in real world using computers [14]. Regarding the context of awareness and recognizing the current context of a user or device, authors in [15] present an approach based on general and heuristic extensions to the growing NGAS algorithm classifier which allow its direct application for context recognition.

On the other hand, an approach for the optimization of the job scheduling in large distributed systems, based on a self-organizing neural network is presented in [18]. In this approach, the dynamic scheduling system should be seen as adaptive middle layer software, aware of current available resources and making the scheduling decisions using the past experience. Authors in [1] present an architecture aiming to address the problem in a collaborative learning activity to create groups among students. They used a neural network algorithm to obtain homogenous groups.

As far as we know, there is not any non-supervised based model for learning cooperation on CDE by using the awareness information originated in this environment. Next section presents backgrounds about VQ and NGAS technologies.

3 Vector Quantization and Neural-Gas

Vector Quantization (VQ) is the process of quantizing n-dimensional input vectors to a limited set of n-dimensional output vectors referred to as *code-vectors*. The set of possible *code-vectors* is called the *codebook*. The *codebook* is usually generated by clustering a given set of training vectors (called *training set*). Clustering can be described then, as the process of organizing the *codebook* into groups whose members share similar features in some way.

Neural-Gas (NGAS) is a VQ technique with soft competition between the units. In each training step, the squared Euclidean distances between a randomly selected input vector x_i from the training set and all *code-vectors* m_k are computed; the vector of these distances, expressed in (3.1) is d. Each centre k is assigned a rank $r_k(d) = 0, \ldots,$

N-1, where a rank of 0 indicates the closest and a rank of N-1 the most distant centre to x. The learning rule is expressed as it is indicated in (3.2).

$$d_{ik} = \|x_i - m_k\| = (x_i - m_k)^T * (x_i - m_k) \tag{3.1}$$

$$m_k = m_k + \varepsilon * h_\rho [r_k(d)] * (x - m_k) \tag{3.2}$$

$$h_\rho(r) = e^{(-r/\rho)} \tag{3.3}$$

A monotonically decreasing function of the ranking that adapts all the centers, with a factor exponentially decreasing with their rank is represented in (3.3). The width of this influence is determined by the neighborhood range ρ. Learning rule is also affected by a global learning rate ε. Values of ρ and ε decrease exponentially from an initial value ($\rho(0)$, $\varepsilon(0)$) to a smaller final value ($\rho(T)$, $\varepsilon(T)$) according to expressions (3.4) and (3.5) respectively; t is the time step and T the total number of training steps.

$$\rho(t) = \rho(0) * [\rho(T)/\rho(0)]^{(t/T)} \tag{3.4}$$

$$\varepsilon(t) = \varepsilon(0) * [\varepsilon(T)/\varepsilon(0)]^{(t/T)} \tag{3.5}$$

4 CAwANN: CDE by Means of an Awareness and ANN Model

4.1 Awareness Representation

CAwANN has been designed by taking into account the key awareness concepts proposed by Herrero et al in [6, 7]. Given a collaborative distributed environment E containing a set of n nodes N_i ($1 \leq i \leq n$), where r items can be shared as a collaborative mechanism to be shared among different nodes. In CAwANN these items are called resources R_j ($1 \leq j \leq r$). The distinction among different resources and the particular association of these awareness concepts for each resource is the key to distinguish the types of collaboration that exist in the CDE. We have:

1) N_i.*Focus*(R_j): It can be interpreted as the subset of the space (environment/ medium) on which the user (agent) in N_i has focused his attention aiming of interaction/collaboration with it, according to the resource (collaboration item) R_j.

2) N_i.*NimbusState*(R_j): Indicates the current grade of collaboration N_i can give over R_i. It could have three possible values: *Null*, *Medium* or *Maximum*. If the current grade of collaboration N_i that is given about R_j is not high, and N_i could collaborate more over R_j, then N_i.*NimbusState*(R_j) will get the *Maximum* value. If the current grade is high but N_i could collaborate a little more, then N_i.*NimbusState*(R_j) would be *Medium*. Finally, if N_i is overload, its *NimbusState*(R_j) would be *Null*.

3) N_i.*NimbusSpace*(R_j): The subset of the space where N_i is currently collaborating with R_j.

4) R_j.*AwareInt*(N_a, N_b): This concept quantifies the degree of collaboration over R_j between a pair of nodes N_a and N_b. Following the awareness classification introduced by Greenhalgh in [4], it could be Full, Peripheral or Null according to (4.1).

$$R_j.AwareInt(N_a, N_b) = \begin{cases} N_b \in N_a.Focus(R_j) \wedge N_a \in N_b.Nimbus(R_j), \mathit{Full} \\ (N_b \in N_a.Focus(R_j) \wedge N_a \notin N_b.Nimbus(R_j)) \vee \\ (N_b \notin N_a.Focus(R_j) \wedge N_a \in N_b.Nimbus(R_j)), \mathit{Peripheral} \\ \text{Otherwise, } \mathit{Null} \end{cases} \tag{4.1}$$

5) N_i.*RequiredTask*(R_j): Indicates that N_i requires collaboration with R_j.

6) N_i.*TaskResolution*(R_j): Determines the score needed for R_j to collaborate in N_i. It is represented with a value within [0, 1]. The higher the value is to 1 the higher is the complete willingness to collaborate the R_j necessity.

Based on the concepts previously mentioned, the basic idea is to solve any collaborative requested action given by N_i.*RequiredTask*(R_j) according to the current state of the CDE given by N_i.*Focus*(R_j) and N_i.*Nimbus* (R_j), and therefore following the awareness conditions given by R_j.*AwareInt*(N_a, N_b). To do this it is necessary to obtain the N_i.*TaskResolution*(R_j) for each node associated to R_j, and therefore to identify the most suitable node to be collaborating with.

Also, it is important to emphasize that learning the association (CDE, *TaskResolution*) for any different resource, represents the apprenticeship of different types of awareness, and therefore collaborations occurring in the CDE.

4.2 Learning Awareness in CDE with CAwANN

Collaborative learning can be defined as the process to learn from the association between the situational environment (*E*) is in a given moment, and the collaboration scores for this specific situation (N_a.*TaskResolution*(R_j) for each node). The final goal is to identify those nodes (N_b) with which a given a node (N_a) could collaborate with, based on CDE current conditions and the N_b.*RequiredTask*(R_j) requirements.

However, the calculation of the N_a.*TaskResolution*(R_j) depends on several factors like the awareness type, the collaboration strategy, the number of nodes, and so on, and therefore, these factors make very difficult to define a supervised learning strategy to deal with this problem (an example of a supervised learning strategy related with the same problem can be consulted in [20]). CAwANN makes use of past experiences in the CDE to create a dynamic decision making scheme. A competitive learning algorithm is used to cluster correlated information in the multi-dimensional input space defined by the parameters describing the CDE.

The goal of this learning process is 1) to cluster the input data into a set of partitions such that the intra-cluster variance which remains small compared with the inter-cluster variance, and 2) to estimate the probability density function. This clustering scheme seems possible as we expect a strong correlation among the awareness information involved. The processes of clustering neurons during the learning may be compared with the creation of molecules in a gas. Close neurons are connected, and the update procedure for one neuron will also have an influence on the neighbors. A neuron connection also has an age, and if the two neurons connected are not the closest for a large number of data samples, such a connection will be removed.

The decision about which nodes should collaborate in future task, should be based on identifying the clusters in the total parameter space, identifying which are close to the hyper plane defined in the space by means of the current CDE conditions (i.e., the winner unit/node).

The input vector is defined as follows. Being N_b the node who requires collaboration on R_j and therefore who sends the $N_b.\mathit{RequiredTask}(R_j)$, for each $N_a \neq N_b$ we have: 1) A value $Nst \in [0,1]$ representing $N_a.\mathit{NimbusState}(R_j)$; $Nst = 1 \Leftrightarrow N_a.\mathit{NimbusState}(R_j) = \mathit{Maximum}$, $Nst = 0.5 \Leftrightarrow N_a.\mathit{NimbusState}(R_j) = \mathit{Medium}$, or $Nst = 0 \Leftrightarrow N_a.\mathit{NimbusState}(R_j) = \mathit{Null}$; 2) A value $AwI \in [0,1]$ representing $R_j.\mathit{AwareInt}(N_a, N_b)$; $AwI = 1 \Leftrightarrow R_j.\mathit{AwareInt}(N_a, N_b) = \mathit{Full}$, $AwI = 0.5 \Leftrightarrow R_j.\mathit{AwareInt}(N_a, N_b) = \mathit{Peripheral}$, or $AwI = 0 \Leftrightarrow R_j.\mathit{AwareInt}(N_a, N_b) = \mathit{Null}$.

Therefore, the *code-vectors* for this problem have $2n$ elements. If $N_a = N_b$ then $Nst = AwI = 0$. Patterns for learning are obtained by actual scenarios that have been stored during the dynamics of the CDE.

4.3 Collaboration Process by Using CAwANN

Any node N_a in the CDE is represented by agent who has the corresponding required information about E, i.e.: $N_a.\mathit{Focus}(R_j)$, $N_a.\mathit{NimbusState}(R_j)$ and $N_a.\mathit{NimbusSpace}(R_j)$ for R_j. All these agents are communicated among each other. When there is a collaborative request from any node N_b, a $N_b.\mathit{RequiredTask}(R_j)$ is sent by N_b and it is received by the rest of agents (nodes) in the environment. $N_b.\mathit{Focus}(R_j)$, and $N_b.\mathit{NimbusSpace}(R_j)$ are included as parameters in the message sent.

Once the requested info ($N_b.\mathit{RequiredTask}(R_j)$ and parameters) is received, N_a replies to N_b the corresponding $R_j.\mathit{AwareInt}(N_a, N_b)$ and $N_a.\mathit{NimbusState}(R_j)$. After that, CAwANN is used for N_b to decide the better node to collaborate with it. Fig. 1 shows a general outline of this collaboration process.

Next section presents some aspects related with the evaluation of the CAwANN model proposed in this paper and previously defined.

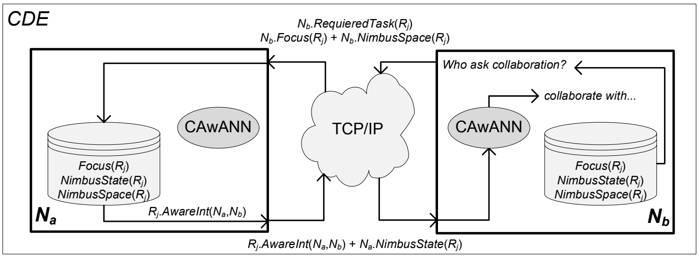


Fig. 1. General outline of the collaboration process using CAwANN

5 Evaluation

CAwANN was implemented using Java. The evaluation of the model was conducted in a PC with the following hardware platform: Intel T2600 (2.16 GHz) with 2 GB RAM. It was carried out by considering the following aspects:

1) The NGAS network was trained with $T = 40.000$ signals (size of the *codebook*).

2) The signals for the training process were generated by creating different CDE conditions, i.e. to define $N_a.\mathit{Focus}(R_j)$, $N_a.\mathit{NimbusState}(R_j)$ and $N_a.\mathit{NimbusSpace}(R_j)$ for each N_a ($1 \leq a \leq n$) as well as to define a $N_b.\mathit{RequiredTask}(R_j)$ in a randomly way.

3) The parameters used for configuring the ANN are the following: $\varepsilon(0) = 1.58$; $\varepsilon(T) = 0.02$; $\rho(0) = 5.59$; $\rho(T) = 0.07$. In fact, a genetic program was used to find the best configuration to deal with this problem.

4) After the training, the network was being tested with another 500 signals randomly defined.

5) With the aim of determining the efficiency of the model, i.e. if the winner unit N_w given by the NGAS network is the most suitable node to collaborate with, based on the CDE current conditions, it was necessary to define a deterministic and hypothetic expression for calculating the $N_a.TaskResolution(R_j)$. In this regard, the expression (5.1) highlight awareness (75 %) compared with the current state of the node.

6) From the CDE scenario as well as the $N_a.TaskResolution(R_j)$ scores calculated by using (5.1), the most suitable node to collaborate with (hypothetical case) was identified by finding the node N_h with the higher score (more than one node can have the same high score). A success ξ occurs when N_w and N_h are the same node.

7) If N_w is equal to N_b (the node who required collaboration) then the answer was considered as a fault τ.

8) Efficiency of the model was calculated then by considering the average of the total of success excluding the number of faults, i.e. $(\xi - \tau) / 100$.

9) The experiment was conducted with different CDE scenarios by changing the quantity of nodes from 3 to 18 varying in 3 nodes for each case. Fig. 5.1 shows the corresponding efficiency calculated according to (5.1) for each of these cases. Results indicate that there is an average efficiency of 46.13%. It was hoped that the increased efficiency was achieved with fewer nodes and vice versa.

$$N_a.TaskResolution(R_j) = 0.25 * N_a.NimbusState(R_j) + 0.75 * R_j.AwareInt(N_a, N_b) \qquad (5.1)$$

Below an example of the evaluation process with $n = 8$. The CDE condition, given by $N_a.Focus(R_j)$, $N_a.NimbusState(R_j)$ and $N_a.NimbusSpace(R_j)$ is as follows:

$N_0.Focus(R_j) = \{N_0, N_1, N_3, N_5\} = [1, 1, 0, 1, 0, 1, 0, 0]$
$N_1.Focus(R_j) = \{N_1, N_2, N_4, N_5, N_6, N_7\} = [0, 1, 1, 0, 1, 1, 1, 1]$
$N_2.Focus(R_j) = \{N_1, N_2, N_3, N_4, N_5, N_6, N_7\} = [0, 1, 1, 1, 1, 1, 1, 1]$
$N_3.Focus(R_j) = \{N_0, N_1, N_2, N_3, N_5, N_6, N_7\} = [1, 1, 1, 1, 0, 1, 1, 1]$
$N_4.Focus(R_j) = \{N_0, N_1, N_2, N_4, N_5, N_6, N_7\} = [1, 1, 1, 0, 1, 1, 1, 1]$
$N_5.Focus(R_j) = \{N_0, N_1, N_2, N_3, N_4, N_5, N_6, N_7\} = [1, 1, 1, 1, 1, 1, 1, 1]$
$N_6.Focus(R_j) = \{N_0, N_2, N_3, N_5, N_6, N_7\} = [1, 0, 1, 1, 0, 1, 1, 1]$
$N_7.Focus(R_j) = \{N_0, N_1, N_2, N_3, N_4, N_5, N_6, N_7\} = [1, 1, 1, 1, 1, 1, 1, 1]$
$N_0.NimbusSpace(R_j) = \{N_0, N_1, N_3, N_5\} = [1, 1, 0, 1, 0, 1, 0, 0]$
$N_1.NimbusSpace(R_j) = \{N_1, N_5, N_7\} = [0, 1, 0, 0, 0, 1, 0, 1]$
$N_2.NimbusSpace(R_j) = \{N_2, N_3, N_4, N_5, N_6, N_7\} = [0, 0, 1, 1, 1, 1, 1, 1]$
$N_3.NimbusSpace(R_j) = \{N_1, N_2, N_3, N_5, N_6, N_7\} = [0, 1, 1, 1, 0, 1, 1, 1]$
$N_4.NimbusSpace(R_j) = \{N_0, N_1, N_5\} = [1, 1, 0, 0, 0, 1, 0, 0]$
$N_5.NimbusSpace(R_j) = \{N_0, N_2, N_3, N_6, N_7\} = [1, 0, 1, 1, 0, 0, 1, 1]$
$N_6.NimbusSpace(R_j) = \{N_0, N_2, N_5, N_6, N_7\} = [1, 0, 1, 0, 0, 1, 1, 1]$
$N_7.NimbusSpace(R_j) = \{N_0, N_1, N_3, N_7\} = [1, 1, 0, 1, 0, 0, 0, 1]$
$N_0.NimbusState(R_j) = Medium = 0.5$; $N_1.NimbusState(R_j) = Maximum = 1$
$N_2.NimbusState(R_j) = Maximum = 1$; $N_3.NimbusState(R_j) = Maximum = 1$
$N_4.NimbusState(R_j) = Maximum = 1$; $N_5.NimbusState(R_j) = Null = 0$
$N_6.NimbusState(R_j) = Medium = 0.5$; $N_7.NimbusState(R_j) = Maximum = 1$

By considering the previous CDE as well as the N_2.*RequiredTask*(R_j) (b = 2) necessity of collaboration with N_2, the corresponding *code-vector* or signal/pattern [...[Nst_i, AwI_i]...] ($1 \leq i \leq n$) used for training is as follows (see Section 4.2):

[[0.5, 0], [1, 0.5], [0, 0], [1, 1], [1, 1], [0, 1], [0.5, 1], [1, 1]]

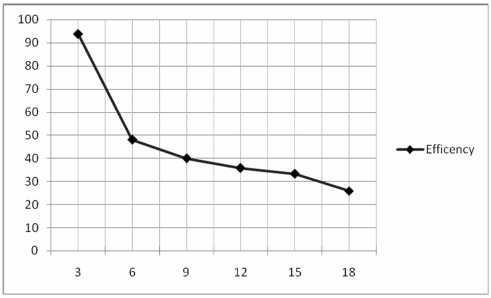


Fig. 2. Efficiency versus CDE scenarios by changing the quantity of nodes

6 Conclusions and Ongoing Work

Theoretical aspects of awareness are used in this paper to define a non-supervised learning model to improve cooperation/collaboration in distributed collaborative environments. This paper presents a novel non-supervised model named CAwANN, which has been designed from the beginning to learn from different types of awareness as well as from previous collaborations that were carried out in the environment to foresee future collaborative/cooperative scenarios. The results obtained show that the proposed model has an average efficiency of 46.13%.

Although this method has not been tested yet in a real Collaborative Distributed Environment (CDE), it has been designed to be suitable for real environments. In fact the experimentation and validation carried out to present demonstrate that this model could be extended to real scenarios in CDE with no problems.

We are currently working on testing this method in real CDE. We are also working in combining non supervised and supervised strategies aiming to improve the quality of this learning process.

References

1. Blanchard, E., Frasson, C.: Designing a Multi-agent Architecture based on Clustering for Collaborative Learning Sessions. In: Proc. International Conference in Intelligent Tutoring Systems. LNCS, Springer, Heidelberg (2002), `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.4799`
2. Dourish, P., Bly, S.: Portholes: Supporting awareness in a Distributed Work Group. In: Proc. Conference on Human Factors in Computing Systems (CHI 1992), pp. 541–548 (1992)
3. Fish, R., Kraut, R., Chalfonte, B.: The Video Window System in Informal Communications. In: Proc. Computer Supported Cooperative Work (CSCW 1990), pp. 1–12 (1990)

4. Greenhalgh, C.: Large Scale Collaborative Virtual Environments, Doctoral Thesis, University of Nottingham (1997)
5. Gutwin, C., Stark, G., Greenberg, S.: Support for Workspace Awareness in Educational Groupware. In: Proc. Computer Supported Collaborative Learning (CSCL 1995), pp. 147–156 (1995)
6. Herrero, P., Bosque, J.L., S. Pérez, M.: An Agents-Based Cooperative Awareness Model to Cover Load Balancing Delivery in Grid Environments. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 64–74. Springer, Heidelberg (2007)
7. Herrero, P., Bosque, J.L., Pérez, M.S.: Managing Dynamic Virtual Organizations to Get Effective Cooperation in Collaborative Grid Environments. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part II. LNCS, vol. 4804, pp. 1435–1452. Springer, Heidelberg (2007)
8. Kohonen, T., Mäkisara, K., Saramäki, T.: Phonotopic maps – insightful representation of phonological features for speech recognition. In: Proc. 7th International Conference on Pattern Recognition, pp. 182–185 (1984)
9. Kuwana, E., Horikawa, K.: Coordination Process Model - based Design for Synchronous Group Task Support System, Technical Reports of Information Processing Society of Japan, Groupware, No. 13, 1-6 (1995)
10. Makhoul, J., Roucos, S., Gish, H.: Vector quantization in speech coding. Proc. IEEE 73, 1551–1588 (1985)
11. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. ACM Computing Surveys 26(1), 87–119 (1994)
12. Martinetz, T.M., Schulten, K.J.: A neural-gas network learns topologies. In: Kohonen, T., Mäkisara, K., Simula, O., Kangas, J. (eds.) Artificial Neural Networks, pp. 397–402 (1991)
13. Matsushita, Y., Okada, K. (eds.): Collaboration and Communication. Distributed collaborative media series, vol. 3. Kyoritsu Press (1995)
14. Matsuura, N., Hidaka, T., Okada, K., Matsushita, Y.: VENUS: An informal Communication Environment supporting Interest Awareness. Trans. Information Processing Society of Japan 36(6), 1332–1341 (1995)
15. Mayrhofer, R., Radi, H.: Extending the Growing Neural Gas Classifier for Context Recognition. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2007. LNCS, vol. 4739, pp. 920–927. Springer, Heidelberg (2007)
16. Nasrabadi, N.M., Feng, Y.: Vector quantization of images based upon the Kohonen self-organizing feature maps. In: IEEE International Conference on Neural Networks, pp. 1101–1108 (1988)
17. Naylor, J., Li, K.P.: Analysis of a Neural Network Algorithm for vector quantization of speech parameters. In: Proc. First Annual INNS Meeting, pp. 310–315. Pergamon Press, NY (1988)
18. Newman, H.B., Legrand, I.C.: A self-organizing neural network for job scheduling in distributed systems. In: Advanced Computing and Analysis Techniques in Physics Research: VII International Workshop; ACAT 2000. AIP Conference Proc., vol. 583, pp. 113–115 (2000)
19. Ogata, H., Yano, Y.: Knowledge Awareness: Bridging Learners in a Collaborative Learning Environment. International Journal of Educational Telecommunications 4(2), 219–236 (1998)
20. Paletta, M., Herrero, P.: Learning Cooperation in Collaborative Grid Environments to Improve Cover Load Balancing Delivery. In: Proc. IEEE/WIC/ACM Joint Conferences on Web Intelligence and Intelligent Agent Technology, vol. E3496, pp. 399–402. IEEE Computer Society, Los Alamitos (2008)

Improving Classification under Changes in Class and Within-Class Distributions[⋆]

Rocío Alaiz-Rodríguez[1], Alicia Guerrero-Curieses[2], and Jesús Cid-Sueiro[3]

[1] Universidad de León, Dpto. de Ingeniería Eléctrica y de Sistemas,
Campus de Vegazana, 24071 León, Spain
`rocio.alaiz@unileon.es`
[2] Universidad Rey Juan Carlos. Dpto. de Teoría de la Señal y Comunicaciones, 28943
Fuenlabrada-Madrid, Spain
`alicia.guerrero@urjc.es`
[3] Universidad Carlos III de Madrid. Dpto. de Teoría de la Señal y Comunicaciones,
Leganés-Madrid, Spain
`jcid@tsc.uc3m.es`

Abstract. The fundamental assumption that training and operational data come from the same probability distribution, which is the basis of most learning algorithms, is often not satisfied in practice. Several algorithms have been proposed to cope with classification problems where the class priors may change after training, but they can show a poor performance when the class conditional data densities also change. In this paper, we propose a re-estimation algorithm that makes use of unlabeled operational data to adapt the classifier behavior to changing scenarios. We assume that (a) the classes may be decomposed in several (unknown) subclasses, and (b) the prior subclass probabilities may change after training. Experimental results with practical applications show an improvement over an adaptive method based on class priors, while preserving a similar performance when there are no subclass changes.

1 Introduction: Imprecise Operating Environment

A basic assumption of the supervised classification paradigm is that the data used to design the classifier are representative of the future real-world distribution from which new data will be drawn. However, in most real classification problems, such as credit scoring, direct mail marketing or land-cover mapping [6,5], these assumptions rarely fulfill, either because the environment is not (space- or time-) stationary or because of the costs of data collection/labeling are class-dependent. Thus, any mismatch between operational ("real-world") data and training data will result in a sub-optimal performance.

The discrepancy between learning and operating conditions has been addressed in the supervised classification literature from different perspectives, depending on the nature of the discrepancy. We can categorize two interesting types of distribution changes analyzed in classification scenarios: Situations where the input data distribution $p(\mathbf{x})$ varies from training to test, but follow

---

[⋆] Supported by the Spanish MEC projects DPI2006-02550 and TEC2008-01348/TEC.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 122–130, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the same conditional distribution $p(\mathbf{y}|\mathbf{x})$, what is commonly referred as covariate shift [8,10]; and scenarios, where class prior probability $p(\mathbf{y})$ varies from training to test, but conditional density function $p(\mathbf{x}|\mathbf{y})$ does not change [4,9]. This paper focuses on a generalization of this latter problem which relaxes the basic assumption (class prior changes) to changes in subclass priors.

Differences in operating conditions imputed to changes in prior class probabilities have been tackled in different ways depending on the knowledge attributed to the end user. If the end user is expected to know the new priors (uncertain at the learning stage), the use of ROC curves [6] is shown to be more robust than traditional metrics. For the case of unknown new priors, several techniques have been proposed, guided by robustness, such as the minimax approaches [1], or by adaptivity, such as those based on re-estimating the priors [7].

The approaches that deal with the changes in class distributions, usually assume that the true and false positive rates will remain invariant under varying class distributions, or equivalently, that the class conditional density functions do not change. This is commonly referred in the literature as no "concept drift". [4]. However, in a recent study [9], the authors discuss common circumstances in real-world applications under which this assumption is not satisfied , but they do not provide any solution and consider it as an open problem. [4,5,9].

In this paper, we propose an adaptive strategy to deal with changes in data distributions that is based on the assumption that (a) each class can be decomposed in several subclasses, which is a natural assumption in many practical problems [9], and (b) that operating data may be affected by changes in prior subclass probabilities. In this way, we provide the classifier a higher flexibility to cope with changes in the environment where the classifier is deployed, while preserving invariance in subclass conditional densities. Our simulation work shows that adaptive approaches based on changes in subclass priors can be more efficient than other approaches assuming changes in class priors.

Our proposal is a neural-based approach able to adjust the classifier, without retraining it, to unknown conditions at application time. We show how to extend the adaptive approach presented in [7] to a more general scenario, with possible subclass distribution changes that lead to a concept drift. The new operation conditions are re-estimated by using the EM algorithm and a model that provides estimates of class and subclass posterior probabilities, in a supervised and unsupervised way, respectively.

In this paper, Sect. 2 presents the adaptive re-estimation approach and a classifier that provides class and subclass posterior probability estimates. Sect. 3 reports experimental results and the main conclusions are summarized in Sect. 4.

2 Re-estimation of New Operating Conditions

We address the problem of training a classifier by using a set of observations statistically related with a hierarchy of classes and subclasses, and by taking into account that (1) only class labels are observed, (2) the number of subclasses per class is unknown, and (3) training and test sets may differ in prior subclass probabilities as well as in prior class probabilities.

To be more precise, we will formulate the problem mathematically: we denote each observation as a vector, $\mathbf{x} \in \mathbf{R}^N$, and each class label as an L-dimensional vector $\mathbf{d} \in U_L = \{\mathbf{u}_0, \ldots, \mathbf{u}_{L-1}\}$, where L is the number of classes. For mathematical convenience, class-i label $\mathbf{u}_i$ is a unit vector with components $u_{ij} = \delta_{i-j}$ (i.e. every component is equal to 0, except $u_{ii} = 1$). In addition, we assume that each class is partitioned into several *subclasses*: let M_i be the number of subclasses from class i. The subclass for a given sample $\mathbf{x}$ will be represented by a vector, $\mathbf{z}$, with components $z_{ij} \in \{0, 1\}$, $i = 0, \ldots, L-1$, $j = 1, \ldots, M_i$, such that one and only one of them is equal to 1 and[1]

$$d_i = \sum_{j=1}^{M_i} z_{ij}, \qquad 0 \leq i \leq L-1 \ . \tag{1}$$

We assume that training and test sets are given by $S_t = \{(\mathbf{x}^k, \mathbf{d}^k), k = 1, \ldots, N_t\}$ and $S'_t = \{(\mathbf{x}'^k, \mathbf{d}'^k), k = 1, \ldots, N'_t\}$, where samples in S_t have been generated according to an unknown probability model $p(\mathbf{x}, \mathbf{d}|\mathbf{\Pi})$ and samples in S'_t arise from $p(\mathbf{x}, \mathbf{d}|\mathbf{\Pi}')$, where $\mathbf{\Pi}$ (with components $\pi_{ij} \in [0, 1]$, $i = 0, \ldots, L-1$, $j = 1, \ldots, M_i$) and $\mathbf{\Pi}'$ (with components π'_{ij}) are the prior subclass probability vectors, so that

$$\begin{aligned} P\{z_{ij} = 1|\mathbf{\Pi}\} &= \pi_{ij} \ , \\ P\{z_{ij} = 1|\mathbf{\Pi}'\} &= \pi'_{ij} \ . \end{aligned} \tag{2}$$

Additionally, we assume that labels in S'_t are not observed and only the unlabeled set, $S'_u = \{\mathbf{x}'^k, k = 1, \ldots, N'_t\}$, is known. Our goal is to estimate the posterior class probabilities $P\{\mathbf{d} = \mathbf{u}_i|\mathbf{x}, \mathbf{\Pi}'\}$ using samples in S_t and S'_u in order to minimize the decision error rates on the test set.

2.1 Adjusting Classifier Outputs for New Priors

Our approach to this problem is based on the estimation of posterior subclass probabilities, $P\{z_{ij} = 1|\mathbf{x}, \mathbf{\Pi}'\}$ so that, using Eq. (1), we can write

$$P\{\mathbf{d} = \mathbf{u}_i|\mathbf{x}, \mathbf{\Pi}'\} = \sum_{j=1}^{M_i} P\{z_{ij} = 1|\mathbf{x}, \mathbf{\Pi}'\} \tag{3}$$

Note that the training set has no subclass labels, and thus, in order to estimate posterior subclass probabilities, we will need (1) a classifier structure able to compute posterior subclass probabilities, and (2) a learning algorithm able to work in supervised mode at the class level and unsupervised at the subclass level. This will be described in the next section. From now on, let us assume that such estimates of posterior subclass probabilities (based on S_t) are available, and we can compute certain estimates

$$y_{ij} \approx P\{z_{ij} = 1|\mathbf{x}, \mathbf{\Pi}\} \tag{4}$$

---

[1] In spite of the double index in z_{ij}, we do not use a matrix but a vector, $\mathbf{z}$, to group all subclass labels because the number of subclasses may change with the class.

We will use the posterior subclass probabilities estimated from data in S_t to compute the new ones based on the estimated new priors. To do so, it is easy to see that the posterior subclass probabilities based on priors $\mathbf{\Pi}'$ can be estimated from those based on $\mathbf{\Pi}$ through a straightforward transformation:

$$P\{z_{ij}=1|\mathbf{x},\mathbf{\Pi}'\} = \frac{\frac{\pi'_{ij}}{\pi_{ij}} P\{z_{ij}=1|\mathbf{x},\mathbf{\Pi}\}}{\sum_{m=0}^{L-1}\sum_{n=1}^{M_m}\frac{\pi'_{mn}}{\pi_{mn}}P\{z_{mn}=1|\mathbf{x},\mathbf{\Pi}\}} \tag{5}$$

According to our problem statement, neither the posterior probabilities, $P\{z_{ij}=1|\mathbf{x},\mathbf{\Pi}\}$, nor the priors $\mathbf{\Pi}$ and $\mathbf{\Pi}'$ are known, and they must be estimated from data. In [7], a two step approach to re-estimate posterior class probabilities based on new priors, is provided. The basic procedure is: (1) estimating the posterior probabilities by using the labeled samples in S_t, and (2) re-estimating the test posterior probabilities by using the samples in S'_u and by adapting the posterior probabilities obtained in step (1) to the new priors, $\mathbf{\Pi}'$, from an Expectation-Maximization (EM) algorithm. We propose to extend the previous method to re-estimate posterior subclass probabilities, as follows:

At initialization, new priors are set equal to the original ones, estimated as[2]

$$\pi'^{(0)}_{ij} = \pi_{ij} \approx \frac{1}{N_t}\sum_{l=1}^{N_t} P\{z_{ij}=1|\mathbf{x}^l,\mathbf{\Pi}\} \tag{6}$$

In the k-th iteration, the estimation of the a priori and a posteriori probabilities relative to the operating conditions is given by

$$\pi'^{(k)}_{ij} = \frac{1}{N'_t}\sum_{l=1}^{N'_t} P\{z_{ij}=1|\mathbf{x}'^l,\mathbf{\Pi}'^{(k-1)}\} \tag{7}$$

$$P\{z_{ij}=1|\mathbf{x}'^l,\mathbf{\Pi}'^{(k)}\} = \frac{\frac{\pi'^{(k)}_{ij}}{\pi_{ij}} P\{z_{ij}=1|\mathbf{x}'^l,\mathbf{\Pi}\}}{\sum_{m=0}^{L-1}\sum_{n=1}^{M_m}\frac{\pi'^{(k)}_{mn}}{\pi_{mn}}P\{z_{mn}=1|\mathbf{x}'^l,\mathbf{\Pi}\}} \tag{8}$$

Since each class may have a relatively large number of subclasses, we propose an alternative in order to accelerate convergence. After each iteration k, where subclass probabilities are updated and thus, class prior and posterior ones, a certain number of iterations N_s are carried out but now, exclusively within each class, what allows to evolve faster to the distribution of subclasses within a class. This procedure is summarized in Algorithm 1.

2.2 Class and Subclass Posterior Probability Estimation

The re-estimation algorithm proposed in Section 2.1 requires some subclass posterior probability estimates. We have applied a neural structure computing a parametric nonlinear mapping $\mathbf{f}(\mathbf{w},\cdot) : \Re^N \rightarrow \mathcal{P}$ from the input space

---

[2] Despite the fact that the posterior probabilities in Eq.(6) depend on $\mathbf{\Pi}$ (which states a recursive relation), they are not actually required. The posteriors will be estimated using a classifier, so that Eq.(6) can be used directly to compute the initial priors.

into probability space $\mathcal{P} = \{\mathbf{p} | 0 \leq p_{ij} \leq 1, \sum_{i=0}^{L-1} \sum_{j=1}^{M_i} p_{ij} = 1\}$, where $\mathbf{w}$ is a parameter vector.We assume that the classifier hard decisions are given by $\widehat{\mathbf{d}} = \arg\max_i y_i$ where *soft* decisions y_i are given by $y_i = \sum_{j=1}^{M_i} y_{ij}$ and $y_{ij} = f_{ij}(\mathbf{w}, \cdot)$ is the $\{i,j\}-$component of the nonlinear mapping, $\mathbf{f}$. In particular, our experiments are conducted are based on the softmax nonlinearity given by $f_{ij}(\mathbf{w}, \mathbf{x}) = \dfrac{\exp(\mathbf{w}_{ij}^T \mathbf{x} + w_{ij0})}{\sum_{m=0}^{L-1} \sum_{n=1}^{M_m} \exp(\mathbf{w}_{mn}^T \mathbf{x} + w_{mn0})}$. We will refer to this network as *Generalized Softmax Perceptron* (GSP).

Algorithm 1. Subclass-level re-estimation

```
Inputs:
Data from $S'_u = \{\mathbf{x}'^l, l = 1, \ldots, N'_t\}$.
Intermediate and final outputs: $y_{ij}^l$, $y_i^l = \sum_{j=1}^{M_i} y_{ij}^l$
Training priors (based on $S_t$), $\pi_{ij}$.

Initialize: $\pi_{ij}'^{(0)} = \pi_{ij}$ and $Q_{ij}^{(0)}(\mathbf{x}'^l) = P\{z_{ij} = 1 | \mathbf{x}'^l, \mathbf{\Pi}'^{(0)}\} \approx y_{ij}^l$
for $k = 1$ to $K$ do

    $$\pi_{ij}'^{(k)} = \frac{1}{N'_t} \sum_{l=1}^{N'_t} Q_{ij}^{(k-1)}(\mathbf{x}'^l)$$

    $$Q_{ij}^{(k)}(\mathbf{x}'^l) = P\{z_{ij} = 1 | \mathbf{x}'^l, \mathbf{\Pi}'^{(k)}\} \approx \frac{\frac{\pi_{ij}'^{(k)}}{\pi_{ij}} y_{ij}^l}{\sum_{m=0}^{L-1} \sum_{n=1}^{M_m} \frac{\pi_{mn}'^{(k)}}{\pi_{mn}} y_{mn}^l} \qquad (9)$$

    for $r = 0$ to $L-1$ do
        Initialize $\pi_{rj}^{*(0)} = \pi_{rj}'^{(k)}$ and $Q_{rj}^{*(0)}(\mathbf{x}'^l) = Q_{rj}^{(k)}(\mathbf{x}'^l)$, $1 \leq j \leq M_r$
        for $s = 1$ to $N_s$ do
            Compute $\pi_{rj}^{*(s)} = \frac{1}{N'_t} \sum_{l=1}^{N'_t} Q_{rj}^{*(s-1)}(\mathbf{x}'^l)$

            $$Q_{rj}^{*(s)}(\mathbf{x}'^l) = P\{z_{rj} = 1 | \mathbf{x}'^l, \mathbf{\Pi}^{*(s)}\} \approx \frac{\frac{\pi_{rj}^{*(s)}}{\pi_{rj}^{*(0)}} Q_{rj}^{*(0)}(\mathbf{x}'^l)}{\sum_{p=1}^{M_r} \frac{\pi_{rp}^{*(s)}}{\pi_{rp}^{*(0)}} Q_{rp}^{*(0)}(\mathbf{x}'^l)} \sum_{j=1}^{M_r} Q_{rj}^{*(0)}(\mathbf{x}'^l) \qquad (10)$$

        end for
    end for
    Update $Q_{ij}^{(k)}(\mathbf{x}'^l) = Q_{ij}^{*(N_s)}(\mathbf{x}'^l)$
end for
Outputs:
Adjusted posterior subclass probabilities: $P\{z_{ij} = 1 | \mathbf{x}'^l, \mathbf{\Pi}'\} \approx Q_{ij}^{(K)}(\mathbf{x}'^l)$
Adjusted posterior class probabilities: $P\{\mathbf{d} = \mathbf{u}_i | \mathbf{x}'^l, \mathbf{\Pi}'^{(K)}\} \approx \sum_{j=1}^{M_i} Q_{ij}^{(K)}(\mathbf{x}'^l)$
```

Since only class labels are available in S_t the learning algorithm must estimate subclass probabilities in an unsupervised way. Moreover, the number of subclasses is usually unknown, and it must be estimated from data. To do so, a learning algorithm, called PPMS (*Posteriori Probability Model Selection*) is proposed in [3] that estimates the GSP parameters, and determines the number of subclasses per class, in an iterative manner from a class-labeled data set. The parameter estimation rules are derived as an instance of the EM algorithm based on taking subclass labels z_{ij} as hidden variables. In its simplest form, the M step is replaced by a single step of a stochastic gradient minimization, which leads to learning rules

$$\mathbf{w}_{ij}^{(k)} = \mathbf{w}_{ij}^{(k-1)} + \mu^{(k)} \frac{y_{ij}^{(k)}}{y_i^{(k)}} (d_i^{(k)} - y_i^{(k)}) \mathbf{x}^{(k)} \tag{11}$$

where $\mu^{(k)}$ is the learning step at iteration k.[3] The number of subclasses is estimated through a combination of pruning, splitting and merging processes[4] (see [3] for more details). In our experiments, learning step $\mu^{(k)}$ decreases according to $\mu^{(k)} = \frac{\mu^{(0)}}{1+k/\eta}$, where η is a decay factor.

The fact that intermediate outputs y_{ij} of the GSP can be interpreted as subclass probabilities provides quite a natural way to cope with the unexplored problem of uncertainty in subclass distributions as already pointed out by [9]. Nonetheless, both architecture and cost function issues are not the goal of this paper, but merely illustrative tools.

3 Experimental Results

We first illustrate the proposed re-estimation strategy on two applications from the UCI and then, we apply it to a real remote sensing application.

Applications from the UCI. The Waveform problem considered derives from the original application and it has been reformulated as a binary problem where class-0 contains just data from one wave and class-1 gathers data from the other two waves. The goal of the Abalone dataset is to discriminate whether the age of an abalone is lower than 10 (class-0) or not (class-1). For simulation purposes, two clusters are identified in class-1: instances with age between 10 and 20 and those older than 20. Training class distribution as well as different test scenarios (simulated by means of over/down sampling) are shown in Table 1[5].

Classification is carried out by a GSP neural network with structure automatically configured by the PPMS algorithm (threshold values $\mu_{\text{prune}} = 0.01$,

---

[3] This rule can also be derived through stochastic minimization of a cross entropy cost function $CE(\mathbf{y}, \mathbf{d}) = -\sum_{i=0}^{L-1} d_i \log y_i$.

[4] Thresholds μ_{prune}, μ_{split} and μ_{merge} are applied to priors in order to activate these actions.

[5] Both classification problems will be assessed in production mode conditions with the same or different class/subclass distributions to those assumed in learning.

Table 1. Class and subclass distributions for the Waveform and Abalone applications

		Class distributions		
		Class-0	Class-1	
	Training	Π_{00}	Π_{10}	Π_{11}
	S0	0.34	0.33	0.33
Distribution changes	*Test*	Π'_{00}	Π'_{10}	Π'_{11}
(a) No changes	S1	0.34	0.33	0.33
(b) At class-level	S2	0.20	0.40	0.40
	S3	0.60	0.20	0.20
(c) At subclass-level	S4	0.34	0.05	0.61
	S5	0.34	0.19	0.47
	S6	0.34	0.61	0.05
(d) At class- and subclass-level	S7	0.20	0.05	0.75
	S8	0.20	0.225	0.575
	S9	0.20	0.75	0.05
	S10	0.60	0.05	0.35
	S11	0.60	0.275	0.125
	S12	0.60	0.35	0.05

$\mu_{\text{merge}} = 0.002$ and $\mu_{\text{split}} = 0.5$). Learning rate is set to $\mu^{(0)} = 0.02$ and $\mu^{(0)} = 2.1$ for the Waveform and Abalone, respectively and $\eta = 20000$.

It can be seen in Table 2 (results averaged over 5 runs) that whenever subclass distributions keep the same as in training ((a) and (b) scenarios), both class and within-class re-estimation strategies show no differences in performance. Note also that with no operating changes, (a), the error rate does not increase because of re-estimation process. On the other hand, in (b), both re-estimation approaches achieve a significant decrease in error rate: 9% in S2 and 16% in S3 for the Waveform dataset and 11% in S2 and 22% in S3 for the Abalone problem.

However, when there are shifts in subclass distributions (with the same or different class priors) the class-level re-estimation process is further improved by the subclass re-estimation approach (cases (c) and (d)). For instance, in Abalone-S12, error rate is reduced from 17.6% to 15.2%, while for the standard classifier is 24.8%. Likewise, in Waveform-S12, the subclass re-estimation strategy leads to 8.1% error rate, whereas the class-level one is 8.9%. Note also that the class-level

Table 2. Test error rate (in %) for the Abalone and Waveform datasets

		Methods					
		Standard		Class-level re-estimation		Subclass-level re-estimation	
Scenarios	Test set	Abalone	Waveform	Abalone	Waveform	Abalone	Waveform
(a)	S1	21.4	8.4	21.5	8.5	21.4	8.4
(b)	S2	18.2	7.4	16.1	6.7	16.5	6.8
	S3	27.8	9.2	21.6	7.7	21.7	7.8
(c)	S4	27.0	7.2	37.4*	7.4	27.5	6.1
	S5	24.2	7.8	26.7	7.9	24.3	7.8
	S6	15.9	9.5	20.1*	9.6	14.8	8.1
(d)	S7	25.6	6.1	35.8*	6.4	19.5	5.0
	S8	21.9	6.8	21.2	6.5	18.1	6.4
	S9	10.8	8.7	15.8*	7.1	10.6	6.5
	S10	30.6	8.6	31.2	6.7	28.6	5.5
	S11	26.2	9.4	18.9	8.2	18.6	8.0
	S12	24.8	9.8	17.6	8.9	15.2	8.1

Table 3. Overall error rate in % averaged over 20 runs for a Landsat image dataset

	Error rate in different test partitions		
Training set	Standard	Class-level Re-estimation	Subclass-level Re-estimation
45% southern	2.86	2.77	2.55
50% southern	2.72	2.24	2.20
55% southern	2.72	2.12	2.06
60% southern	1.21	1.44	1.19
65% southern	0.90	1.03	0.94

re-estimation approach fails in several scenarios, leading to error rates that are sometimes close to 50% higher than the standard classifier: in Abalone-S7, the standard classifier error is 25.6%, and with class-level re-estimation, it increases to 35.8%. However, re-estimating by considering the subclasses reduces the error to 19.5%[6]. Finally, there are some cases where the subclass re-estimation does not reduce error rate but it does not either deteriorate performance. Thus, for Abalone-S9, the standard classifier error is 10.8%, what becomes higher (15.8%) when re-estimating at class level, while subclass re-estimation yields 10.6%.

Landsat Image. This application dataset (10024 examples, 7 features) belongs to a Landsat image that covers an area in the North-West of Spain (the same used in [2]). For us, the goal is to discriminate between *urban* and *non urban* terrains. Class *urban* contains data belonging to different subclasses (yet unknown tag): dense-urban, scattered-urban and industrial. Likewise, class *non urban* gathers data belonging to several different subclasses. Learning rate was initialized to $\mu^{(0)} = 0.03$, decay factor to $\eta = 100000$ and the PPMS algorithm thresholds to $\mu_{\text{prune}} = 0.00005$, $\mu_{\text{merge}} = 0.0001$ and $\mu_{\text{split}} = 0.5$.

Several partitions have been carried out to split the whole dataset into training and test according to the spatial distribution north-south. The goal is to asses the performance of the classifier trained with data belonging to a certain region and afterwards, test it on the remaining region that may not show the same class and/or subclass distribution. Table 3 shows error rate in the test set when the classifier is trained with different percentage splits of southern data (from 45% to 65%). Twenty runs were carried out and Table 3 reports the overall error rate values. We can see that, in two out of the five scenarios, the class-level adaptive approach fails, leading to an increase in error rate (from 1.21 to 1.44 and from 0.90 to 1.03) whereas the within-class re-estimation approach neither significantly improves nor deteriorates performance in these two cases. Although both approaches succeed in the other three scenarios, the subclass re-estimation strategy achieves higher decrease in the error rate (for instance, 2.55 compared with 2.77, for the 45% southern split).

Finally, an experiment was run with the two UCI datasets, in which the split between training data and the test set is carried out by establishing a threshold in one of the feature variables (which afterwards is removed from the dataset).

---

[6] Note that in Table 2, those test sets where the conventional adaptive process leads to an unacceptable increase in error rate are highlighted with "$*$".

We found that the re-estimation process might even deteriorate performance in these scenarios. Data analysis shows significant correlations between features, what we believe leads to a drift in the dependence relationship between the class label (and the estimated subclass) and the feature variables. To guarantee the success of a re-estimation process a method should be derived, to determine if the problem we face implies a deeper underlying change in data distribution.

4 Conclusions and Future Work

We have proposed a method to adapt a classifier to changes in prior subclass probabilities using unlabeled test data. We should make clear that the subclasses are "virtual" entities, because both training and reestimation processes are unsupervised at the subclass level (subclass labels are never used).

Experimental results show that, taking into account the shifting in subclass distributions allows to get an adaption to the new context better than the class-level approach that restricts itself to shifts in class distributions. Moreover, it is been observed that the adaptive method at class level fails (if class conditional densities change), sometimes significantly worsening the performance of a classifier that has not been adjusted. Additionally, some preliminary results show situations of data distribution changes that require further research.

References

1. Alaiz-Rodríguez, R., Guerrero-Curieses, A., Cid-Sueiro, J.: Minimax regret classifier for imprecise class distributions. Journal of Machine Learning Research 8, 103–130 (2007)
2. Alba, J., Docio, L., Docampo, D., Marquez, O.: Growing gaussian mixtures network for classification applications. Signal Process 76(1), 43–60 (1999)
3. Arribas, J.I., Cid-Sueiro, J.: A model selection algorithm for a posteriori probability estimation with neural networks. IEEE Transactions on Neural Networks 16(4), 799–809 (2005)
4. Flawcett, T., Flach, P.: A response to webb and ting's on the application of roc analysis to predict classification performance under varying class distributions. Machine Learning 58, 33–38 (2005)
5. Holte, R.: Elaboration on two points raised in classifier technology and the illusion of progress. Statistical Science 21(1) (2006)
6. Provost, F., Fawcett, T.: Robust classification systems for imprecise environments. Machine Learning 42(3), 203–231 (2001)
7. Saerens, M., Latinne, P., Decaestecker, C.: Adjusting a classifier for new a priori probabilities: A simple procedure. Neural Computation 14, 21–41 (2002)
8. Shimodaira, H.: Improving predictive inference under convariance shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference (2000)
9. Webb, G., Ting, K.: On the application of roc analysis to predict classification performance under varying class distributions. Machine Learning 58(1), 25–32 (2005)
10. Yamazaki, K., Kawanabe, M., Watanabe, S., Sugiyama, M., Müller, K.: Asymptotic bayesian generalization error when training and test distributions are different. In: Proc. of the 24th international conference on Machine learning (2007)

Improving Training in the Vicinity of Temporary Minima*

Ido Roth and Michael Margaliot**

School of Elec. Eng.-Systems, Tel Aviv University, Israel 69978
`michaelm@eng.tau.ac.il`
`www.eng.tau.ac.il/∼michaelm`

Abstract. An important problem in learning using gradient descent algorithms (such as backprop) is the slowdown incurred by temporary minima (TM). We consider this problem for an artificial neural network trained to solve the XOR problem. The network is transformed into the equivalent *all permutations fuzzy rule-base* which provides a symbolic representation of the knowledge embedded in the network. We develop a mathematical model for the evolution of the fuzzy rule-base parameters during learning in the vicinity of TM. We show that the rule-base becomes singular and tends to remain singular in the vicinity of TM.

Our analysis suggests a simple remedy for overcoming the slowdown in the learning process incurred by TM. This is based on slightly perturbing the values of the training examples, so that they are no longer symmetric. Simulations demonstrate the usefulness of this approach.

1 Introduction

A feedforward artificial neural network (ANN) defines a mapping from its input $\boldsymbol{z} \in \mathbb{R}^m$ to its output $\boldsymbol{y} \in \mathbb{R}^n$. It is possible to express this in closed form as $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{z})$. Yet this usually provides little insight on what the ANN is actually computing. This black-box character of ANNs is a major shortcoming. Methods for extracting the knowledge embedded in ANNs are surveyed in [1,2,3,4].

A very different form of knowledge representation is provided by rule-based systems and, in particular, fuzzy rule-based systems. Here, the knowledge is represented as a set of If-Then rules, stated using *natural language*. The system is thus easy to understand, verify, and, if necessary, refine. Considerable research effort has been devoted to developing a synergy between the learning-from-examples ability of ANNs and the white-box character of fuzzy systems [5,6].

The *all-permutations fuzzy rule-base* (FARB) introduced in [7] admits an input–output mapping (IOM) that is identical to that of an ANN. More specifically, there exists an invertible mapping T converting an ANN into a FARB with an identical IOM:

$$T(\text{ANN}) = \text{FARB}, \quad T^{-1}(\text{FARB}) = \text{ANN}. \tag{1}$$

---

[*] Research supported in part by research grants from the Israel Science Foundation (ISF) and the Israeli Ministry of Science and Technology.

[**] Corresponding author.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 131–139, 2009.
© Springer-Verlag Berlin Heidelberg 2009

This provides a bi-directional transformation between ANNs and symbolic FARBs.

Numerous applications of the ANN-FARB equivalence for extracting and inserting symbolic information into ANNs can be found in [8]. Advantages of this approach include: (a) the FARB is based on standard fuzzy logic tools (rather than some special operators tailored to fit the ANN functioning); and (b) the transformation (1) is applicable to a large set of ANN architectures and to any set of ANN parameters.

Consider an ANN during training. Since the transformation T is valid for any set of parameters, it can be applied both before and after every learning iteration. This makes it possible to compare a *symbolic* representation of the knowledge embedded in the ANN after each learning iteration, and, more generally, to study the time evolution of the symbolic FARB parameters, rather than the ANN weights, during on-line learning.

In this paper, we apply this idea to a simple ANN trained to solve the XOR problem. We do this to gain new insight into the learning stagnation in the vicinity of temporary minima (TM) (sometimes referred to as the *plateau phenomena*). Our analysis shows that one reason for the learning slowdown is the symmetry of the training examples. This suggests a new and simple approach for improving the training behavior by slightly perturbing the desired output values in the training examples so that they are no longer symmetric. Simulations demonstrate the effectiveness of this approach.

2 Learning in the Vicinity of TM: The XOR Problem

The dynamics of ANNs during training is extremely complicated. It is thus customary to study this dynamics using simple examples. Following [9,10,11], we consider a 2-2-1 feedforward ANN trained to realize the XOR function (see Fig. 1). All neurons use the sigmoid activation function:

$$\sigma(y) := (1 + \exp(-y))^{-1} \ . \tag{2}$$

Note that differentiating (2) yields: $\sigma'(y) = \sigma(y)(1 - \sigma(y))$.

The IOM of the XOR ANN $y : \mathbb{R}^2 \to \mathbb{R}$ is given by

$$y(\boldsymbol{z}) = \sigma(w_1^h f_1(\boldsymbol{z}) + w_2^h f_2(\boldsymbol{z}) + w_0^h) \ , \tag{3}$$

where $f_i(\boldsymbol{z}) := \sigma(w_1^i z_1 + w_2^i z_2 + w_0^i)$ is the output of the ith hidden neuron.

The training examples are pairs in the form $\boldsymbol{t}^i = (\boldsymbol{z}^i; d_i)$, with $\boldsymbol{t}^1 = (1\ 0\ 0;\ 0)$, $\boldsymbol{t}^2 = (1\ 1\ 0;\ 1)$, $\boldsymbol{t}^3 = (1\ 0\ 1;\ 1)$, and $\boldsymbol{t}^4 = (1\ 1\ 1;\ 0)$, (the first component in $\boldsymbol{z}$ is the bias input). The required IOM is that of the XOR function. Training is performed using the batch version of the backprop (BP) algorithm, i.e. the cost function is

$$E := \frac{1}{2} \sum_{i=1}^{4} (d_i - y_i)^2, \tag{4}$$

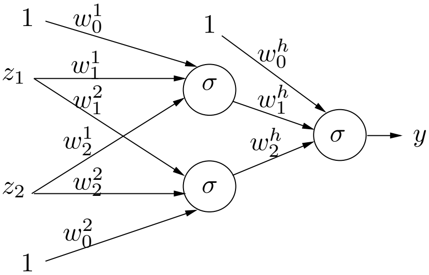


Fig. 1. A 2-2-1 feedforward ANN trained to solve the XOR problem

where $y_i := y(\boldsymbol{z}^i)$. In the kth learning epoch, weight w is updated according to $w(k+1) = w(k) - \eta\frac{\partial E}{\partial w}$, where $\eta > 0$ is the learning rate. It is useful to approximate this as a continuous-time process, i.e. $\dot{w}(t) \approx \frac{w(k+1)-w(k)}{\eta} = -\frac{\partial E}{\partial w}$.

Since BP is a gradient descent algorithm, it may converge to a local minima of E. Furthermore, the algorithm may require many epochs trying to “escape” points where the gradient is almost zero. Such points, that are not local minima, are known as *temporary minima*. These correspond to almost flat regions, or plateaus, in the landscape of the cost function.

The XOR ANN was used to analyze the dynamics of learning in the vicinity of TM [9,10,11]. In this problem no local minima exist, but there do exist four classes of TM, of which only one is characterized by finite weights [9,10,11]. In the vicinity of TM, the ANN demonstrates extremely slow learning. For many epochs of training the change of the weight values and of the error E are close to zero. Furthermore, when the network is initialized with small weights (which is customary practice in order to avoid saturation [12]), the dynamics tends to converge toward a TM. Thus, the very slow learning is by no means a rare event [13]. Slow learning near TM is a major problem in ANN training, as training is usually abandoned altogether if the BP algorithm does not converge after some fixed number of epochs.

Our simulations of the XOR network support this. We implemented BP using MATLAB with learning rate $\eta = 3.5$ for all weight updates. The initial weights were drawn randomly from the uniform distribution $U[-0.2,\ 0.2]$. A typical response is depicted in Fig. 2. It may be seen that the ANN output hardly changes for over 2000 epochs. The behavior for the other three possible inputs is basically similar. After finally breaking away from the TM, learning converges to a correct set of weights rather quickly. This implies that the minima is indeed a temporary minima, and not a local one.

2.1 Dynamics of Learning

In this section, we review the analysis of learning in the vicinity of TM described in [9]. Let $\boldsymbol{z} = (1\ z_1\ z_2)^T$. Denote

$$\boldsymbol{w}^i := (w_0^i\ w_1^i\ w_2^i)^T\ , \quad i = 1, 2,$$

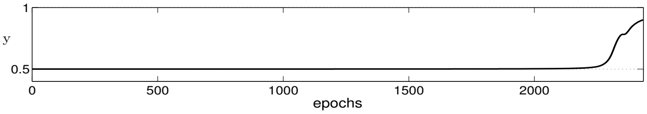


Fig. 2. ANN output y for the input $\boldsymbol{z}^2 = (1, 1, 0)^T$ as a function of learning epochs. The desired output is $d_2 = 1$. There is almost no change in output for over 2000 epochs.

and

$$\begin{aligned} \boldsymbol{\epsilon} &:= (\boldsymbol{w}^1 - \boldsymbol{w}^2)/2, & \boldsymbol{\omega} &:= (\boldsymbol{w}^1 + \boldsymbol{w}^2)/2, \\ u &:= (w_1^h - w_2^h)/2, & v &:= (w_1^h + w_2^h)/2 \ . \end{aligned} \tag{5}$$

Note that when $\boldsymbol{\epsilon} = \mathbf{0}$ and $u = 0$ the ANN output y in (3) becomes $y(\boldsymbol{z}) = \sigma(2w_1^h f_1(\boldsymbol{z}) + w_0^h)$, so the ANN is effectively reduced to a single neuron.

Ampazis et al. [9] derived approximate equations describing the dynamics in the vicinity of $(u = 0, \boldsymbol{\epsilon} = \mathbf{0})$, under the additional assumption of small weight values. We use $o(x)$ to denote functions $h(x)$ that satisfy $\lim_{x \to 0} \frac{h(x)}{x} = 0$.

Lemma 1. *[9] Define*

$$\xi_i := (d_i - y_i) y_i (1 - y_i) \ , \quad i = 1, \ldots, 4 \ . \tag{6}$$

Also define $\boldsymbol{p} \in \mathbb{R}^3$*,* $Q \in \mathbb{R}^{3 \times 3}$*, and* $J \in \mathbb{R}^{4 \times 4}$ *by:*

$$\boldsymbol{p} := \sum_{i=1}^{4} \xi_i \sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)(1 - \sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)) \boldsymbol{z}^i \ , \tag{7}$$

$$Q := \sum_{i=1}^{4} \xi_i \sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)(1 - \sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)) v (1 - 2\sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)) \boldsymbol{z}^i (\boldsymbol{z}^i)^T \ , \tag{8}$$

and $J := \begin{bmatrix} 0 & \boldsymbol{p}^T \\ \boldsymbol{p} & Q \end{bmatrix}$*. Near* $(u = 0, \boldsymbol{\epsilon} = \mathbf{0})$*, the learning dynamics is*

$$\begin{bmatrix} \dot{u} \\ \dot{\boldsymbol{\epsilon}} \end{bmatrix} = J \begin{bmatrix} u \\ \boldsymbol{\epsilon} \end{bmatrix} + \begin{bmatrix} o((\boldsymbol{\epsilon}^T \boldsymbol{z}^p)^2) \\ \sum_{i=1}^{4} \xi_i \boldsymbol{z}^i (v \cdot o((\boldsymbol{\epsilon}^T \boldsymbol{z}^i)^2) + u \cdot o(\boldsymbol{\epsilon}^T \boldsymbol{z}^i)) \end{bmatrix} \ , \tag{9}$$

$$\dot{v} = \sum_{i=1}^{4} \xi_i (\sigma(\boldsymbol{w}^T \boldsymbol{z}^i) + o(\boldsymbol{\epsilon}^T \boldsymbol{z}^i)) \ . \tag{10}$$

Using a similar approach as in [9] also yields the dynamics for $\boldsymbol{\omega}$.

Lemma 2. *Define* $Q' \in \mathbb{R}^{3 \times 3}$ *by:*

$$Q' := \sum_{i=1}^{4} \xi_i \sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)(1 - \sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)) u (1 - 2\sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)) \boldsymbol{z}^i (\boldsymbol{z}^i)^T \ .$$

Near $(u = 0, \boldsymbol{\epsilon} = \mathbf{0})$,

$$\dot{\boldsymbol{\omega}} = v\boldsymbol{p} + Q'\boldsymbol{\epsilon} + \sum_{i=1}^{4} \xi_i \boldsymbol{z}^i (u \cdot o((\boldsymbol{\epsilon}^T \boldsymbol{z}^i)^2) + v \cdot o(\boldsymbol{\epsilon}^T \boldsymbol{z}^i)) \ . \tag{11}$$

Note that the desired output values d_i appear in the definition of ξ_i and thus have a direct effect on the dynamical equations.

3 Equivalent FARB

In this section, we present a FARB that is equivalent to the 2-2-1 ANN, and derive the mathematical equations describing the FARB dynamics during learning in the vicinity of TM.

Consider a fuzzy rule-base with inputs x_1, x_2 and output g described by the following four rules:

$$\begin{aligned}
&R_1 : \text{If } (x_1 \text{ is } \textit{positive}) \text{ and } (x_2 \text{ is } \textit{positive}),\ \text{then } \sigma^{-1}(g) = a_0 + a_1 + a_2,\\
&R_2 : \text{If } (x_1 \text{ is } \textit{positive}) \text{ and } (x_2 \text{ is } \textit{negative}),\ \text{then } \sigma^{-1}(g) = a_0 + a_1 - a_2,\\
&R_3 : \text{If } (x_1 \text{ is } \textit{negative}) \text{ and } (x_2 \text{ is } \textit{positive}),\ \text{then } \sigma^{-1}(g) = a_0 - a_1 + a_2,\\
&R_4 : \text{If } (x_1 \text{ is } \textit{negative}) \text{ and } (x_2 \text{ is } \textit{negative}),\ \text{then } \sigma^{-1}(g) = a_0 - a_1 - a_2,
\end{aligned}$$

where $a_i \in \mathbb{R}$, the *and* operator is modeled using multiplication, and the verbal terms *positive* and *negative* are defined using the membership functions (MFs): $\mu_{pos}(x) := \sigma(x)$, and $\mu_{neg}(x) := \sigma(-x)$. Note that $\mu_{pos}(x) \in (0, 1)$ for all x, and that $\lim_{x \to -\infty} \mu_{pos}(x) = 0$, $\lim_{x \to \infty} \mu_{pos}(x) = 1$.

Let $D_i = D_i(x_1, x_2)$ denote the degree of firing (DOF) of rule R_i, i.e.

$$\begin{aligned}
D_1 &= \mu_{pos}(x_1)\mu_{pos}(x_2),\ D_2 = \mu_{pos}(x_1)\mu_{neg}(x_2),\\
D_3 &= \mu_{neg}(x_1)\mu_{pos}(x_2),\ D_4 = \mu_{neg}(x_1)\mu_{neg}(x_2).
\end{aligned} \tag{12}$$

Let s_i denote the output value in the Then-part of rule R_i,

$$\begin{aligned}
s_1 &= a_0 + a_1 + a_2, & s_2 &= a_0 + a_1 - a_2,\\
s_3 &= a_0 - a_1 + a_2, & s_4 &= a_0 - a_1 - a_2.
\end{aligned} \tag{13}$$

Proposition 1. *Inferring the four-rule FARB using the center of gravity (COG) defuzzification method [14] yields:*

$$\sigma^{-1}(g) = a_0 - a_1 - a_2 + 2a_1\sigma(x_1) + 2a_2\sigma(x_2). \tag{14}$$

This result is a special case of the main result in [7]. Eq. (14) suggests that the IOM of the FARB is very similar to that of an ANN. Indeed, comparing (14) with (3) immediately yields the following result.

Lemma 3. *Consider the ANN with output $y(z_1, z_2)$ given by (3), and the four-rule FARB with inputs*

$$x_i = (\boldsymbol{w}^i)^T \boldsymbol{z} \ , \quad i = 1, 2, \tag{15}$$

and parameter values:

$$a_0 = w_0^h + w_1^h/2 + w_2^h/2, \quad a_1 = w_1^h/2 \ , \ a_2 = w_2^h/2 \ . \tag{16}$$

Then inferencing the FARB yields: $g(x_1, x_2) = y(z_1, z_2)$*.*

In other words, the FARB is mathematically equivalent to the ANN, and thus provides a *symbolic* representation of what the ANN is actually computing.

4 Learning Dynamics of the FARB

We begin by considering the FARB when the ANN parameters satisfy $(u = 0, \boldsymbol{\epsilon} = \mathbf{0})$. In this case, Eqs. (15) and (16) yield: $x_1 = x_2$, and $a_1 = a_2$. This implies that the FARB output is

$$\sigma^{-1}(g) = a_0 - 2a_1 + 4a_1\sigma(x_1). \tag{17}$$

This is actually equivalent to the output of the two-rule FARB:

$$\begin{aligned} &R_1 : \text{If } (x_1 \text{ is } \textit{positive}), \ \text{ then } \sigma^{-1}(g) = a_0 + 2a_1, \\ &R_2 : \text{If } (x_1 \text{ is } \textit{negative}), \ \text{then } \sigma^{-1}(g) = a_0 - 2a_1. \end{aligned} \tag{18}$$

In other words, near the TM, the ANN is reduced to a single neuron, and the corresponding FARB is reduced to a singular, *two-rule* FARB. Our main result in this section describes the dynamics of the FARB in the vicinity of TM. Due to space limitations, proofs are omitted. These will appear in an extended version of this paper [15].

Lemma 4. *Denote* $q_i := \sigma(\boldsymbol{\omega}^T \boldsymbol{z}^i)$, $i = 1, \ldots, 4$. *Near* $(u = 0, \boldsymbol{\epsilon} = \mathbf{0})$*, the learning dynamics of the FARB parameters is:*

$$\begin{aligned} \dot{D}_1 &= \left(\dot{\boldsymbol{\omega}}^T \boldsymbol{z} - x_1(\dot{\boldsymbol{w}}^1)^T \boldsymbol{z}/4 - x_2(\dot{\boldsymbol{w}}^2)^T \boldsymbol{z}/4 + o(x_i^2)\right) D_1, \\ \dot{D}_2 &= \left(\dot{\boldsymbol{\epsilon}}^T \boldsymbol{z} - x_1(\dot{\boldsymbol{w}}^1)^T \boldsymbol{z}/4 - x_2(\dot{\boldsymbol{w}}^2)^T \boldsymbol{z}/4 + o(x_i^2)\right) D_2, \\ \dot{D}_3 &= -\left(\dot{\boldsymbol{\epsilon}}^T \boldsymbol{z} + x_1(\dot{\boldsymbol{w}}^1)^T \boldsymbol{z}/4 + x_2(\dot{\boldsymbol{w}}^2)^T \boldsymbol{z}/4 + o(x_i^2)\right) D_3, \\ \dot{D}_4 &= -\left(\dot{\boldsymbol{\omega}}^T \boldsymbol{z} + x_1(\dot{\boldsymbol{w}}^1)^T \boldsymbol{z}/4 + x_2(\dot{\boldsymbol{w}}^2)^T \boldsymbol{z}/4 + o(x_i^2)\right) D_4, \end{aligned} \tag{19}$$

where $o(x_i^2)$ *stands for* $o(x_1^2) + o(x_2^2)$*, and*

$$\begin{aligned} &\dot{s}_1 = \sum_{i=1}^{4} \xi_i (1 + 2q_i + o(\boldsymbol{\epsilon}^T \boldsymbol{z})) \ , \quad \dot{s}_2 = \sum_{i=1}^{4} \xi_i (1 + q_i + \boldsymbol{\epsilon}^T \boldsymbol{z}^i q_i (1 - q_i) + o(\boldsymbol{\epsilon}^T \boldsymbol{z})) \ , \\ &\dot{s}_3 = \sum_{i=1}^{4} \xi_i (1 + q_i - \boldsymbol{\epsilon}^T \boldsymbol{z}^i q_i (1 - q_i) + o(\boldsymbol{\epsilon}^T \boldsymbol{z})) \ , \quad \dot{s}_4 = \sum_{i=1}^{4} \xi_i \ . \end{aligned} \tag{20}$$

Note that combining (19), Lemma 1 and the fact that the matrix J has eigenvalues close to zero [9], implies that the change in $D_i(t)$, $i = 1, \ldots, 4$, will be very small. Thus, the FARB will tend to remain a singular, two-rule FARB.

5 Simplification of FARB Dynamics

Ampazis et al. [9] showed that for $(u \approx 0, \boldsymbol{\epsilon} \approx \mathbf{0})$ and initial weights that are close to zero, the ANN output is approximately identical for each of the four possible inputs, i.e. there exists $\bar{y} \in \mathbb{R}$ such that: $y_1 \approx \cdots \approx y_4 \approx \bar{y}$. Indeed, letting $net_i := w_1^h f_1(\boldsymbol{z}^i) + w_2^h f_2(\boldsymbol{z}^i) + w_0^h$ (the net input to the output neuron) and using (2) and (3) yields: $y_i = f(\boldsymbol{z}^i) = \frac{1}{2} + \frac{net_i}{4} + o(net_i^2)$, so $\bar{y} = 1/2$.

To simplify the notation, we will use $p \approx q$ to denote that $p - q$ includes terms that are linear (or higher-order) in $\boldsymbol{\omega}^T \boldsymbol{z}^i$ or net_i. For example, $y_i \approx \bar{y}$.

Similarly, the small weight values imply that $q_1 \approx \cdots \approx q_4 = \bar{q}$, with $\bar{q} = 1/2$.

Lemma 5. *Near* $(u = 0, \boldsymbol{\epsilon} = \mathbf{0})$*:*

$$\dot{D}_i(\boldsymbol{z}) \approx 0, \;\; \dot{s}_i \approx 0, \quad i = 1, \ldots, 4. \tag{21}$$

Analysis of the proof of this lemma shows that many terms in the derivatives of the parameters vanish because for the XOR problem:

$$d_2 + d_4 - 2\bar{y} = d_3 + d_4 - 2\bar{y} = 0 \ . \tag{22}$$

In other words, the symmetry of the vector $\boldsymbol{d} = (d_1, d_2, d_2, d_4)^T = (0, 1, 1, 0)^T$ contributes to the slowdown of learning in the vicinity of TM. This suggests a simple approach for alleviating this problem, namely, slightly perturb the desired output values so that (22) no longer holds, say, $\boldsymbol{d} = (0.1, 1, 0.95, 0)^T$. Note that if the mean square error (MSE) during learning converges to zero, then the final ANN can obviously be said to solve the XOR problem even with this perturbation.

The fact that (22) does not hold implies that the zero terms in the expression for the derivatives become non-zero. These terms have small absolute value, but will have a strong effect on the learning behavior. This is because near TM, the small values of the derivatives slowly accumulate until a bifurcation takes place, leading the system away from the TM region [9]. The small non-zero terms accumulate much faster and thus tend to considerably reduce the number of epochs spent near TM. The next section demonstrates the effectiveness of this simple idea.

6 Using a Non-symmetric $\boldsymbol{d}$

We ran 1000 simulations of BP training for the XOR problem using: (1) $\boldsymbol{d} = (0, 1, 1, 0)^T$; and (2) $\boldsymbol{d} = (0.1, 1, 0.95, 0)^T$. All the other parameters (e.g., initial weight values, learning rate) were identical. Training was stopped when either the number of training epochs reached 18,000, or the ANN output y satisfied $|d_i - y_i| \leq 0.15$, for $i = 1, \ldots, 4$.

Table 1 summarizes the results. It may be seen that when the training converges, using a non-symmetric $\boldsymbol{d}$ speeds up learning by a factor of about 4.5.

Table 1. Training performance in the symmetric and non-symmetric cases

	symmetric *d*	**non-symmetric *d***
Mean number of epochs until convergence	1850	415
Standard deviation	1700	260
Failure rate	1.5%	3.7%
Failure reason	Weight saturation & epoch number limit	Weight saturation

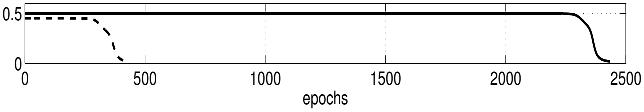


Fig. 3. MSE during learning: symmetric (solid line) and non-symmetric case (dashed)

This is mainly due to reducing the time spent near TM (see Fig. 3 for a typical example). The non-symmetric case does however increase failure rate. Failure is always due to saturation and was overcome using the modified BP algorithm defined in [16], bringing failure rate down to 0.6%. In the symmetric case, failure is due to both saturation and the slowdown of learning near TM, so the ANN is still "stuck" on the plateau after 18,000 learning epochs.

7 Discussion

Amari et al. [13] regard the plateau region as a pseudo-attractor for the learning dynamics. This highlights the importance of understanding and overcoming the problem of slow learning in the vicinity of TM.

In this paper, we analyzed this problem for the XOR ANN using the equivalent FARB. Our analysis led to a new approach for overcoming slow learning which is based on slightly perturbing the learning examples so that they are no longer symmetric. This indeed reduces the time needed to traverse the TM region, but may increase the probability of failure to converge due to saturation.

To overcome this we suggest combining a non symmetric $\boldsymbol{d}$ with learning using an improved BP algorithm that is designed to overcome weight saturation. Simulations demonstrate the usefulness of this approach.

Perturbing the training examples to improve learning is not an entirely new idea. Bishop [17] demonstrated the relationship between training with noise and Tikhonov regularization. Wang and Principe [18] suggested adding a small random noise at each learning step. Our approach is different as it is derived from an analysis of the evolution of the equivalent FARB when the ANN is in the vicinity of TM.

More research is needed in order to study the implications of using non-symmetric training examples in other learning problems. Preliminary experiments in the parity problem for 4, 5, 6, and 7 bits suggest that the approach presented here is indeed successful. Furthermore, as the complexity of the learning problem increases, overcoming the slow learning problem becomes more important.

References

1. Andrews, R., Diederich, J., Tickle, A.B.: Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based Systems 8, 373–389 (1995)
2. Cloete, I., Zurada, J.M. (eds.): Knowledge-Based Neurocomputing. MIT Press, Cambridge (2000)
3. Jacobsson, H.: Rule extraction from recurrent neural networks: A taxonomy and review. Neural Computation 17, 1223–1263 (2005)
4. Tickle, A.B., Andrews, R., Golea, M., Diederich, J.: The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks. IEEE Trans. Neural Networks 9, 1057–1068 (1998)
5. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Englewood Cliffs (1997)
6. Li, C., Cheng, K.H.: Recurrent neuro-fuzzy hybrid-learning approach to accurate system modeling. Fuzzy Sets and Systems 158, 194–212 (2007)
7. Kolman, E., Margaliot, M.: Are artificial neural networks white boxes? IEEE Trans. Neural Networks 16, 844–852 (2005)
8. Kolman, E., Margaliot, M.: Knowledge-Based Neurocomputing: A Fuzzy Logic Approach. Studies in Fuzziness and Soft Computing, vol. 234. Springer, Heidelberg (2009)
9. Ampazis, N., Perantonis, S.J., Taylor, J.G.: Dynamics of multilayer networks in the vicinity of temporary minima. Neural Networks 12, 43–58 (1999)
10. Annema, A.J., Hoen, K., Wallinga, H.: Learning behavior and temporary minima of two-layer neural networks. Neural Networks 7, 1387–1404 (1994)
11. Hamey, L.G.C.: XOR has no local minima: A case study in neural network error surface analysis. Neural Networks 11, 669–681 (1998)
12. Lee, O., Oh, S.H., Kim, M.W.: An analysis of premature saturation in back propagation learning. Neural Networks 6, 719–728 (1993)
13. Amari, S., Park, H., Ozeki, T.: Singularities affect dynamics of learning in neuromanifolds. Neural Computation 18, 1007–1065 (2006)
14. Sousa, J.W.C., Kaymak, U.: Fuzzy Decision Making in Modeling and Control. World Scientific, Singapore (2002)
15. Roth, I., Margaliot, M.: Analysis of learning near temporary minima using the all permutations fuzzy rule-base (submitted for publication) (2009), `http://www.eng.tau.ac.il/~michaelm`
16. Wang, X., Tang, Z., Tamura, H., Ishii, M., Sun, W.: An improved backpropagation algorithm to avoid the local minima problem. Neurocomputing 56, 455–460 (2004)
17. Bishop, C.M.: Training with noise is equivalent to Tikhonov regularization. Neural Computation 7(1), 108–116 (1995)
18. Wang, C., Principe, J.: Training neural networks with additive noise in the desired signal. IEEE Trans. Neural Networks 10, 1511–1517 (1999)

Convergence in an Adaptive Neural Network: The Influence of Noise Inputs Correlation

Adel Daouzli[1], Sylvain Saïghi[1], Michelle Rudolph[2], Alain Destexhe[2],
and Sylvie Renaud[1]

[1] IMS Labs, University of Bordeaux, 351 cours de la Libération,
33400 Talence, France
`adel.daouzli@ims-bordeaux.fr`
[2] UNIC - CNRS, 1 av. de la Terrasse,
F91198 Gif-sur-Yvette, France

Abstract. This paper presents a study of convergence modalities in a small adaptive network of conductance-based neurons, receiving input patterns with different degrees correlation . The models for the neurons, synapses and plasticity rules (STDP) have a common biophysics basis. The neural network is simulated using a mixed analog-digital platform, which performs real-time simulations. We describe the study context, and the models for the neurons and for the adaptation functions. Then we present the simulation platform, including analog integrated circuits to simulate the neurons and a real-time software to simulate the plasticity. We also detail the analysis tools used to evaluate the final state of the network by the way of its post-adaptation synaptic weights. Finally, we present experimental results, with a systematic exploration of the network convergence when varying the input correlation, the initial weights and the distribution of hardware neurons to simulate the biological variability.

Keywords: Neuromorphic Engineering, Silicon Neurons, STDP, Hodgkin-Huxley Model.

1 Introduction

Since 1949 and the proposition of the plasticity role in behavior and cognitive skills explanation by Donald Hebb [1], we try to understand humans and animals high capabilities through the exploration of neural networks activity. Nowadays, plasticity has been formalized, from physiological observations, into rules related to synaptic transmission modulation efficacy. These rules link synaptic modifications to correlations in the firing activity of pre- and postsynaptic neurons. A mainly used model for describing the adaptation temporal mechanisms is the spike-timing dependent plasticity rule (STDP) in the cortex [2]. The STDP model is derived from studies exhibiting Long Term Potentiation (LTP) and Long Term Depression (LTD) phenomenon [3].

Several models of STDP display enhancement of synaptic efficacy when neurons are synchronized [4]. In parallel, activities visualized with techniques such

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 140–148, 2009.
© Springer-Verlag Berlin Heidelberg 2009

as EEG of fMRI show the result of a synchronized activity of neurons implicated in a same task. Thus, neurons do receive correlated inputs in situations where the subject is performing a learning task. Because a cortical neuron receives a large number of synapses, the set of its inputs can be modeled by a noise input. Studies have investigated the effect on single cells of the correlation factor of the noise on synaptic strength convergence driven by STDP [3].

It has been recently shown [5] that in an all-to-all connected small excitatory neural network, where neurons receive individual correlated noise input patterns, the convergence of synaptic weights driven by STDP depends on the noise input correlation. For a low correlation, the weights evolution was confined in a limited range. When all input patterns are identical (maximum correlation factor), the convergence becomes bimodal.

In this paper, we will show that if we have in a network two groups of neurons receiving noise input with two different correlation factors, weights convergence does not depend only on this factor but also on the direction of the synapse regarding to the two groups. The study was realized with a mixed hardware-software instrumentation tool, where neurons are implemented on analog VLSI circuits [6].

In section 2, we define the models used for neurons, synapses, plasticity and noise correlation. In section 3, we present the hardware and the method used to determine biophysical equivalence to the parameters of the hardware parameters. In section 4, we present the simulation configuration and the analysis tools. In section 5, we detail the experimental results and their analysis. As a conclusion, we discuss the results of these experiments.

2 Models

2.1 Neuron and Synapse Models

Neuron. The neuron model is based on the Hodgkin and Huxley formalism describing the dynamics of neuron's ionic currents [7]. Four voltage-dependent ionic currents are implemented in our model: I_{Na+}, I_{K+}, I_{LEAK} and I_{MOD}, a modulating slow voltage-dependent potassium current. The modeled neuron is the glutamate excitatory regular spiking neuron [8], and is characterized by its static

Table 1. Ionic channels parameters for the implemented model relative to a neuron with a membrane area of $22.10^{-5}cm^2$

Leak	$I_{LEAK} = g_{LEAK}(V_{MEM} - E_{LEAK})$ with $g_{LEAK} = 33nS$, $E_{LEAK} = -80mV$
Na	$I_{Na} = g_{Na}m^3h(V_{MEM} - E_{Na})$ with $g_{Na} = 11\mu S$, $-E_{Na} = 50mV$ $m : V_{OFFSET} = -37\,mV$, $V_{SLOPE} = 7,2\,mV$, $h : V_{OFFSET} = -42\,mV$ $V_{SLOPE} = -4,6\,mV$, $\tau(m) = 0,03\,ms$, $\tau(h) = \begin{cases} 3{,}00ms\ if\ V_{MEM}>0 \\ 0{,}25ms\ if\ V_{MEM}<0 \end{cases}$
K	$I_K = g_K n^4(V_{MEM} - E_K)$ with $g_K = 1,1\mu S$; $E_K = -100mV$ $n : V_{OFFSET} = -37mV$, $V_{SLOPE} = 11,38mV$, $\tau(n) = 3ms$
Mod.	$I_{MOD} = g_{MOD}p(V_{MEM} - E_{MOD})$ with $g_{MOD} = 10nS$, $E_{MOD} = -100mV$ $p : V_{OFFSET} = -35mV$, $V_{SLOPE} = 11,4mV$, $\tau(p) = \begin{cases} 300ms\ if\ V_{MEM}<0 \\ 8ms\ if\ V_{MEM}>0 \end{cases}$

parameters as time constants, potential offsets, conductance values (table 1), and by its functional features as f(I) curves and spike-frequency adaptation.

The neuron model parameters are listed in table 1. m, n, p in case of activation and h in case of inactivation are state variables (s), describing the state of ionic channels, defined by equation 1.

$$\tau(V_{MEM}).\frac{\delta s}{\delta t}(t) = \frac{1}{1+\exp(\pm\frac{V_{MEM}-V_{OFF}}{V_{SLP}})} - s(t) \tag{1}$$

where V_{MEM} is membrane potential, V_{OFF} and V_{SLP} are the offset and the slope of the sigmoid. The sign in the exponential is - for activation and + for inactivation.

Synapse. The synapse conductance-based model used is the kinetic synapse model presented in [9]. The synaptic current I_{syn} is given by:

$$I_{syn}(t) = \overline{g}_{syn} \cdot r(t) \cdot [V_{MEM} - E_{syn}] \text{ with } \frac{\delta r}{\delta t} = \alpha[T](1-r) - \beta r \tag{2}$$

where g_{syn} is the maximum conductance of the synapse, $r \in [0-1]$ represents the part of channels opened by neurotransmitters, V_{MEM} is the postsynaptic membrane potential, E_{syn} is the synaptic reversal potential. $[T]$ is the neurotransmitters concentration. α and β are rate constants for transmitters binding.

This model describes the synaptic strength as the duration of postsynaptic receptors opening (AMPA receptors for excitatory synapses). A pulse length represents the conductance increase due to the release of transmitters [6]. Presynaptic events are gathered on a single input to generate the synaptic pulses, applied to the post-synaptic neuron.

2.2 Plasticity: STDP Model

The STDP algorithm implemented for the synaptic adaptation is a biophysical model [10]. The equation representing the synaptic strength change depending on spike timing is:

$$\frac{d\omega_{ji}}{dt} = \epsilon_j\epsilon_i\Big[(\omega_{ji}-\omega_{LTP})\sum_k P[t-\tilde{t}_j(t)]\delta(t-t_{i,k}) \\ + (\omega_{ji}-\omega_{LTD})\sum_l Q[t-\tilde{t}_i(t)]\delta(t-t_{j,l})\Big] \tag{3}$$

where ω_{ji} is the synaptic strength from neuron j (presynaptic) to i (postsynaptic). $t_{i,k}$ and $t_{j,l}$ are respectively the sets of post- and presynaptic spikes times. P and Q are respectively the amount of LTP (potentiation) and LTD (depression) change, given by: $P(t) = A_+ \exp(t/\tau_P)$ and $Q(t) = A_- \exp(t/\tau_Q)$. ϵ_k are functions taking into account spikes history of a neuron and are given by $\epsilon_k = 1 - exp[-(t-\tilde{t}_k(t))/\tau_{\epsilon_k}]$. ω_{LTP} and ω_{LTD} are respectively maximal and minimal soft bounds. $\tilde{t}_k$ is the neuron's k last spike time. Parameters, bio-inspired, for exponential constants are $A_+ = 0.1$ and $A\_ = 0.005$. Time constants are $\tau_P = 14.8ms$ and $\tau_Q = 33.8ms$. The eligibility ϵ (influence of previous spikes of a same neuron), has an exponential time constant for the presynaptic neuron $\tau_{\epsilon_j} = 28ms$ and for the postsynaptic neuron $\tau_{\epsilon_i} = 88ms$ [11]. This takes into account features like frequency dependence and spike triplets.

2.3 Input Correlation

Neurons in the cortical area are connected to thousands of other neurons. Hence, one neuron is constantly stimulated and its spiking activity can be considered as randomized with a Poisson rule distribution. The synchrony induced by temporal coincidence of environmental events can lead some neurons to receive correlated stimulations. In our experiments we consider the noise input stimulating a neuron as a unique spike train input pattern. This pattern can be correlated with other input patterns stimulating neurons in the same network.

The noise patterns P applied to neurons are generated from a Poisson distribution Y and correlated with a defined factor ε (see (4)). First we generate a Poisson distribution X: $X = \{x_1, ..x_n\} \ / \ x_i = N(0,1).\sqrt{m - 1/2} + m$ where $N(0,1)$ is a normal distribution, m the average. Then X is converted in an absolute time pattern Y: $X \rightarrow Y = y_1, ..y_n \ / \ y_i = \sum_{j=1}^{i} x_j$.

The noise input pattern P_i for neuron i are generated as one event for each Y event. The time-lap between the event Y and a P_i pattern event is given by ε:

$$\varepsilon = N(0,1).(\alpha - 1).\frac{T}{6} \qquad (4)$$

where $N(0,1)$ is a normal distribution, T is the average period and $\alpha \in [0,1]$ is the correlation factor.

3 Material

3.1 System Description

We used a hardware real-time simulation environment for this study. The four neuron's ionic currents I_{Na}, I_K, I_{LEAK}, I_{MOD} are implemented in configurable analog VLSI circuits. The ionic channels modulate the neuron's membrane potential, measured on capacitors connected externally to the circuits. The synapse model is also built in the analog circuits. The neural network connectivity is driven by a custom hardware-software system [12]. The whole system is embedded in a computer through a PCI interface. The computer is in charge of the hardware configuration and of the STDP computation; the hardware computes the ionic and synaptic currents in real-time. The software features are: processor Intel Pentium 4(R), dual core, 2.6GHz, cache: 512Ko, SDRAM: 1Go. The operating system is the Ubuntu(R) Linux system.

In the hardware system, stimulation currents are coded by digital values that can slightly differ from one neuron to the other to trigger a same spiking frequency. This phenomenon is due to the mismatch and variations in the VLSI circuits fabrication process. We use the f(I) curves to benchmark the circuits and tune the stimulation parameters, as well as the VLSI neuron model parameters. The measured f(I) curves match the software simulations of the corresponding model. Differences exist concerning origin and scale values for the current range of the f(I) curves. Spike frequency adaptation shape observed on raster-plots is consistent with biological data.

3.2 Determining Biophysical Equivalence for the Hardware

Stimulation currents applied to the neurons have to be tuned to set the resting state of the cells (values in range [0-4095]). Synaptic conductances have also to be adjusted in range [0-255]. The translation rules for these values between the VLSI parameters and the models are linear, but can differ between the VLSI samples. To adjust for each VLSI these translation rules we developed a neuron model using the software NEURON [13] corresponding to the ideal VLSI neuron model. The morphology of the neuron is a cylinder section of diameter 96 nm and length 73 nm. Having equivalent models, we developed a protocol to define the biophysical equivalent to the digital parameters values used in our hardware system. With that protocol application, to deduce biophysical values equivalent to hardware values, we obtained for each neuron, (a, b) and (a', b') respectively for synaptic conductances and stimulation currents equations: $\omega_{BIO} = a.\omega_{HARD} + b$ and $I_{BIO} = a'.I_{HARD} + b'$.

4 Method

4.1 Experiments Configuration

The experiments were done on a neural network of six excitatory regular spiking neurons, with all-to-all synaptic connections (see A in Fig. 1). All connections followed the STDP rule defined in § 2.2, with ω_{LTP} fixed in order to avoid any bursting activity. All noise input patterns were rated at 5Hz. The neurons were tuned (by applying a sub-firing-threshold constant current) in order to trigger 3Hz oscillations when excited by the noise patterns, applied on the multi-synaptic inputs of the VLSI circuits. The stimulation currents applied to neurons were in the range equivalent to [0.4nA-0.5nA] (see § 3.2). Two groups of neurons were having input noise correlated. Initial synaptic weights were randomized with two initial states in the range equivalent to [0nS-20nS] (see § 3.2). For the experiments, we divided the network in 2 groups of varying size, between 2 and 4. We varied the inputs patterns correlations of Group 2 and Group 1 with $\alpha_{Group1} >= \alpha_{Group2}$ (α from (4)). We also used two different sets of initial randomized weights. In total, almost 1000 simulations were realized.

The duration of each simulation is 300s, corresponding to 300s of biological time due to the real-time characteristics of the VLSIs. The data recorded are for

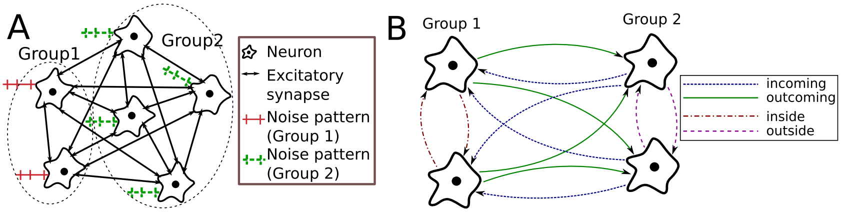


Fig. 1. A. Network of 6 neurons divided in 2 groups. **B.** Synapses categories relatively to the neurons groups.

each neuron the timing of each spike, and the timing and value of each synaptic weight modified by STDP.

4.2 Analysis Method

Synapse Directions. We consider four categories of synapses relatively to the 2 groups of neurons (see B of Fig. 1). Synapses connecting the neurons of Group 1 to the neurons of Group 2 are named outgoing synapses; those connecting the neurons of Group 2 to the neurons of Group 1 are named incoming synapses. Synapses connecting the neurons inside Group 1 are named inside synapses, and inside Group 2 outside synapses.

Representing Weights Kind of Convergence. To investigate the weight convergence, we build weights distribution histograms. A such histogram is then

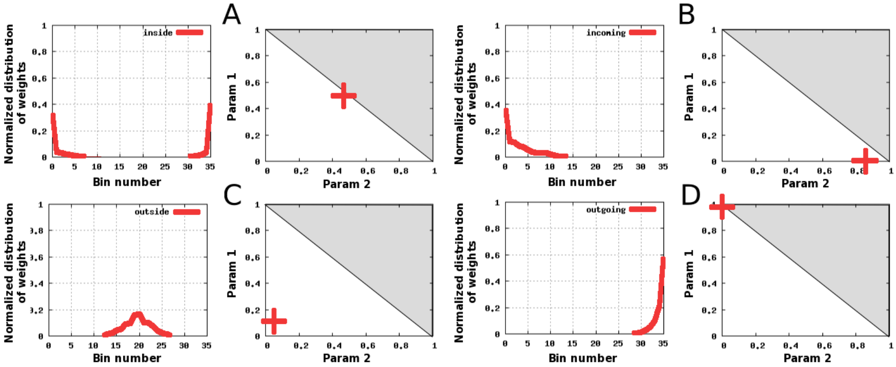


Fig. 2. For each group of plots, left is the normalized weights distribution histogram, right is the *param*1 vs *param*2 plot. *A*: bimodal convergence, $(x, f(x))$ tends to (0.5, 0.5). *B*: minimized convergence, $(x, f(x))$ tends to (1, 0). *C*: confined convergence, $(x, f(x))$ tends to (0, 0). *D*: maximized convergence, $(x, f(x))$ tends to (0,1). The colored part of each right graph is unreachable.

converted into a graph representing the kind of distribution using two weighting parameters $param1$ (linked to high weight values) and $param2$ (linked to low weight values). Indeed, simulations showed four kinds of post-STDP weight distributions (see Fig. 2, left graphs in A, B, C and D). To analyze several results in the same time, we want to represent these categories with a single element such as a point in 2D-graph. We realized two weighted sums of the distribution to obtain the characteristic point (right graphs in Fig. 2). These graphs are then added to 3D-graphs showing the kind of convergence depending on the input correlation (Fig. 3).

5 Results

Figure 3 presents some experimental results when exploring the network convergence for different correlation values for Group 1 and Group 2. Figure 3 - A, C, E relate to the incoming synapses (from Group 2 to Group 1), Fig. 3 - B, D, F relate to the outgoing synapses. For high correlations of Group 1 noise inputs ($\alpha_{Group1} >= 0.8$), and when the number of neurons of Group 1 is greater than the number for Group 2, convergence is independent of α_{Group2} for incoming synapses, weights are close to their minimal values. The outgoing synapses weights are, at the opposite, close to their maximum. We also observed that when $\alpha_{Group1} = \alpha_{Group2} >= 0.7$, in the case where Group1 was counting 2 neurons (out of 6 neurons in the network), incoming synaptic weights were converging to their maximum and outgoing synapses to their minimum.

For high values of α_{Group2} (note that $\alpha_{Group1} >= \alpha_{Group2}$) the inside and outside synapses weights convergence are bimodal and for weak values of α_{Group2}

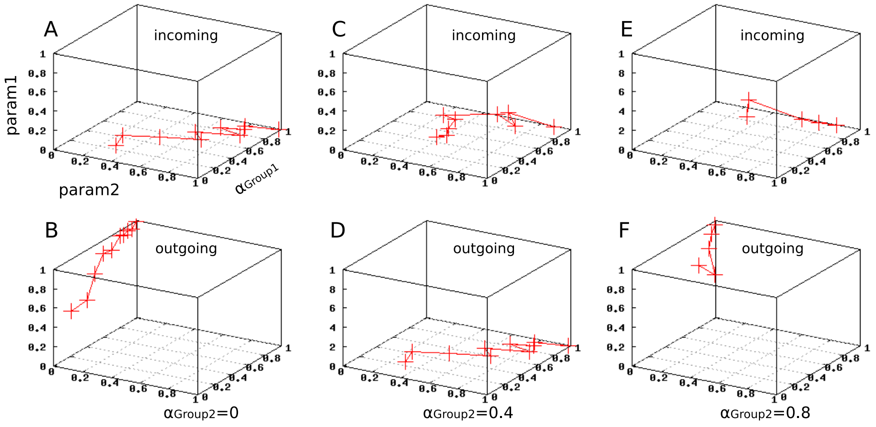


Fig. 3. Synapses convergence when varying α_{Group1} and α_{Group2}. Axis of the plots are: param1, param2, α_{Group1}. A and B: $\alpha_{Group2} = 0$; C and D: $\alpha_{Group2}= 0.4$; E and F: $\alpha_{Group2} = 0.8$; Top line is for incoming synapses; bottom line is for outgoing synapses.

and $\alpha_{Group1} <= 0.4$, weights are confined in a mean range of values. When α_{Group2} growths the convergence tends to minimal weights values for the incoming synapses, to maximal values for the outgoing synapses, and to both extrema (bimodal convergence) for the inside and outside synapses.

6 Conclusion

In this study, we have considered neurons subject to two independent groups of correlated random synaptic activity together with STDP at the synapses. When both amount of neurons and noise input correlation of the Group 1 are higher than the other group, STDP leads the distribution of weights to minimum values for incoming synapses, and maximum values for outgoing ones. This shows the leadership role of the group counting the more neurons and with a more important correlation in its inputs. Inside the groups, we noticed that when noise input correlation is important, synapses are discriminated and converge to extremal values. Otherwise, when the correlation of the group is weak, the synapses have a weight distribution in a range of mean values. We conclude that STDP enables the neuron to discriminated correlated groups within its inputs.

The use of analog VLSIs allows a large number of simulations in a real time, ie. faster than many digital implementation (software or hardware). We prepare experiments on larger networks, where computation time is clearly an issue.

Acknowledgements. Work supported by the European Community Grant FACETS (IST-2005-15879).

References

1. Hebb, D.O.: The Organization of Behaviour. John Wiley and Sons, Chichester (1949)
2. Markram, H., Lubke, J., Frotscher, M., Sackmann, B.: Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science 275, 213–215 (1997)
3. Song, S., Abbott, L.: Cortical development and remapping through spike timing-dependent plasticity. Neuron 32, 339–350 (2001)
4. van Rossum, M.C.W., Turrigiano, G.: Correlation based learning from spike timing dependent plasticity. Neurocomputing 38-40, 409–415 (2001)
5. Daouzli, A., Saïghi, S., Buhry, L., Bornat, Y., Renaud, S.: Weights convergence and spikes correlation in an adaptive neural network implemented on vlsi. In: Proc.s of the Int. Conf. on Bio-inspired Systems and Signal Processing (BIOSIGNALS), pp. 286–291 (2008)
6. Zou, Q., Bornat, Y., Saïghi, S., Tomas, J., Renaud, S., Destexhe, A.: Analog-digital simulations of full-conductance-based networks of spiking neurons with spike timing dependent plasticity. Network: computation in neural systems 17, 211–233 (2006)
7. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of Physiology 117, 500–544 (1952)

8. Connors, B., Gutnick, M.: Intrinsic firing patterns of diverse neocortical neurons. Trends in Neurosciences 13, 99–104 (1990)
9. Destexhe, A., Mainen, Z., Sejnowski, T.J.: An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. Neural Computation 6, 10–14 (1994)
10. Badoual, M., Zou, Q., Davison, A.P., Rudolph, M., Bal, T., Frégnac, Y., Destexhe, A.: Biophysical and phenomenological models of multiple spike interactions in spike-timing dependent plasticity. Int. J. Neural Syst. 16(2), 79–98 (2006)
11. Froemke, R.C., Dan, Y.: Spike-timing-dependent plasticity modification induced by natural spike trains. Nature 416, 433–438 (2002)
12. Renaud, S., Tomas, J., Bornat, Y., Daouzli, A., Saighi, S.: Neuromimetic ICs with analog cores: an alternative for simulating spiking neural networks. In: InternationaI Symposium on Circuits And Systems, pp. 3355–3358. IEEE, Los Alamitos (2007)
13. Hines, M.L., Carnevale, N.T.: The neuron simulation environment. Neural Computation 9, 1179–1209 (1997)

Adaptative Resonance Theory Fuzzy Networks Parallel Computation Using CUDA

M. Martínez-Zarzuela, F.J. Díaz Pernas, A. Tejero de Pablos,
M. Antón Rodríguez, J.F. Díez Higuera, D. Boto Giralda,
and D. González Ortega

Higher School of Telecommunications Engineering
University of Valladolid (Spain)
`mario.martinez@tel.uva.es`
`http://gti.tel.uva.es`

Abstract. Programming of Graphics Processing Units (GPUs) has evolved in a way they can be used to address and speed-up computation of algorithms exemplified by data-parallel models. In this paper parallelization of a Fuzzy ART algorithm is described and a detailed explanation of its implementation under CUDA is given. Experimental results show the algorithm runs up to 52 times faster on the GPU than on the CPU for testing and 18 times faster for training under specific conditions.

1 Introduction

In the last years, GPU (Graphics Processing Unit) programming has become a natural solution in order to speed-up the execution of different kind of algorithms [1]. In the beginnings of GPGPU (General Purpose computation on the GPU), algorithms had to be translated into graphics terms and programs were written using graphics APIs, such as OpenGL or DirectX and shading languages, such as Cg or HLSL [2]. Modern GPUs include new hardware resources and new frameworks like CUDA (Compute Unified Device Architecture) [3] or OpenCL (Open Computing Language) can be used for writing general purpose applications.

Adaptative Resonance Theory (ART) is a theory on how brain processes information. Fuzzy ART is an unsupervised artificial Neural Network (NN) capable of incremental learning, offering a great level of malleability [4]. It has been widely used in a universe of applications as medical sciences, economics and finance, engineering and computer science, or pattern recognition and classification. However, execution of Fuzzy ART on CPUs is a very time demanding task, thus is difficult to apply it for time dependent applications, such as computer vision.

Several researchers have shown great results in means of performance with their GPU-based artificial NNs implementations, speeding up execution dozens of times. Tze-Yui Ho et al. [5], explains the advantages and disadvantages of using a GPU-based simulation of a cellular neural network (CNN) instead of other methods such as VLSI circuits. The results show that the GPU-based

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 149–156, 2009.
© Springer-Verlag Berlin Heidelberg 2009

CNN simulator can run 8-17 times faster than a CPU-based CNN simulator, improving the performance at a low cost. Honghoon Jang et al. [6] designed an heterogeneous implementation for pattern recognition and image processing using CUDA for execution on a GPU and OpenMP for execution on a multicore CPU. The computational times for an algorithm of text detection showed about 15 times faster than an implementation using the CPU and about 4 times faster than an implementation running only on the GPU. In the area of Fuzzy ART, Martínez-Zarzuela et al. [7] presented the first GPU implementation of the algorithm using OpenGL, which speeds-up NN testing process.

This article describes how to adapt the original algorithm for data-parallel programming and optimize its execution under CUDA enabled platforms. Experimental results show both NN training and testing are faster on the GPU than on the CPU and a comparison with the aforementioned OpenGL implementation is included [7]. Section 2 gives a quickly overview of Fuzzy ART algorithm. In section 3, CUDA language and its main features are introduced. Section 4 focuses in the reformulation of the algorithm for parallel execution and its implementation using CUDA. Section 5, shows the time improvements achieved with respect to CPU and OpenGL implementations. Finally, section 6 draws the main conclusions and further researching tasks.

2 Fuzzy ART Networks

Fuzzy ART is a self-organizing neural network capable of incremental learning [4]. This kind of NN can be used to clusterize a stream of P input patterns into N different categories, generated during an unsupervised training phase. In Fuzzy ART systems, the first layer of neurons F_1 receives the input pattern and neurons in the upper layer F_2 represent a specific category from those emerged during the self-organizing training phase. F_1 activity vector is denoted by $\boldsymbol{I_p} = (I_1, \cdots, I_M)$ where each component I_i is within the $[0,1]$ interval. F_2 neurons sinapses are weighted by long-term memory (LTM) traces denoted by $\boldsymbol{w_j} = (w_{j1}, \cdots, w_{jM})$. Activity of these neurons is computed as $T_j(\boldsymbol{I}) = |\boldsymbol{I} \wedge \boldsymbol{w_j}|/(\alpha + |\boldsymbol{w_j}|)$ and the

```
foreach $I_p$ $(p = 1 \cdots P)$ in training set do
    foreach $w_j$ $(j = 1 \cdots N)$ in $F_2$ do
        Compute $T_j(I_p)$ ;
    end
    $T' = sort(T)$ descendingly;
    foreach $T'_k(I_p)$ $(k = 1 \cdots N)$ do
        if vigilance is met then
            update $w_k$;
            break;
        end
    end
    if no vigilance is met then
        new neuron in $F_2$;
    end
end
```

(a) Sequential training

```
foreach $I_p$ $(p = 1 \cdots P)$ in the testing set do
    foreach $w_j$ $(j = 1 \cdots N)$ in $F_2$ do
        Compute $T_j(I_p)$ ;
    end
    $T' = sort(T)$ descendingly;
    foreach $T'_k(I_p)$ $(k = 1 \cdots N)$ do
        if vigilance is met then
            $I_p \in k$ ;
            break;
        end
    end
    if no vigilance is met then
        $I_p \notin [1, N]$ ;
    end
end
```

(b) Sequential testing

Fig. 1. Trainining and testing Fuzzy ART NN pseudocodes

category choice is indexed by J, where $T_J = \max(T_j : j = 1 \cdots N)$. The system enters in *resonance* only if the *match function* meets the *vigilance criterion* $(|\boldsymbol{I} \wedge \boldsymbol{w_J}|)/|\boldsymbol{I}| \geq \rho$. When this occurs and learning is enabled ($\beta \neq 0$), associated $\boldsymbol{w_J}$ is updated $\boldsymbol{w_J^{new}} = \beta(\boldsymbol{I} \wedge \boldsymbol{w_J^{old}}) + (1 - \beta)\boldsymbol{w_J^{old}}$. Otherwise, node J is inhibited and the node in F_2 with the next higher activity is selected. If no neuron j is found to meet the vigilance criterion, a new neuron is committed in F_2. Figures 1(a) and 1(b) describe neural network computation on a sequential processor as the CPU.

3 Programmable Graphics Hardware Using CUDA

CUDA (Compute Unified Device Architecture) is a parallel computing framework created by NVIDIA for GPUs from the G8X series graphics cards and higher. C for CUDA includes some extensions to standard ANSI C99 language, giving access to the native instruction set and memory of the parallel computational elements in CUDA enabled GPUs. Accelerating an algorithm using CUDA includes translating it into data-parallel sequences of operations and mapping computations to the underlying resources to get maximum performance.

GPUs have a parallel many-core multiprocessor architecture. Each core is capable of running a series of operations *(kernel)* over several data simultaneously, delivering computations to thousands of GPU threads. Threads are grouped into a *block of threads*, that is executed on the same multiprocessor, thus allowing data sharing between threads through *shared memory.* Computations are delivered into a number of blocks called *grid.* When calling a GPU kernel from the CPU, dimensions of the grid and the block have to be specified. The design must be optimal to exploit hardware resources, being necessary to take into account aspects like the use of shared memory, coalescence or level of occupancy.

Shared memory is one of the most useful resources available in modern GPUs. The access to global memory suffers from a bottleneck effect, which can dramatically slow down the performance of an implementation. Shared memory is an intermediate cache between global memory and *stream processors*, as fast as registers. Threads belonging to the same block can store intermediate results of a kernel without spending a huge amount of time reading from or writing into the GPU global memory. The amount of shared memory is quite limited and has to be shared among all the threads of each block. A *coallesced* reading or writing performed by various threads simultaneously, can speed up the process, although it has to be specifically designed depending on the implementation. Designing the kernel for the execution of more than one block per multiprocessor can also help hiding latency in global memory loads. Execution of threads belonging to different blocks can overlap, but only if there are enough hardware resources for two or more blocks in the multiprocessor.

A synchronization execution point for the threads belonging to the same block can be needed in specific parts of the code. In *reduction operations*, for example, threads collaborate to compute a global operation over the elements of a vector, such as the maximum or the sum-up of every component. For parallelization, different threads compute a small part of the operation in several stages and algorithm converges on a result on the first position of the vector (figure 5(a)). Synchronization points between stages guarantees the consistence of the data.

4 Parallel Computation of Fuzzy ART Using CUDA

In this section, we describe how to adapt the Fuzzy ART algorithm for data-parallel computation. Also, an explanation of main issues to bear in mind when implementing the code in CUDA is included.

Figure 2(a) contains the Fuzzy ART parallelized version for training. Every input vector I_p has to be sequentially analyzed, since learning process can modify the weights associated to previous committed categories. However, it is possible to simultaneously compute the *match criterion* and the activity of every output neuron. The winning neuron is the most fired one that also matches this criterion. Figure 2(b) describes parallelization of the testing phase. In this case every input vector I_p can be processed in parallel and the activity of nodes in F_2 are sequentially computed.

The main advantage of using a graphics card is the possibility to execute the same operation on various vectors simultaneously. In figure 3 data organization in CUDA is depicted. Single-dimension blocks and grid are used in order to simplify vector accesses. A grid of blocks is generated to simultaneously deal with neural network weights on training and with the input patterns on testing, as described in pseudocodes in figures 2(a) and 2(b). On each block various vectors are included. Padding with zeros block and grid data might be necessary so that coalesced readings and coherent reductions can be done during processing. Dimensions of the grid are dynamically computed during the training, so that depending on the number of committed categories by the neural network, more blocks of threads are generated.

Training phase in figure 2(a) is computed in 3 different kernels: labelled κ_1, κ_2 and κ_3. This division of the training is made because each kernel works with vectors of different dimension, what means that for an optimal performance it is

```
         foreach $I_p$ $(p = 1 \cdots P)$ in training set do
$\kappa_1$     forall $w_j$ $(j = 1 \cdots N)$ in parallel do
                 if vigilance is met then
                     Compute $T_j(I_p)$ ;
                 else
                     $T_j = 0$;
                 end
             end
$\kappa_2$     $T_J = \max(T_j)$ ;
$\kappa_3$     if $T_J \neq 0$ then
                 update $w_J$;
             else
                 new neuron in $F_2$;
             end
         end
```

(a) Parallel training

```
$\kappa_4$ forall $I_p$ $(p = 1 \cdots P)$ in parallel do
         $T_J = 0$;
         foreach $w_j$ $(j = 1 \cdots N)$ in $F_2$ do
             if vigilance is met then
                 Compute $T_j(I_p)$ ;
                 if $T_j > T_J$ then
                     $T_J = T_j$;
                     $I_p \in j$ ;
                 end
             end
         end
         if $T_J == 0$ then
             $I_p \notin [1, N]$ ;
         end
     end
```

(b) Parallel testing

Fig. 2. Trainining and testing parallelized Fuzzy ART NN pseudocodes

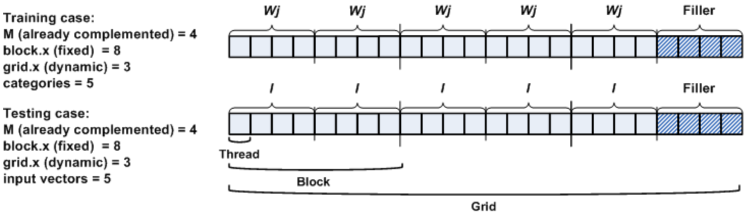
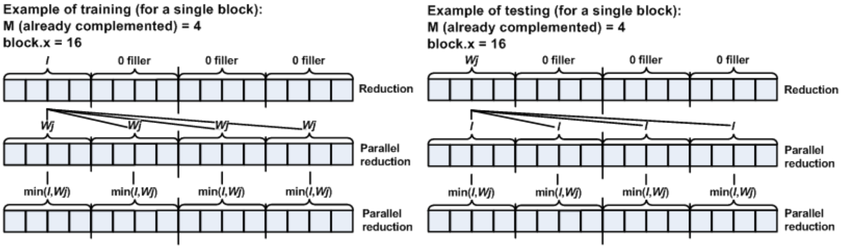


Fig. 3. Grid and kernel data organization for training and testing phases



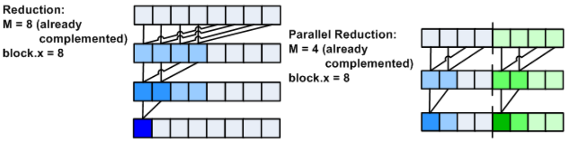
(a) Training (b) Testing

Fig. 4. Vector handling for parallel training and testing operations inside a block

necessary to call the kernels with their respective values of block and grid; but the real obstacle is that the only way to synchronize all the threads after the operation performed in the second kernel is calling another kernel, thus algorithm can not be implemented in a single function call. Testing phase in figure 2(b), on the other hand, can be computed in a single call to kernel κ_4.

Activities vector is declared as a `float2` value, which stores activity T_j and the associated j index. In order to compute the norm of a vector, reductions are needed in κ_1 and κ_4. During training, finding T_J involves another reduction in κ_2 to find the neuron in F_2 with the largest activity. If $T_J = 0$, then another neuron in F_2 is generated and dimensions of the grid may change if another block is needed to store its associated weights. Because of this parallelization, the number of vectors has to be adapted to fit into the number of blocks defined, so the last block can work with all its threads. This happens with weight vectors on training and input vectors on testing and the solution is as easy as adding vectors filled with $0s$ until the last block is filled (figure 3).

As various vectors can be stored on a single block, an specialized kind of reduction has to be done, in order to compute the norm of every one independently. This kind of *parallel reduction* is described in figure 5(b). This way, various reductions can be calculated simultaneously (figure 4). One remarkable technique which helps speeding-up reductions is *unrolling kernel loops* [8]. When applied for reduction operations, the size of the block in each iteration of the algorithm can be adjusted to the number of needed threads. There are a couple of restrictions when using this technique: reductions have to be made on vectors whose



(a) Reduction over a single vector (b) Reduction over various vectors

Fig. 5. Vector handling for reduction operations inside a block

size is a power of two to have a coherent result, because of the continuous halving in the process. Reductions have to use all the threads of the block; therefore, the size of the vector to reduce has to be equal to the number of threads. These two restrictions are easily solved in our program adjusting the size of the vector with 0s, as it does not alter the result of the operations.

5 Experimental Results

This section includes a performance comparison among the Fuzzy ART implementation using CUDA, a GPU implementation using OpenGL [7] and a CPU C++ implementation. Timings were taken on a dual-core 3.2 GHz Pentium 4 with 1GB RAM and different graphics cards belonging to G8X series from NVIDIA and include times for copying GPU results to the CPU. A synthetic *benchmark* comprised of several sets of P patterns was generated. In each set, dimension of input vectors I_p and the number of expected categories N varies. In order to guarantee the number of categories that could be committed was not too much influenced by the length and number of input patterns, a *Multivariate Normal Distribution* was used for pattern generation, so that $P = NxM$, being M the number of patterns expected to be included in one of N different categories [7]. Tests have been done for I_p dimensions of 4, 8, 16 and 32. The block size has been chosen to get the higher occupancy on the GPU. More occupancy means less idle threads while executing the program. Given the number of registers and shared memory needed for the kernels in the program, a block size of 128 is the optimal in most of the cases, and has been chosen for all the tests presented in this section.

The semi logarithmic plot in figure 6 compares the time needed for training the network on the CPU and on the GPU for $P = NxM$ patterns of length 32. Parallelization of the algorithm becomes useful when the number of committed categories is very large and so is the number of weight vectors. Otherwise, most of the time there will be a big number of idle threads. The CPU is faster for less than $N = 1500$ categories. When N is larger, the relative speed-up achieved on the GPU becomes exponential. During tests, a maximum speed-up of 18 was obtained for $N = 10000$.

Testing, on the other hand, is always much faster on the GPU than on the CPU, because it is always possible to process in parallel a huge amount of input

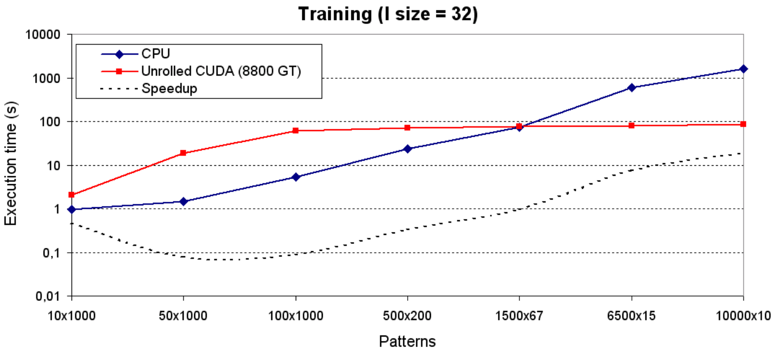
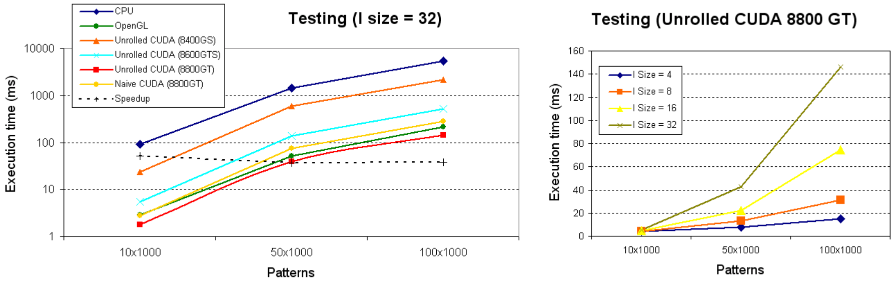


Fig. 6. Training for I_p of size 32



(a) Testing on different platforms for I_p of size 32 (b) Testing on 8800GT

Fig. 7. Performance during testing

vectors. Figure 7(a) shows testing performance under different platforms for patterns of length 32. Worth to mention is that a naïve CUDA implementation, not using the advanced *unrolling technique* does not manage to win the OpenGL implementation on a 8800GT GPU equipped graphics card. The speedup achieved for the best GPU implementation with respect to CPU varies between 37 and 52. In figure 7(b) a performance comparison depending on the dimension of the input patterns is included.

6 Conclusions and Future Work

In this paper, parallelization of a Fuzzy ART NN algorithm for training and testing was presented. An implementation of the algorithm on CUDA was detailed and tested against OpenGL [7] and CPU implementations. While a first naïve CUDA implementation was not as fast as the OpenGL version for testing, final version using unrolling loops technique considerably speed-up CUDA implementation. A peak speed-up of x57 is achieved for testing against the sequential version of the algorithm running on the CPU. For training, when the number of categories created in the NN is large enough to take advantage of all the GPU hardware resources, the relative speed-up between GPU and CPU grows exponentially.

Acknowledgments

This work has been partially supported by the *Spanish Ministry of Education and Science* under project TIN2007-67236 and by the *University Rey Juan Carlos* in collaboration with the *Community of Madrid* under project URJC-CM-2007-CET-1724.

References

1. Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., Purcell, T.J.: A survey of general-purpose computation on graphics hardware. Computer Graphics Forum 26 (2007)
2. Harris, M.: Mapping computational concepts to gpus. In: Pharr, M. (ed.) GPU Gems 2, pp. 493–508. Addison-Wesley, Reading (2005)
3. CUDA: Nvidia cuda zone: programming resources, `http://www.nvidia.com/object/cuda_home.html` (last visit, January 2009)
4. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4(6), 759–771 (1991)
5. Ho, T.Y., Park, A., Jung, K.: Parallelization of cellular neural networks on gpu. Pattern Recogn 41(8), 2684–2692 (2008)
6. Jang, H., Park, A., Jung, K.: Neural network implementation using cuda and openmp. In: DICTA 2008: Proceedings of the 2008 Digital Image Computing: Techniques and Applications, Washington, DC, USA, pp. 155–161. IEEE Computer Society Press, Los Alamitos (2008)
7. Martínez-Zarzuela, M., Díaz Pernas, F.J., Díez Higuera, J.F., Antón-Rodríguez, M.: Fuzzy art neural network parallel computing on the gpu. In: Hernández, F.S., Prieto, A., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 463–470. Springer, Heidelberg (2007)
8. Harris, M.: Parallel prefix sum (scan) with cuda. In: Nguyen, H. (ed.) GPU Gems 3, pp. 851–876. Addison Wesley Professional, Reading (2007)

A Supervised Learning Method for Neural Networks Based on Sensitivity Analysis with Automatic Regularization

Beatriz Pérez-Sánchez, Oscar Fontenla-Romero, and Bertha Guijarro-Berdiñas

Department of Computer Science
Faculty of Informatics, University of A Coruña
Campus de Elviña s/n, 15071, A Coruña, Spain

Abstract. The Sensitivity-Based Linear Learning Method (SBLLM) is a learning method for two-layer feedforward neural networks based on sensitivity analysis, that calculates the weights by solving a linear system of equations. Therefore, there is an important saving in computational time which significantly enhances the behavior of this method compared to other learning algorithms. In this paper a generalization of the SBLLM that includes a regularization term in the cost function is presented. The estimation of the regularization parameter is made by means of an automatic technique. The theoretical basis for the method is given and its performance is illustrated by comparing the results obtained by the automatic technique and those obtained manually by cross-validation.

1 Introduction

At present there exists many successful algorithms for training feedforward neural networks. Most of them are based on the classical gradient descent method proposed by Rumelhart et al. [1]. For this reason, many of these algorithms have two main drawbacks: convergence to local minima and slow learning speed. In [2] the authors described a new learning method that contributes to the solution of the second problem. The main innovations of this method, called the Sensitivity-Based Linear Learning Method (SBLLM), are: 1) the cost function that is minimized is based on the sensitivities of each layer's parameters with respect to its inputs and outputs and 2) the weights of each layer of the network are calculated by solving a linear system of equations. Thus, the method reaches a minimum error in few epochs of training and exhibits a higher speed when it is compared to other classical methods as demonstrated in [2]. In general, this behavior is very suitable in situations that deal with large data sets and networks. However, when the training set is not representative enough for the problem being solved, the few iterations employed by the method makes it very difficult to avoid overtraining by employing techniques like *early stopping*.

Another usual technique to avoid overfitting, which encourages smoother network mappings, is the regularization technique that consists in adding a penalty term to the loss function. In this paper, a generalized version of the SBLLM that

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 157–164, 2009.
© Springer-Verlag Berlin Heidelberg 2009

includes a regularization term in the cost function to be minimized is presented. This term is based on the well-known *weight decay* regularizer defined as the sum of squares of all adaptive parameters in the network. Moreover, a regularization parameter controls the influence of the regularization term in the cost function. In a previous work [6], the estimation of this parameter was made by means of a cross-validation method. This technique implies to establish manually a range of possible values of the regularization parameter (α) and running the algorithm for each of those different values. Usually, the curve that pictures the validation error versus the alpha's values monotonically decreases down to a certain optimum α and after this point the validation error increases due to the excessive influence of the regularization term in the cost function. Finally, the value of α that leaves to the minimum validation error will be chosen. This technique has an important drawback that is the high computational cost. Due to it, we have investigated different techniques that carry out an automatic estimation of the optimum value of the regularization parameter. Finally, we selected the approximation developed by Guo et al. [7]. This approach can be adapted for its application to the SBLLM in order to obtain a good approximation of α.

2 The Sensitivity Learning Algorithm with Regularization

Consider the feedforward neural network illustrated in Fig.1 where S is the size of the training data set, with I inputs, x_{is}, and J desired outputs, d_{js}.

The proposed learning method considers this network as composed of two subnetworks (see Fig. 1). The novelty of the SBLLM is that the weights of layers 1 and 2 are calculated independently by minimizing for each layer l a loss function,

$$Q^{(l)} = L^{(l)} + \alpha R^{(l)}, \quad (1)$$

where α is the regularization parameter that controls the influence of the regularization term in the cost function, and $R^{(l)}$ is the regularization term defined as $R^{(l)} = \sum_{i=0}^{M} \sum_{j=0}^{N} w_{ji}^2$, being M and N, respectively, the number of inputs and outputs of layer l. The term L measures the training error, that is usually

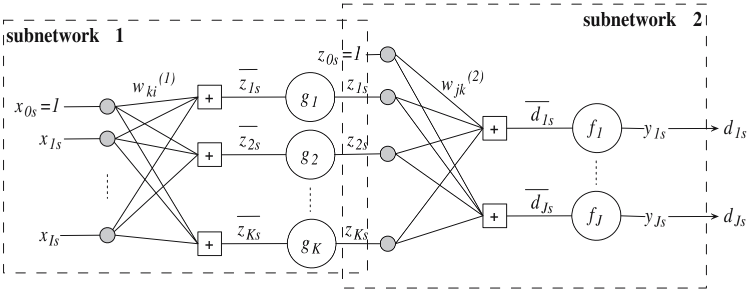


Fig. 1. Two-layer feedforward neural network

based on the mean squared error. The SBLLM considers L as the sum of squared errors *before* the nonlinear activation functions (g_k and f_j) instead of *after* them as it is usually done. Therefore, being z_{ks} the desired output for hidden neuron k and $\bar{z}_{ks} = g_k^{-1}(z_{ks})$, the alternative loss function used for solving subnetwork 1 can be written as

$$L^{(1)} = \sum_{s=1}^{S} \sum_{k=1}^{K} \left(g_k'(\bar{z}_{ks}) \left(\sum_{i=0}^{I} w_{ki}^{(1)} x_{is} - \bar{z}_{ks} \right) \right)^2 , \tag{2}$$

where $g_j'(\bar{z}_{ks})$ (and $f_j'(\bar{d}_{js})$ in equation 3) is a scaling term which weighs the errors. This term ensures that each error is magnified appropriately according to the operation point of the nonlinearity at the corresponding value of the desired data point. Analogously the loss function for the subnetwork 2 is defined,

$$L^{(2)} = \sum_{s=1}^{S} \sum_{j=1}^{J} \left(f_j'(\bar{d}_{js}) \left(\sum_{k=0}^{K} w_{jk}^{(2)} z_{ks} - \bar{d}_{js} \right) \right)^2 , \tag{3}$$

where d_{js} is the desired output for output neuron j and $\bar{d}_{js} = f_j^{-1}(d_{js})$. This loss function, that measures the error before the nonlinearity, was proposed in [3,4,5]. In these previous works it was shown that the optimum of this alternative loss function, up to first order of a Taylor serie, is the same as the one of the loss function that is obtained when the sum of squared errors *after* the nonlinear activation functions is employed. The advantage of the presented loss functions is that the optimum set of weights, for each layer, can be easily calculated by solving a system of linear equations that are obtained deriving $Q^{(1)}$ and $Q^{(2)}$ with respect to the weights and equating to zero. Another problem to be solved is to select in equation 1, the value of α that achieves the minimum error. We have investigated different techniques that carry out an automatic estimation of the optimum value of the regularization parameter. Finally, the approximation developed by Guo et al. [7] was selected. In this paper, they shown that one particular case of the system entropy reduces into the first-order Tikhonov regularizer for feedforward neural networks where the smoothing parameter in the kernel density function plays the role of the regularization parameter. Based on this fact, under the framework of the Kullback-Leibler distance, the authors derived a formula for approximately estimating the regularization parameter of a neural network using only training data. Considering this theoretical base the authors obtained an approximate estimation of α by using the sum-square-error and penalty term that can be applied to any adaptive learning algorithm. At the beginning of the training processing a small value of α is initalized and used to evaluate the weights, then it is periodically reestimated during training. The advantage of this result is that only applying training data is sufficient for estimating regularization coefficients, and α can be optimized on-line using the minimized generalization error. Considering the ideas previously commented, the proposed learning method is described.

Step 0: Initialization. Initialize weights and α with small random values. We assign to the outputs of the intermediate layer the output associated with some random weights $\mathbf{w}^{(1)}(0)$ plus a small random error,

$$z_{ks} = g_k\left(\sum_{i=0}^{I} w_{ki}^{(1)}(0)x_{is}\right) + \epsilon_{ks}; \quad \epsilon_{ks} \sim U(-\eta, \eta); k = 1, \ldots, K; s = 1, \ldots, S,$$

where η is a small number.

Step 1: Subproblem solution. The weights of layers 1 and 2 are calculated independently by minimizing for each layer l the loss function in equation 1, by solving the system of linear equations that are obtained deriving $Q^{(1)}$ and $Q^{(2)}$ with respect to the weights and equating to zero. That is:

$$\sum_{i=0}^{I} A_{pi}^{(1)} w_{ki}^{(1)} + \alpha w_{kp}^{(1)} = b_{pk}^{(1)}; \quad p = 0, 1, \ldots, I; \quad k = 1, \ldots, K$$

$$\sum_{k=0}^{K} A_{qk}^{(2)} w_{jk}^{(2)} + \alpha w_{jq}^{(2)} = b_{qj}^{(2)}; \quad q = 0, \ldots, K; \quad j = 1, \ldots, J,$$

where $A_{pi}^{(1)} = \sum_{s=1}^{S} x_{is}x_{ps}g_k'^2(\bar{z}_{ks})$; $b_{pk}^{(1)} = \sum_{s=1}^{S} \bar{z}_{ks}x_{ps}g_k'^2(\bar{z}_{ks})$; $\bar{z}_{ks} = g_k^{-1}(z_{ks})$
and $A_{qk}^{(2)} = \sum_{s=1}^{S} z_{ks}z_{qs}f_j'^2(\bar{d}_{js})$; $b_{qj}^{(2)} = \sum_{s=1}^{S} \bar{d}_{js}z_{qs}f_j'^2(\bar{d}_{js})$; $\bar{d}_{js} = f_j^{-1}(d_{js})$

Step 2: Evaluate the sum of squared errors. Using the new weights obtained in the previous step, the MSE is evaluated for the entire network and also the new cost function defined as,

$$Q(\mathbf{z}) = Q^{(1)}(\mathbf{z}) + Q^{(2)}(\mathbf{z})$$

This cost function measures the global errors of the network as the sum of the errors of each layer but calculated before the nonlinearities. Later on, based on this cost function the new values of the $\mathbf{z}$ will be obtained.

Step 3: Estimate the value of the regularization parameter α. Using the MSE of the previous step, the α value is reestimated. The employed formula is the adaptation of the approach of Guo et al. [7] for the SBLLM, $\alpha = I^2\left[1 + (I-1)^2\right]\frac{MSE}{W^2}$ where I is the number of inputs, MSE is the sum of squared errors calculated in step 2, S is the size of the training set and W^2 is the sum of the square of all the elements of the weight matrices of both layers.

Step 4: Convergence checking. If $|Q - Q_{previous}| < \gamma$ or $|MSE_{previous} - MSE| < \gamma'$ stop and return the weights and the sensitivities. Otherwise, continue with Step 4. Initially, the values of $Q_{previous}$ and $MSE_{previous}$ will be arbitrary large numbers.

Step 5: Check improvement of Q. If $Q > Q_{previous}$ reduce the value of ρ, and restore the weights, $\mathbf{z} = \mathbf{z}_{previous}$, $Q = Q_{previous}$ and go to Step 5. Otherwise, store the values $Q_{previous} = Q$, $MSE_{previous} = MSE$ and $\mathbf{z}_{previous} = \mathbf{z}$ and obtain the sensitivities of the cost function Q respect to the output $\mathbf{z}$ of the hidden layer, $\frac{\partial Q}{\partial z_{ks}} = \frac{\partial Q^{(1)}}{\partial z_{ks}} + \frac{\partial Q^{(2)}}{\partial z_{ks}}$ where, $\frac{\partial Q^{(1)}}{\partial z_{ks}}$ is defined as,

$$-2\left(g_k'(\bar{z}_{ks})(g^{-1})'(z_{ks})\left(\sum_{i=0}^{I} w_{ki}^{(1)} x_{is} - \bar{z}_{ks}\right)\right)\left(g_k'(\bar{z}_{ks}) - g_k''(\bar{z}_{ks})\left(\sum_{i=0}^{I} w_{ki}^{(1)} x_{is} - \bar{z}_{ks}\right)\right)$$

and

$$\frac{\partial Q^{(2)}}{\partial z_{ks}} = 2\sum_{j=1}^{J}\left(f_j'(\bar{d}_{js})\left(\sum_{r=0}^{K} w_{jr}^{(2)} z_{rs} - \bar{d}_{js}\right)\right) f_j'(\bar{d}_{js}) w_{jk}^{(2)}$$

being $\bar{z}_{ks} = g_k^{-1}(z_{ks})$, $\bar{d}_{js} = f^{-1}(d_{js})$, $k = 1, \ldots, K$, $j = 1, \ldots, J$ and $z_{0s} = 1, \forall s$

Step 6: Update intermediate outputs. Using the Taylor series approximation,

$$Q(\mathbf{z} + \Delta\mathbf{z}) = Q(\mathbf{z}) + \sum_{k=0}^{K}\sum_{s=1}^{S} \frac{\partial Q(\mathbf{z})}{\partial z_{ks}} \Delta z_{ks} \approx 0,$$

the following increments are calculated to update the desired outputs of the hidden neurons $\Delta\mathbf{z} = -\rho\frac{Q(\mathbf{z})}{||\nabla Q||^2}\nabla Q$, where ρ is a relaxation factor or step size. This procedure continues from Step 1 until a convergence condition is achieved. The complexity of this method is determined by the complexity of Step 1 which solves a linear system of equations for each network's layer. Several efficient methods can be used with a complexity of $O(n^2)$. Therefore, the resulting complexity of the proposed learning method is also $O(n^2)$, being n the number of weights of the network.

3 Experimental Results

In this section the proposed method is illustrated by its application to several regression problems. The Kobe data set was obtained at Time Series Data Library[1], the Dow-Jones index data set from StatLib Datasets Archive[2] and finally the Lorenz data set and the Laser K.U. Leuven time series were taken from the Eric´s Home page[3]. The characteristics of these data sets are summarized in Table 1. Although the data sets include greater number of samples, we have selected a subgroup of them to verify the effect of overfitting and the regularization parameter.

---

[1] Available at http://www.robjhyndman.com/TSDL

[2] Available at http://lib.stat.cmu.edu/datasets

[3] Available at http://www.cse.ogi.edu/∼ericwan/data.html

Table 1. Characteristics of the regression data sets

Data Set	Samples	No. inputs
Kobe (seismograph of the Kobe earthquake, recorded at Tasmania)	100	5
Dow-Jones (Dow Jones Industrial Average closing values)	100	8
Lorenz (chaotic time series generated by code)	500	8
Laser K.U. Leuven (laser time series from Santa Fe Competition)	250	8

In order to check the improvement of the proposed method (regularized SBLLM), all simulations were carried out using both the original SBLLM and the regularized version presented in this work. Moreover, for the regularized version we compare the results obtained by employing the automatic estimation of the regularization parameter α with those obtained manually by cross-validation. In order to be comparable, all learning algorithms were run under the same following conditions:

- The input data set was normalized (mean = 0 and standard deviation = 1). In addition, random noise $\epsilon \in N(0, \sigma)$ was added to the time series, where σ corresponds to the standard deviation of every specific data set.
- The logistic and linear functions were used for hidden and output neurons respectively. Regarding the number of hidden units of the networks it is important to remark that the aim was not to investigate the optimal one for each data set, but to check the added value of the regularization term. Therefore, the topologies were arbitrarily chosen to have a high number of weights with respect to the number of training samples. Finally, in all experiments the networks employed 10 hidden neurons.
- Step size equal to 0.2.
- In order to obtain statistically significant results, 20 simulations were performed using for each one a different initial set of values for $\mathbf{z}$, which were randomly generated from a normal distribution $N(0.5, 0.01)$ and subsequently normalized in the interval [0.05,0.95] to be in the output range of the activation functions. Also for each simulation the performance was calculated using 10-fold cross validation for all data sets.
- Finally, statistical tests were performed in order to check whether the differences in accuracy were significant among the compared algorithms.

In table 2, we present the performance of the method obtained with $\alpha = 0$ that is, without regularization. These values will be used as a reference to check the usefulness of the regularization term. In all cases, the performance is measured as the mean MSE over the 20 simulations. Table 3 shows the performance obtained by using the optimum value, that is the value of α that produces the minimum MSE, estimated manually by means of the cross-validation. Finally table 4 shows the values of α and the performance obtained by employing the automatic technique.

Table 2. Mean MSE ± Standard deviation for training and test using $\alpha = 0$

Data Set	**Mean Mse Train ± std**	**Mean MSE Test ± std**
Kobe	$2.64 \times 10^{-2} \pm 1.59 \times 10^{-2}$	$1.38 \times 10^{-3} \pm 1.40 \times 10^{-2}$
Dow-Jones	$2.31 \times 10^{-2} \pm 2.30 \times 10^{-3}$	$2.80 \times 10^{-3} \pm 1.02 \times 10^{-2}$
Lorenz	$7.94 \times 10^{-3} \pm 1.50 \times 10^{-3}$	$1.45 \times 10^{-4} \pm 1.20 \times 10^{-3}$
Leuven	$1.30 \times 10^{-2} \pm 2.40 \times 10^{-3}$	$3.52 \times 10^{-4} \pm 1.10 \times 10^{-3}$

Table 3. Mean MSE ± Standard deviation for training and test for regularized SBLLM (cross-validation)

Data Set	**Mean Mse Train ± std**	**Mean Mse Test ± std**	**Optimum α**
Kobe	$2.68 \times 10^{-2} \pm 1.59 \times 10^{-4}$	$7.27 \times 10^{-4} \pm 3.40 \times 10^{-5}$	1.43×10^{-3}
Dow-Jones	$2.38 \times 10^{-2} \pm 6.08 \times 10^{-5}$	$5.61 \times 10^{-4} \pm 8.33 \times 10^{-6}$	1.64×10^{-1}
Lorenz	$7.97 \times 10^{-3} \pm 1.20 \times 10^{-4}$	$6.67 \times 10^{-5} \pm 9.70 \times 10^{-4}$	9.59×10^{-4}
Leuven	$1.35 \times 10^{-2} \pm 1.99 \times 10^{-4}$	$4.43 \times 10^{-5} \pm 1.65 \times 10^{-5}$	1.87×10^{-3}

Table 4. Mean MSE ± Standard deviation for training and test for regularized SBLLM (automatic method)

Data Set	**Mean Mse Train ± std**	**Mean Mse Test ± std**	**Optimum α**
Kobe	$2.66 \times 10^{-2} \pm 1.14 \times 10^{-5}$	$8.66 \times 10^{-4} \pm 3.13 \times 10^{-5}$	1.43×10^{-3}
Dow-Jones	$2.37 \times 10^{-2} \pm 2.23 \times 10^{-5}$	$9.12 \times 10^{-4} \pm 4.56 \times 10^{-5}$	1.29×10^{-1}
Lorenz	$7.97 \times 10^{-3} \pm 1.48 \times 10^{-7}$	$8.62 \times 10^{-5} \pm 1.69 \times 10^{-6}$	4.56×10^{-4}
Leuven	$1.34 \times 10^{-2} \pm 4.03 \times 10^{-6}$	$8.17 \times 10^{-5} \pm 3.50 \times 10^{-6}$	1.98×10^{-3}

First, we made a comparative the SBLLM with and without regularization parameter. As it can be observed, in all cases the use of the regularization method improves the performance as was demonstrated by applying the Kruskal-Wallis test [8,9] to check that the differences in the MSE, over the test sets, were statistically significant. Moreover, when the regularization is applied, the experiments show that the automatically estimated regularization parameter is in the same order as that estimated manually by cross-validation. Being also the results obtained by the manual and the automatic methods in the same order, a remarkable advantage of the automatic method is that it evidently requires lesser computacional resources.

4 Conclusions

The generalized version of the algorithm, the regularized Sensitivity Based Linear Learning Method, improves the performance of the original method in those situations in which the overfitting is possible. However, the estimation of a suitable value for the regularization term has a high computational cost if it is calculated by means of cross-validation. For this reason, we have considered an alternative technique that allows us to obtain a good approach of the regularization parameter in less time.

In this paper we presented a new technique for the estimation of the regularization term, which is based on the approximation developed by Guo et al. [7]. The comparative study carried out allows us to establish that the automatic method obtains a good approach regarding both the performance and the optimum value of the parameter α with an important saving in computational time.

Acknowledgements

We would like to acknowledge support for this work from the Xunta de Galicia (project PGIDT08TIC012105PR) and the Ministerio de Ciencia e Innovación, Spain (project TIN2006-02402), partially supported by the European Union ERDF.

References

1. Rumelhart, D.E., Hinton, G.E., Willian, R.J.: Learning Representations of Back-Propagation Errors. Nature 323, 533–536 (1986)
2. Castillo, E., Guijarro-Berdiñas, B., Fontenla-Romero, O., Alonso-Betanzos, A.: A very fast learning method for Neural Networks Based on Sensitivity Analysis. Journal of Machine Learning Research 7, 1159–1182 (2006)
3. Castillo, E., Fontenla-Romero, O., Alonso-Betanzos, A., Guijarro-Berdiñas, B.: A Global Optimum Approach for One-Layer Neural Networks. Neural Computation 14(6), 1429–1449 (2002)
4. Erdogmus, D., Fontenla-Romero, O., Principe, J.C., Alonso-Betanzos, A., Castillo, E.: Linear-Least-Squares Initialization of Multilayer Perceptrons Through Backpropagation of the Desired Response. IEEE Transactions on Neural Network 16(2), 325–337 (2005)
5. Fontenla-Romero, O., Erdogmus, D., Principe, J.C., Alonso-Betanzos, A., Castillo, E.: Linear least-squares based methods for neural networks learning. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 84–91. Springer, Heidelberg (2003)
6. Guijarro-Berdiñas, B., Fontenla-Romero, O., Perez-Sanchez, B., Alonso-Betanzos, A.: A Regularized Learning Method for Neural Networks Based on Sensitivity Analysis. In: European Symposium on Artificial Neural Networks, pp. 289–294 (2008)
7. Guo, P., Lyu, M.R., Philip Chen, C.L.: Regularization Parameter Estimation for Feedforward Neural Networks. IEEE Transactions on System, Man and Cybernetics 33(1), 35–44 (2003)
8. Hollander, M., Wolfe, D.A.: Nonparametric Statistical Methods. Wiley, Chichester (1973)
9. Gibbons, J.D.: Nonparametric Statistical Inference, 2nd edn. Marcel Dekker, New York (1985)

Ensemble Methods for Boosting Visualization Models

Bruno Baruque[1], Emilio Corchado[1], Aitor Mata[2], and Juan M. Corchado[2]

[1] Department of Civil Engineering, University of Burgos, Spain
escorchado@ubu.es, bbaruque@ubu.es
[2] Department of Computer Science and Automatics, University of Salamanca, Spain
aitor@usal.es, corchado@usal.es

Abstract. Topology preserving mappings are great tools for data visualization and inspection in large datasets. This research presents a study of the combination of different ensemble training techniques with a novel summarization algorithm for ensembles of topology preserving models. The aim of these techniques is the increase of the truthfulness of the visualization of the dataset obtained by this kind of algorithms and, as an extension, the stability conditions of the former. A study and comparison of the performance of some novel and classical ensemble techniques, using well-known datasets from the UCI repository (Iris and Wine), are presented in this paper to test their suitability, in the fields of data visualization and topology preservation when combined with one of the most widespread of that kind of models such as the Self-Organizing Map.

Keywords: topology preserving mappings, boosting, bagging, unsupervised learning.

1 Introduction

From the range of tools that can be used to treat the high amounts of data that industrial and business operations processes, one of the most useful is the unsupervised leaning, in the field of artificial neural networks (ANNs). For unsupervised learning only the input and the network's internal dynamics are the two elements required. No external mechanism is used to obtain the results. The present work is centred on one of the major methods of unsupervised learning: competitive learning, where the output neurons of a neural network compete among themselves for being the one to be active.

The Self-Organising Map (SOM) [1] is probably the most widely used algorithm making use of this kind of learning. It is based on an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, or sets of samples in a specific domain of the input space. The SOM was conceived as a visualization tool to enable the representation of high-dimensional datasets on 2-dimensional maps and thereby facilitating data interpretation tasks for human experts.

The principal problem of the models based on competitive learning is, as happens with all ANNs, their instability. This means that even running the same algorithm several times with the same parameters can lead to rather different results. The present research is focused on the comparison and study of some novel and classical ensemble extension versions of the two competitive learning models based on the topology preserving concept. A novel summarization of topology preserving

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 165–173, 2009.
© Springer-Verlag Berlin Heidelberg 2009

ensembles is presented and included in this comparison. This algorithm aims to obtain a more trustful representation of the datasets by combining the best features of several trained maps. The summarization algorithms are tested for the SOM model in combination with two ensemble techniques such as the Bagging [2] and the AdaBoost [3]. The purpose of this comparison is to verify if the performance of these unsupervised connectionist models can be improved by means of these ensemble meta-algorithms. AdaBoost is applied for the first time in this paper in combination with the WeVoS algorithm.

2 Self-Organizing Maps

The Self-Organizing Map (SOM) algorithm [4] 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 [5]. Its aim is to produce a low dimensional representation of the training samples while preserving the topological properties of the input space.

The main feature of the SOM algorithm is that the neighbours on the lattice are also allowed to learn – i.e. to adapt their characteristics to the input - as well as the winning neuron. Thus, the neighbouring neurons gradually come to represent similar inputs, and their representations become ordered on the map lattice.

This updating of neighbourhood neurons in SOM can be expressed as:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t)\big(x(t) - w_v(t)\big) \tag{1}$$

where, w_k is the weight vector associated with neuron k; $\alpha(t)$ is the learning rate of the algorithm; $\eta(v,k,t)$ is the neighbourhood function (usually, the Gaussian function or a difference of Gaussians), in which v represents the position of the winning neuron in the lattice, or the best matching unit (BMU); k, the positions of neighbouring neurons and x , the network input.

3 Quality Measures

Several quality measures have been proposed in literature to study the reliability of the results displayed by topology preserving models in representing the dataset that have been trained with [6, 7]. There is not a global and unified one, but rather a set of complementary ones, as each of them asses a specific characteristic of the performance of the map in different visual representation areas. The three used in this study are briefly presented in the following paragraphs.

Classification Error [8]**.** Topology preserving models can be easily adapted for classification of new samples using a semi-supervised procedure. A high value in the classification accuracy rate implies that the units of the map 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 precisely.

Topographic Error [9]**.** It consists on finding the first two best matching units (BMU) for each entry of the dataset and testing whether the second is in the direct neighbourhood of the first or not.

Distortion [10, 11]**.** When using a constant radius for the neighbourhood function of the learning phase of a SOM; the algorithm optimizes a particular function. This function can be used to quantify in a more trustful way than the previous one, the overall topology preservation of a map by means of a measure, called distortion measure in this work.

Goodness of Map [12]**.** This measure combines two different error measures: the square quantization error and the distortion. It takes account of both the distance between the input and the BMU and the distance between the first BMU and the second BMU in the shortest path between both along the grid map units, calculated solely with units that are direct neighbours in the map.

4 Unsupervised Competitive Learning Ensembles

The ultimate goal of constructing an ensemble is to improve the performance obtained by a single working unit. When talking about classification it is generally accepted that 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 [13]. Many ensemble models and theories have been previously developed and have been applied mainly to models designed specifically for classification, especially supervised classifiers [14]. In the present study the central idea is to verify the improvements that an ensemble technique can provide in the multi-dimensional data visualization [15] field over an unsupervised learning process such as the Competitive Learning.

4.1 Bagging and AdaBoosting

Boosting meta-algorithms consists on training a simple classifier in several stages by incrementally adding new capacities to the current learned function. In the case of the present work the decision taken was to begin by implementing simpler boosting algorithm to initially study its effect on some topology preserving algorithms. Bagging and AdaBoost are the two boosting selected.

Bagging (or bootstrap aggregating) [2] is one of the simplest techniques for ensemble construction. It consists on training each of the classifiers composing the ensemble separately using a different subset of the main training dataset. This is accomplished by using re-sampling with replacement over the training set. The technique provides the ensemble with a balance between variability and similarity.

The idea of AdaBoost [3] is very similar to that of the Bagging. The difference is that it is taken into accounts which of the training samples are not correctly classified by the current classifier. When a sample is not well classified its associated probability is increased, so there are more chances that it will be presented to the next trained classifier as input. That way, the ensemble concentrates in the samples that are harder to classify, improving its learning capabilities. There have been proposed two slightly different versions of the algorithm [15]. AdaBoost.M1 is recommended for datasets with samples that belong to two different classes while AdaBoost.M2 is recommended for dataset with more than two different classes.

The Adaboost algorithm requires a measure of accuracy of classification of each of the components of the ensemble. Therefore, a semi-supervised learning technique [8] is applied in this case to enable its use under the frame of topology preserving models.

4.2 Summarizing Some Applied Ensembles Techniques

Several algorithms for fusion of maps have been tested and reviewed recently by the authors of this work [16, 17].

In this case, an algorithm devised by the authors of this work is used to generate the final network summarizing the results obtained by the different networks included in the ensemble. It is called Weighted Voting Summarization (WeVoS) [18]. As the SOM is mainly designed as visualization tools, it is desirable that a combination of several of this kind of maps presents a truthful representation of data for visual inspection, based in the parts of the maps that were representing that portion of the data space the most correctly. The WeVoS tries to achieve this by taking into account one of the most important features of these algorithms: topology preservation. To do so, it obtains the final units of the map by a weighted voting among the units in the same position in the different maps, according to a quality measure. This measure can be any of the previously presented or other found in literature, as long as can be calculated in a unit by unit basis.

Algorithm 1. Weighted Voting Superposition (WeVoS)

```
train several networks by using the bagging (re-sampling with replacement) meta-algorithm
for each map (m) in the ensemble
  for each unit position (p) of the map
    calculate the quality measure/error chosen for the current unit
  end
end
calculate an accumulated quality/error total for each homologous set of units Q(p) in all
  maps
calculate an accumulated total of the number of data entries recognized by an  homologous
  set of units  in all maps D(p)
for each unit position (p)
  initialize the fused map (fus) by calculating the centroid (w') of the units of all maps in
    that position (p)
end
for each map (m) in the ensemble
  for each unit position (p) of the map
    calculate the vote weight of the neuron (p) in the map (m) by using Eq. 2
    feed the weight vector of the neuron (p), as if it were a network input, into the fused
      map (fus), using the weight of the vote calculated in Eq. 2 as the learning rate and the
      index of that same neuron (p) as the index of the BMU.
      The unit of the composing ensemble ($w_p$) is thereby approximated to the unit of the
      final map (w') according to its weighting system.
  end
end
```

The voting process used is the one described in Eq. 2:

$$V_{p,m} = \frac{\sum b_{p,m}}{\sum_{i=1}^{M} b_{p,i}} \cdot \frac{q_{p,m}}{\sum_{i=1}^{M} q_{p,i}} \tag{2}$$

where, $V_{p,m}$ is the weight of the vote for the unit included in map m of the ensemble, in its in position p, M is the total number of maps in the ensemble, $b_{p,m}$ is the binary vector used for marking the dataset entries recognized by unit in position p of map m, and $q_{p,m}$ is the value of the desired quality measure for unit in position p of map m.

A detailed pseudo-code of the WeVoS algorithm is presented in Table 1.

5 Experiment Details

Several experiments have been performed to check the suitability of using the previously described boosting and combining techniques under the frame of the mentioned topology preserving models. The datasets selected are Iris and Wine that were obtained from the UCI repository [19].

For all the tests involving this combination of networks the procedure is the same. A simple n-fold cross-validation is used in order to employ all data available for training and testing the model and having several executions to calculate an average of its performance. In each step of the cross-validation first, an ensemble of networks must be obtained. The way the ensemble is trained does not affect the way the combination is computed. Then the computation of the combination is performed. Finally, both the ensemble and the combination generated from it are tested employing the test fold.

Visualization results are displayed in Fig.1 while analytical results appear in Fig.2. In Fig. 1 the maps obtained by the different combination of algorithms are displayed. Fig. 1 shows the results of applying a different ensemble algorithm to the same dataset with the same topology preserving algorithm. Both Adaboost.M1 and Adaboost.M2 have been tested for the sake of comparison taking into account that first algorithm can be applied also to multi-class datasets, although the second one should be more suitable. All SOMs in all ensembles showed where trained using the same parameters. Fig. 1(a) displays the map obtained by a single SOM. It contains three different classes, one of them (class one, represented by circles) clearly separated from the other two. Fig. 1(b) represents the summary obtained by the WeVoS algorithm over an ensemble trained using the bagging meta-algorithm. In this case, as all dataset entries are considered of the same importance in all iterations a smooth map is obtained. It is worth noting that classes are displayed in a more compact way than in the single SOM. Class 1 appears more distant to class 2 (squares) and class 2 and 3 (triangles) are more horizontally separated in the top of the image, although this separation is not so clear in the middle-left part of the image. Fig. 1(c) represents the map obtained from an ensemble trained on AdaBoost.M1 algorithm. As this algorithm tries to concentrate in difficult to classify classes, only one neuron is used in the final map to represent class 1, which is obviously the most easy to distinguish from the three of them. This can be considered a desired or not so desired result, regarding to what the final resultant network is going to be used. In the case of intending to use the map for classification purposes, this map can be considered more suitable then the single one. In the case of intending to use the map for visualization purposes, this could be considered quite the contrary. Finally, the result of using the Adaboost.M2 is showed in Fig. 1(d). As this algorithm uses a finer granularity for classification than the previous version, it again represents class 1 in a greater detail than the AdaBoost.M1 (Fig. 1(c)), but showing a bit more compact groups than the single algorithm (Fig. 1(a)) and showing a more clear separation of groups than the Bagging algorithm (Fig. 1(b)) does.

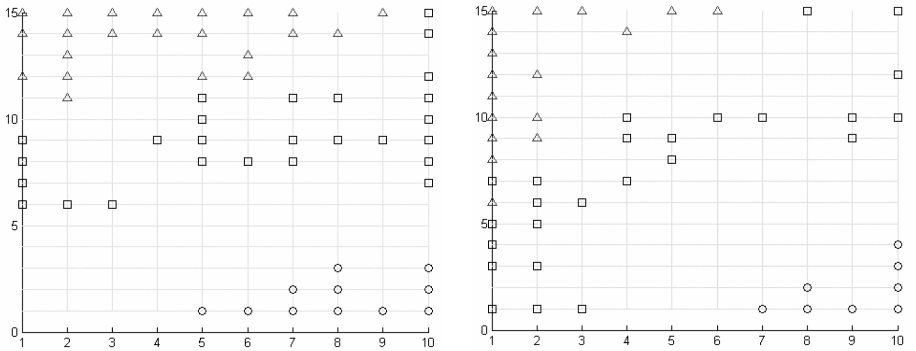


Fig. 1(a). Single SOM

Fig. 1(b). WeVos from an ensemble trained using the Bagging algorithm

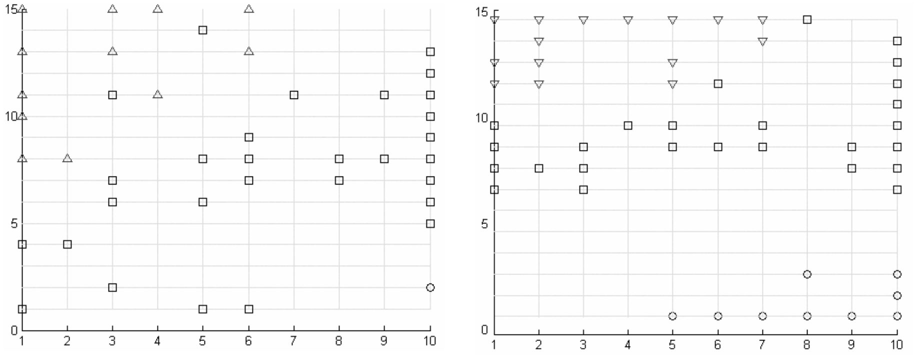


Fig. 1(c). WeVos from an ensemble trained using the AdaBoost.M1 algorithm

Fig. 1(d). WeVos from an ensemble trained using the AdaBoost.M2 algorithm

Fig. 1. This figure shows the 2D maps representing the Iris dataset. Each figure represents a map obtained by training the ensemble of SOM algorithms using a different meta-algorithm and the applying the WeVoS algorithm to all of them.

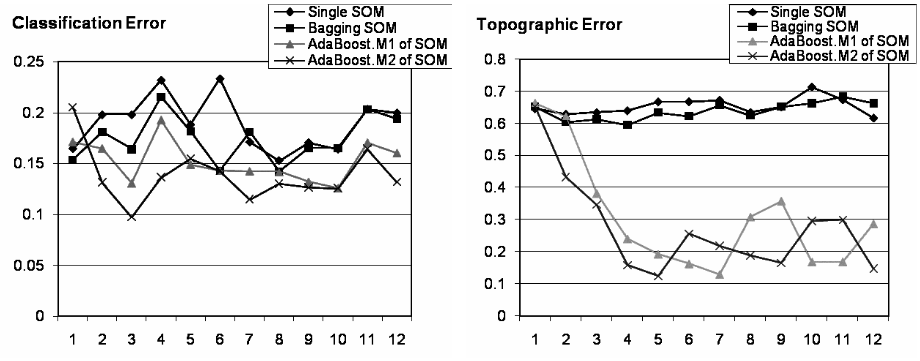


Fig. 2(a). Classification Error **Fig. 2(b).** Topographic Error

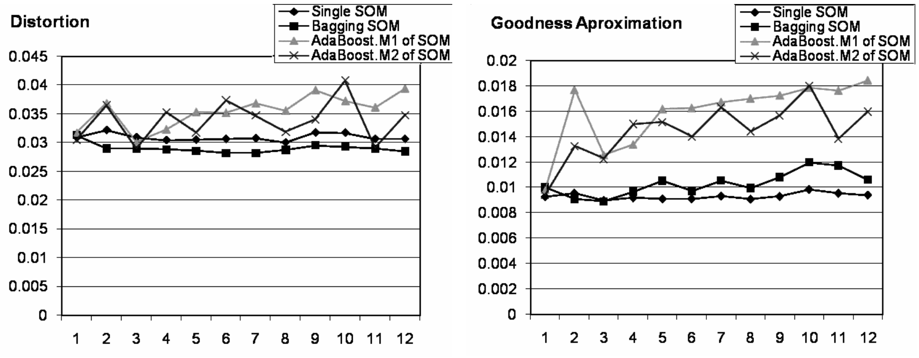


Fig. 2(c). Distortion **Fig. 2(d).** Goodness of Approximation

Fig. 2. Displays the quality of maps measures for each of the ensemble models presented, along with the corresponding single model. All models were trained and tested using the Wine dataset. The X-axis represents the number of single maps composing each of the ensembles and the Y-axis represents the value of each measure.

Fig. 2 represents the results of the calculation of the different quality measures previously described over the different algorithms described. All are error measures, so the desired value is always as close to 0 as possible.

In Fig. 2, the quality measure values obtained from the ensemble algorithms are measured from their corresponding WeVoS summarization, while data from single models is obviously obtained directly by that model. The WeVoS algorithm has previously been compared with other summarization methods (Fusion by Euclidean distance [20], Fusion by similarity in Voronoi polygons [21]), showing an improvement in its results [18]. The results of those models are not included in this work for the sake of brevity. Fig. 2(a) represents the classification error of the different variants under study. Obviously, as that is their original purpose, the maps obtained through ensemble algorithms outperform the single model. Fig 2(b) represents the topographic ordering of the final map. Again, ensemble models obtain lower errors than the single one, especially the two variants of AdaBoost. Fig 2(c) measures in a greater detail the topological ordering of the maps and shows a different situation than Fig. 2(b) for the AdaBoost algorithm. This points to an overfitting problem. Finally, Fig. 2(d) represents a measure combining quantization and distortion errors. In this case, the results for the two variants of AdaBoost were expected, as the algorithm tries to concentrate not on the whole dataset, but on the most difficult to classify samples of it, increasing the distance to other samples. The bagging algorithm obtains lower error, although a bit higher than the simple model. This is due to the nature of the WeVoS algorithm, which benefits the topology preservation of the summary over the quantization side of the model, being the first one of the most the characteristic features of the family of models under study.

6 Conclusions and Future Work

This study presents an interesting mixture of techniques for representation of multi-dimensional data in 2-D maps. They are based on the combination of several maps trained over slightly different datasets to form an ensemble of networks with self-organizing capabilities. This idea can be especially useful when a reduced dataset is available. The training of the ensemble of networks has been tested by using the bagging and boosting techniques for their comparison. As an ensemble of maps is impossible to represent, a summarization algorithm is also presented and used.

These techniques have been tested in two widely known real datasets. Future work will include far exhaustive testing of the presented combination of techniques using several more complex datasets, as well as adapting the present model to other novel boosting meta-algorithms to check if more improvements can be obtained. It is reasonable to think that, adapting the AdaBoost algorithm to enhance other capabilities of the SOM, such as topology preservation, these results can be improved.

Also the problem of the overfitting when using the AdaBoost algorithm will be subject of further analysis by authors of this paper.

Acknowledgments

This research has been partially supported through project BU006A08 by the Junta de Castilla y León.

References

1. Kohonen, T.: Self-organizing maps. Series in Information Sciences, vol. 30. Springer, Berlin (1995)
2. Breiman, L.: Bagging Predictors. Machine Learning 24, 123–140 (1996)
3. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
4. Kohonen, T.: The Self-Organizing Map. Neurocomputing 21, 1–6 (1998)
5. Kohonen, T., Lehtio, P., Rovamo, J., et al.: A Principle of Neural Associative Memory. Neuroscience 2, 1065–1076 (1977)
6. Pölzlbauer, G.: Survey and Comparison of Quality Measures for Self-Organizing Maps. In: WDA 2004, pp. 67–82 (2004)

7. Polani, D.: Measures for the organization of self-organizing maps. In: Seiffert, U., Jain, L.C. (eds.) Self-Organizing Neural Networks: Recent Advances and Applications. Studies in Fuzziness and Soft Computing, vol. 16, pp. 13–44. Physica-Verlag, Heidelberg (2003)
8. Vesanto, J.: Data Mining Techniques Based on the Self-Organizing Map, 63 (1997)
9. Kiviluoto, K.: Topology Preservation in Self-Organizing Maps. In: ICNN 1996, vol. 1, pp. 294–299 (1996)
10. Lampinen, J.: On Clustering Properties of Hierarchical Self-Organizing Maps. Artificial Neural Networks 2, II, 1219–1222 (1992)
11. Vesanto, J., Sulkava, M., Hollmén, J.: On the Decomposition of the Self-Organizing Map Distortion Measure. In: WSOM 2003, pp. 11–16 (2003)
12. Kaski, S., Lagus, K.: Comparing Self-Organizing Maps. In: Vorbrüggen, J.C., von Seelen, W., Sendhoff, B. (eds.) ICANN 1996. LNCS, vol. 1112, pp. 809–814. Springer, Heidelberg (1996)
13. Heskes, T.: Balancing between Bagging and Bumping. NIPS 9, 466–472 (1997)
14. Schwenk, H., Bengio, Y.: Boosting Neural Networks. Neural Computation 12, 1869–1887 (2000)
15. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm, pp. 148–156 (1996)
16. Baruque, B., Corchado, E., Yin, H.: Quality of Adaptation of Fusion ViSOM. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 728–738. Springer, Heidelberg (2007)
17. Corchado, E., Baruque, B., Yin, H.: Boosting Unsupervised Competitive Learning Ensembles. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 339–348. Springer, Heidelberg (2007)
18. Baruque, B., Corchado, E., Rovira, J., et al.: Application of Topology Preserving Ensembles for Sensory Assessment in the Food Industry, pp. 491–497 (2008)
19. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007) (2008)
20. Georgakis, A., Li, H., Gordan, M.: An Ensemble of SOM Networks for Document Organization and Retrieval. In: AKRR 2005, pp. 6–141 (2005)
21. Saavedra, C., Salas, R., Moreno, S., et al.: Fusion of Self Organizing Maps. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 227–234. Springer, Heidelberg (2007)

New Artificial Metaplasticity MLP Results on Standard Data Base[*]

Alexis Marcano-Cedeño, Aleksandar Jevtić, Antonio Álvarez-Vellisco,
and Diego Andina

Group for Automation in Signals and Communications,
Technical University of Madrid,
Madrid, Spain

Abstract. This paper tests a novel improvement in neural network training by implementing Metaplasticity Multilayer Perceptron (MMLP) Neural Networks (NNs), that are based on the biological property of metaplasticity. Artificial Metaplasticity bases its efficiency in giving more relevance to the less frequent patterns and subtracting relevance to the more frequent ones. The statistical distribution of training patterns is used to quantify how frequent a pattern is. We model this interpretation in the NNs training phase. Wisconsin breast cancer database (WBCD) was used to train and test MMLP. Our results were compared to recent research results on the same database, proving to be superior or at least an interesting alternative.

1 Introduction

The Multilayer Perceptron (MLP) training has some classical problems in many applications. Long training sessions are often required in order to find an acceptable weight solution because of the well known difficulties inherent in gradient descent optimization [1][2] [3]. For this reason, many new approaches on how to improve the convergence have been proposed. In [4], Orozco-Monteagudo et al. applied Cellular Genetic Algorithms (CGA) to train NNs. This method was implemented as software, CGANN-Trainer, which was used to generate binary classifiers for recognition of patterns associated with breast cancer images in a multi-objective optimization problem, improving the accuracy in the patterns classification. Guijarro-Berdias *et al.* [5] proposed a learning algorithm that applies linear-least-squares. Where the estimation of the desired output and the calculation of the weights are simultaneously obtained, layer by layer, in a backward way. This algorithm improved performance of classical Backpropagation and getting to low computational cost. Li *et al.* [6] proposed a Backpropagation

---

[*] This work has been supported by National Spanish Ministry (MICINN) under the project: PTFNN (MCINN ref: AGL2006-12689/AGR). The author wishes to thank to The National Founda tion of Science Technology and Innovation (FONACIT) of the Bolivariana Republic of Venezuela for its contribution in the development of his PhD doctoral studies.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 174–179, 2009.
© Springer-Verlag Berlin Heidelberg 2009

algorithm that improved based conjugate gradient. In the conjugate gradient algorithms, a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions.

The Metaplasticity as a biological concept is widely known in Biology, Medical Computer Science, Neuroscience, Physiology, Neurology and others [7][8][9][10]. We simulated the Metaplasticity property in a Multilayer Perceptron (MMLP) proposed in [13]. In this paper, we test the MMLP Neural Networks on the cancer images classification.

For a correct understanding of metaplasticity mechanisms, we will start with an introduction to the synaptic plasticity. In Section 2 we present an introduction to the regulatory mechanisms of neuronal plasticity, to allow the understanding of the biological metaplasticity mechanisms. In Section 3 we introduce the neural network computational model that makes use of the neuronal plasticity properties explained in Section 2. The experimental setup of the simulation in MATLAB © is also explained. Section 4 presents the results obtained from the experiments. Finally, Section 5 compares the experimental results with the latest research results, concluding that the results of the simulation are consistent proving our method to be an interesting alternative for pattern classification.

2 Metaplasticity Mechanisms

The Metaplasticity is defined by many scientists as the plasticity property of synaptic plasticity [14] [15]. The prefix "meta", Greek for "beyond" or "above", is used to indicate a higher level of plasticity, expressed as a change or transformation in the way synaptic efficacy is modified. An understanding of metaplasticity might yield new insights into how the modification of synapses is regulated and how information is stored by synapses in the brain. For a correct understanding of this mechanisms we will start with an introduction to synaptic plasticity.

2.1 Synaptic Plasticity

The Synaptic plasticity refers to the modulation of the efficacy of information transmission between neurons, being related to the regulation of the number of ionic channels in synapses. The first model of synaptic plasticity was postulated by Hebb and it is known as the Hebb rule [16][17].

2.2 Synaptic Metaplasticity

One of the important biological characteristics of the synaptic rule is that it also models metaplasticity, which is an important homeostatic mechanism of neurons (because it regulates weight variation, down-regulating weight increment in synapses with initially-high weights and up-regulating weight increment in synapses with initially-low weights)[18]. It slows down the process of weight change, making more difficult for the neuron to become either inactive or saturated.

2.3 Intrinsic Plasticity

Although synaptic metaplasticity makes it difficult for synaptic weights to become either null or saturated, it does not totally preclude either of these two extreme situations. The homeostatic property intrinsic plasticity was used in order to totally preclude the possibility of either weight annihilation or saturation [10].

3 MMLPs Neural Network

The artificial metaplasticity is modeled in this paper as the ability to change the efficiency of artificial plasticity giving more relevance to the less frequent patterns and resting relevance to the frequent ones [13]. We modeled and trained the metaplasticity using MLP with the following weight function

$$f_X^*(x) = \frac{A}{\sqrt{(2\pi)^N}.e^{B\sum_{i=1}^{N} x_i^2}} \tag{1}$$

where N is the number of hidden layers of neural network and A, B are parameters that will be estimated empirically. Note that we have assumed that a *posteriori* probabilities follow a Gaussian distribution. If this diverges from reality, the training can even not converge [13].

4 Experiments and Results

One real-world dataset was used for empirical comparison of Classical Backpropagation Train MLP and our proposed method. Wisconsin Breast Cancer Database from UCI Machine Learning Repository [19] is used in the experiments.

4.1 Wisconsin Breast Cancer Database

This breast cancer database was obtained from the University of Wisconsin Hospital. The database contains 699 examples, where 16 samples have missing values which are discarded in a pre-processing step, so only 683 were used. Each sample has one of 2 possible classes: benign or malignant. The dataset contains 444 benign samples (65%) and 239 malignant samples (35%). Each record in the database has nine attributes.

4.2 Characteristics of the Experiments

The MMLP proposed as the classifier was implemented using MATLAB © software (MATLAB version 7.4, R2007a). The nine attributes detailed in the previous section (4.1) were used as the inputs of the ANNs.

The ANNs applied are MMLP with the structure of 9 neural inputs, 8 hidden layers and 1 output, determined empirically (many researchers have proposed methodologies to determine the number of ANN layers, but until now none was 100% effective)[11][12]. The activation function is sigmoidal with scalar output in the range (0,1) and it is the same for all the neurons. All the experiments were performed on the same training data set and tested with the evaluation data set. The training set consisted of 450 samples, 225 malignant records and 225 benign records. The testing set consisted of 233 samples with 117 malignant records and 116 benign records.

For the experiments, we generated 100 MMLP with different weights whose values were random with normal distribution (mean 0 and variance 1). In each experiment 100 networks were trained in order to achieve an average result that does not depend on the initial random value of the weights of the ANN. Two different criterions were applied to stop the training: in one case it was stopped when the error reached 0.01 and in other after performing 2000 epochs.

Two different types of experiments were performed. One to determine the degree of accuracy the MMLP algorithm trying with several structures of network, varying with metaplasticity parameters A and B, until the most efficient structure was obtained, with the criteria being the smaller number of patterns for the training and shortest the time of convergence of algorithm. And other experiment was to compare our algorithm with a classical Backpropagation training.

4.3 The Best Obtained Network

After the tests, the best performing network was obtained. Table 1, shows the structure of the network, metaplasticity parameters, accuracy and others results. The Mean Square Error (MSE) of the best network was obtained with only 355 iterations, that is, less than two epochs were needed to reach the 0.01 classification error and used only 265 patterns. We can see that the network has a rapid error evolution to 0.01 value, with a low number of iterations. This allows us to save time and resources. The results are confirming [13] that the proposed algorithm and the strategy of training turn out to be very interesting to apply in the cases where the number of patterns available to perform training of ANN is low.

Table 1. MMLP results obtained for different network structures and parameters of metaplasticity algorithm

Network Structure			Metaplasticity Parameters		Numbers of Epochs	Numbers Training Patterns	Time of Training	Clustering Accuracy (%)	
I	HL	O	A	B				Training	Testing
9	8	1	39	0.5	2000	265	77.5132 seg	99.99 %	99.14 %
9	8	1	41	0.25	2000	282	88.7929 seg	98.89 %	98.71 %
9	7	1	39	0.25	2000	336	75.2212 seg	99.11 %	98.71 %

5 Comparison

Our MMLP algorithm has been compared with algorithms proposed recently by other investigators for example: Übeyli in [20] obtained 99.54 % of accuracy using different Neural Networks. Karabatak and Cevdet in [21] reached 97.4% of accuracy applying an Association Rule and Neural Network (AR+NN). Guijarro-Berdias *et al.* in [5] achieve 98.6 % of accuracy applying a linear learning method for MLPs using least-squares. In order to determine the performance of MMLPs and BPNNs used in this research for detection of the breast cancer, the classification accuracies on testing sets are presented in Table 2.

Table 2. Clustering accuracy obtained for experiment 100 networks Training and Testing

Classification Accuracy (%)				
Type Network	Network Structure	Training	Testing	Standard Desviation
MMLPNs	9-8-1	99.99 %	99.14 %	0.0037
BPNNs	9-8-1	97.25 %	98.28 %	0.1947

6 Conclusions

The analyzed results allow us to confirm that the proposed MMLP method performs better than BPNNs and its variants. Our method was validated using the Wisconsin breast cancer database in order to evaluate the proposed method performance. The correct classification rate of proposed method for an average of 100 networks was 99.99 % in training and 99.14 % in testing, superior to recent results on the same database. The test stage also shows that this method is an interesting combination of speed, reliability and simplicity and proving to be an interesting alternative in cases where the number of patterns available to perform the training of ANN is low, and as well has a low computational cost.

References

1. Basheer, I.A., Hajmeer, M.: Artificial neural networks: fundamentals, computing, design, and application. Journal of Micro biological Methods 43(1), 3–31 (2000)
2. Chaudhuri, B.B., Bhattacharya, U.: Eficient training and improved performance of multilayer perceptron in pattern classification. Neurocomputing 34(1), 11–27 (2000)
3. Haykin, S., McWhirter, J., Principe, J.C., Sejnowski, T.: New directions in statistical signal processing: From systems to brain. Macmillan, New York (2006)
4. Orozco-Monteagudo, M., Taboada-Crispí, A., Del Toro-Almenares, A.: Training of Multilayer Perceptron Neural Networks by Using Cellular Genetic Algorithms. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 389–398. Springer, Heidelberg (2006)

5. Guijarro-Berdiñas, B., Fontenla-Romero, O., Pérez-Sánchez, B., Fraguela, P.: A linear learning method for multilayer perceptrons using least-squares. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 365–374. Springer, Heidelberg (2007)
6. Li, J.I., XiaoDong, W., XuShul, Y., ShuShen, L., LianSheng, W.: Backpropagation network improved by conjugate gradient based on genetic algorithm in qsar study on endocrine disrupting chemicals. Chinese Science Bulletin 53(1), 33–39 (2008)
7. Jedlicka, P.: Synaptic plasticity, metaplasticidad and bcm theory. Institute of Pathophysiology, Medical Faculty. Comenius University, Bratislava, Slovakia, vol. 103(4-5), pp. 137–143 (2002)
8. Kinto, E., Del-Moral-Hernandez, E., Marcano, A., Ropero-Pelaez, J.: A preliminary neural model for movement direction recognition based on biologically plausible plasticity rules. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4528, pp. 628–636. Springer, Heidelberg (2007)
9. Ropero-Pelaez, J., Piqueira, J.R., Ropero-Pelaez, J., Piqueira, J.R.: Biological clues for up-to-date artificial neurons. In: Andina, D., Pham, D.T. (eds.) Computational Intelligence for Engineering and Manufacturing, Springer, Heidelberg (2007)
10. Monteiro, J.L., Lobo-Netto, M., Andina, D., Ropero-Pelaez, J.: Using neural networks to simulate the alzheimer's disease. In: World Automation Congress 2008, WAC 2008, pp. 1–6 (2008)
11. Cofiño, A.S., Gutiérrez, J.M., Ivanissevich, M.: Evolving modular networks with genetic algorithms: application to nonlinear time series. Expert Systems 21(4), 208–216 (2004)
12. Maniezzo, V.: Genetic Evolution of the Topology and Weight Distribution of Neural Networks. IEEE Transactions on Neural Networks 5(1), 39–53 (1994)
13. Andina, D., Antonio, A.-V., Jevtić, A., Fombellida, J.: Artificial metaplasticity can improve artificial neural network learning. In: Andina, D. (ed.) Intelligent Automation and Soft Computing, Special Issue in Signal Processing and Soft Computing, vol. 15(4), pp. 681–694. TSI Press, USA (2009)
14. Abraham, W.C., Bear, M.F.: Metaplasticity: the plasticity of synaptic plasticity. Trends in Neuroscience 19(4), 126–130 (1996)
15. Abraham, W.C., Tate, W.P.: Metaplasticity: a new vista across the field of synaptic plasticity. Progress in Neurobiology 52, 303–323 (1997)
16. Hebb, D.: The Organisation of Behaviour. Wiley, New York (1949)
17. Peréz-Otaño, I., Ehlers, M.D.: Homeostatic plasticity and NMDA receptor trafficking. Trends in Neuroscience 28, 229–238 (2005)
18. Turrigiano, G.G., Nelson, S.B.: Homeostatic plasticity in the developing nervous system. Nature Reviews Neuroscience 5, 97–107 (2004)
19. `http://archive.ics.uci.edu/ml/datasets.html`
20. Übeyli, E.D.: Implementing automated diagnostic systems for breast cancer detection. Expert Systems with Applications 33(4), 1054–1062 (2007)
21. Karabatak, M., Cevdet-Ince, M.: An expert system for detection of breast cancer based on association rules and neural network. Expert Systems with Applications 36, 3465–3469 (2009)

Probabilistic Self-Organizing Graphs

Ezequiel López-Rubio, Juan Miguel Ortiz-de-Lazcano-Lobato,
and María Carmen Vargas-González

Department of Computer Languages and Computer Science
University of Málaga
Bulevar Louis Pasteur, 35, 29071 Málaga
Spain
{ezeqlr,jmortiz}@lcc.uma.es

Abstract. Self-organizing neural networks are usually focused on prototype learning, while the topology is held fixed during the learning process. Here we propose a method to adapt the topology of the network so that it reflects the internal structure of the input distribution. This leads to a self-organizing graph, where each unit is a mixture component of a Mixture of Gaussians (MoG). The corresponding update equations are derived from the stochastic approximation framework. Experimental results are presented to show the self-organization ability of our proposal and its performance when used with multivariate datasets.

Keywords: self-organization, probabilistic mixtures, unsupervised learning, multivariate data analysis.

1 Introduction

Self-organization is a pervasive concept which seems to explain several neural structures of the brain that perform invariant feature detection [1]. The study of these structures gave rise to the proposal of computational maps designed to explore multidimensional data. The original self-organizing map (SOM) was proposed by Kohonen [2], where each neuron had a weight vector to represent a point of the input space, and the map topology was kept fixed throughout the learning process. The Self-Organizing Dynamic Graph [3] overcame this limitation by learning a topology along with the weight vectors.

Other self-organizing approaches search for topologically correct maps by modifying the update rule of the original SOM in order to achieve shape recovery, notably the Visual Induced Self-Organizing Map (ViSOM, [4]), and other recent approaches inspired in active contours such as the Generalized Topology Preserving SOM [5]. Nevertheless, in these methods there is a fixed topology for the neurons, and the topology preservation is only related to the units' weight vectors.

Here we propose a self-organizing model which is based on a probabilistic mixture of multivariate Gaussian components, where each mixture component is associated with a 'unit' or 'neuron'. This strategy has been explored in [6] and [7]. Furthermore, we use the stochastic approximation framework to learn both the mixture and the

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 180–187, 2009.
© Springer-Verlag Berlin Heidelberg 2009

network topology. Hence, a self-organizing graph is built, whose topology represents the underlying structure of the input data.

We define the model in Section 2. The stochastic approximation learning method for the mixture of multivariate Gaussian distributions and the network topology is considered in Section 3. We show some experimental results in section 4, where our method produces faithful representations of input distributions, and has more classification performance than two well known probabilistic self-organizing models ([6] and [7]). Finally, Section 5 is devoted to conclusions.

2 The Model

Each unit of a *Probabilistic Self-Organizing Graph* (PSOG) model stores a mixture component of a mixture of multivariate Gaussians probability distributions (MoG). The observed (input) space dimension is D, and the likelihood of the observed data $\mathbf{t}$ is modelled by the mixture probability density function:

$$p(\mathbf{t}) = \sum_{i=1}^{H} \pi_i p(\mathbf{t} \mid i) \tag{1}$$

where H is the number of mixture components, and π_i are the prior probabilities or mixing proportions. The multivariate Gaussian probability density associated with each mixture component i is given by

$$p(\mathbf{t} \mid i) = (2\pi)^{-D/2} \left|\mathbf{C}_i\right|^{-1/2} \exp\left(-\frac{1}{2} (\mathbf{t} - \boldsymbol{\mu}_i) \mathbf{C}_i^{-1} (\mathbf{t} - \boldsymbol{\mu}_i)^T \right) \tag{2}$$

where $\boldsymbol{\mu}_i$ and $\mathbf{C}_i$ are the mean vector and covariance matrix of mixture component i, respectively.

Along with this probability distribution, there is an *adjacency matrix* $\mathbf{Z}$ of dimension H×H, whose components z_{ij} are Heskes' confusion probabilities [8]. Here z_{ij} is the probability that mixture component j has the maximum posterior probability $P(j \mid \mathbf{t})$ of having generated an input sample $\mathbf{t}$, while in fact $\mathbf{t}$ was generated by mixture component i. Adjacency matrix $\mathbf{Z}$ represents a topology among the mixture components which is adapted to the input data through the learning process, as we see in the next section.

3 Learning by Stochastic Approximation

When a new sample $\mathbf{t}_n$ has been presented to the network at time step n, its information should be used to update the mixture components. If we want to update mixture component i with the information from sample $\mathbf{t}_n$, an online learning method for a MoG is required. This strategy has been examined by Sato & Ishii [9] for general (non self-organizing) Gaussian PCA mixtures. A Bayesian model selection is carried out to decide which mixture component has the most probability $p(i \mid \mathbf{t}_n)$ of having generated the sample. This component is called the *winning component* j, and is computed as follows:

$$j = Winner(n) = \arg\max_i \{P(i \mid \mathbf{t}_n)\} = \arg\max_i \{\pi_i p(\mathbf{t}_n \mid i)\} \tag{3}$$

Our online method generates the updated values $\pi_i(n)$, $\boldsymbol{\mu}_i(n)$, $\mathbf{C}_i(n)$ and $\mathbf{Z}(n)$ from the old values $\pi_i(n-1)$, $\boldsymbol{\mu}_i(n-1)$, $\mathbf{C}_i(n-1)$, $\mathbf{Z}(n-1)$ and the new sample $\mathbf{t}_n$. The application of Robbins-Monro stochastic approximation algorithm (see [10]) yields the following update equations for each mixture component i:

$$\pi_i(n) = (1-\varepsilon(n))\pi_i(n-1) + \varepsilon(n)R_{ni} \tag{4}$$

$$\mathbf{m}_i(n) = (1-\varepsilon(n))\mathbf{m}_i(n-1) + \varepsilon(n)R_{ni}\mathbf{t}_n \tag{5}$$

$$\boldsymbol{\mu}_i(n) = \frac{\mathbf{m}_i(n)}{\pi_i(n)} \tag{6}$$

$$\mathbf{M}_i(n) = (1-\varepsilon(n))\mathbf{M}_i(n-1) + \varepsilon(n)R_{ni}(\mathbf{t}_n - \boldsymbol{\mu}_i(n))(\mathbf{t}_n - \boldsymbol{\mu}_i(n))^T \tag{7}$$

$$\mathbf{C}_i(n) = \frac{\mathbf{M}_i(n)}{\pi_i(n)} \tag{8}$$

$$z_{ij}(n) = (1-\varepsilon(n))z_{ij}(n-1) + \varepsilon(n)R_{ni}R_{nj} \tag{9}$$

where $\mathbf{m}_i(n)$ and $\mathbf{M}_i(n)$ are auxiliary variables required to update the model parameters, and R_{ni} is the posterior probability of of mixture component i for generating the observed sample $\mathbf{t}_n$:

$$R_{ni} = p(i \mid \mathbf{t}_n) = \frac{\pi_i(n-1)p(\mathbf{t}_n \mid i)}{\sum_{h=1}^{H} \pi_h(n-1)p(\mathbf{t}_n \mid h)} \tag{10}$$

The rows of $\mathbf{Z}$ are subsequently normalized in the 1-norm so that z_{ij} are proper probabilities. Finally, $\varepsilon(n)$ is the step size of the Robbins-Monro algorithm and is typically chosen as

$$\varepsilon(n) = \frac{1}{an+b}, \ 0 < a < 1 \tag{11}$$

so that it falls smoothly and the method converges. In our experiments we have found that selecting $b = 1/a$ often yields good results because the network remains plastic long enough, that is, the parameters to not converge too early. So, we have used

$$\varepsilon(n) = \frac{1}{\varepsilon_0 n + \frac{1}{\varepsilon_0}}, \quad 0 < \varepsilon_0 < 1 \tag{12}$$

where ε_0 is near to zero because higher values produce an excessive plasticity (variability) of the estimations.

4 Experimental Results

4.1 Topology Learning Experiments

This first set of experiments is designed to show graphically how the PSOG model is able to adapt its topology to the underlying structure of the input distribution. We have selected four distributions with D=2. These are uniform distributions on regions with different shapes. We have drawn 20,000 training samples from each distribution. The PSOG models had 16 mixture components, and they were trained during 100,000 time steps.

The input distributions and the corresponding PSOG models are shown in Figures 1 to 4. It must be noted that we have plotted the three strongest adjacencies for each unit with straight lines. As in all cases the diagonal of $\mathbf{Z}$ has the largest elements, the adjacencies z_{ii} are represented as small dots in the position of $\boldsymbol{\mu}_i$. Furthermore, the unit

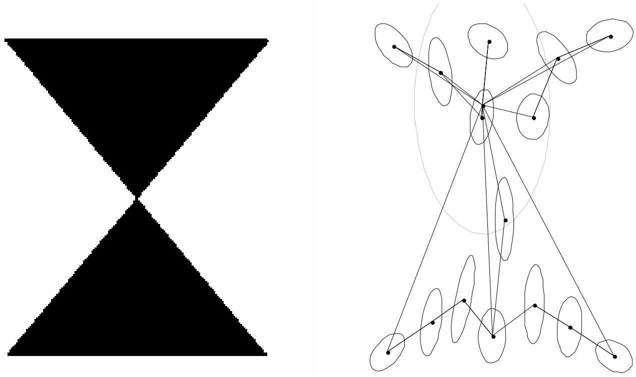


Fig. 1. Bidimensional 'sand clock' input distribution (left) and corresponding PSOG network (right)

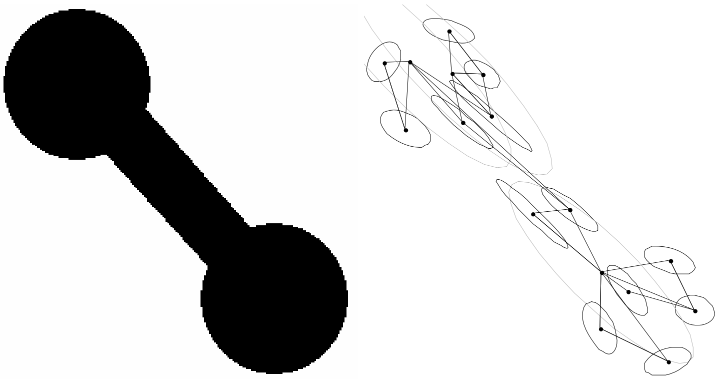


Fig. 2. Bidimensional 'dumbbell' input distribution (left) and corresponding PSOG network (right)

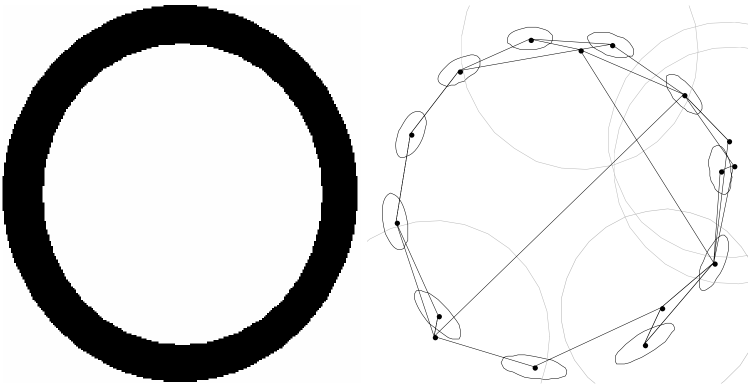


Fig. 3. Bidimensional 'ring' input distribution (left) and corresponding PSOG network (right)

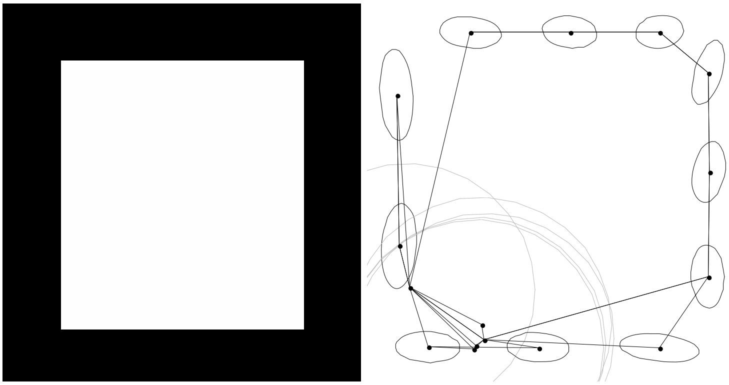


Fig. 4. Bidimensional 'hollow square' input distribution (left) and corresponding PSOG network (right)

Mahalanobis distance ellipses are shown with a grey level which is proportional to the mixing proportions π_i (the mixture components with $\pi_i \approx 0$ have nearly white ellipses).

The figures illustrate how the PSOG model is able to develop connectivity among its units so that it is coherent with the input distribution at hand.

4.2 Classification Experiments

Classification is a typical application of self-organizing map models. Some recent developments are [11], [12], [13], [14] and [15]. We have chosen six standard classification benchmarks from the UCI Machine Learning Repository [16]. We have trained a self-organizing graph for each class of every database, and then the test samples $\mathbf{t}_n$ are assigned to the class corresponding to the map which gives the highest probability $p(\mathbf{t}_n)$.

Moschou et al. [17] have proposed a methodology to assess SOM-type algorithms in classification. Here we follow their method, which considers six quality measures. Among these, the first one is the classification accuracy, i.e. the probability of successful classification.

We have tried PSOG models with 4, 9, 16, 25 and 36 mixture components. They were trained during 100,000 time steps. The databases have been also presented to the Self-Organizing Mixture Model (SOMM) by Verbeek et al. [7] and the joint entropy maximization kernel based topographic maps (KBTM) by Van Hulle [6]. The optimized version of SOMM has been used for the tests. Both KBTM and optimized SOMM compute mean vectors and isotropic covariance matrices at each unit. We have simulated the KBTM for 2,000,000 steps, with parameters η_w=0.01, η_σ=$10^{-4}\eta_w$. We have tested SOMM and KBTM maps with 2×2, 3×3, 4×4, 5×5 and 6×6 square topologies.

We have run 10-fold cross-validations, with disjoint training (90% of the data) and test (10%) sets for each model setup. The results of the 10-fold cross-validation for the best performing number of units of each proposal are shown in Tables 1-6, with the standard deviations in parentheses. Each table corresponds to a quality measure considered in [17]. The tables show that PSOG is able to perform classification tasks with good results. This is consistent with the fact that the other models do not learn the topology of the input distribution, which means that they impose their own topology to the input data. Hence, their adaptation capability is lower.

Table 1. Classification accuracy (higher is better). Standard deviations in parentheses.

	PSOG	SOMM	KBTM
BalanceScale	**0.9339 (0.0513)**	0.7290 (0.0549)	0.7587 (0.0334)
Cloud	**1.0000 (0.0000)**	0.9996 (0.0013)	0.9949 (0.0041)
Dermatology	**0.8861 (0.0898)**	0.8510 (0.0826)	0.3965 (0.1035)
Liver	**0.6685 (0.0806)**	0.5722 (0.0553)	0.5507 (0.0676)
Vowel	**0.8978 (0.0335)**	0.8930 (0.0214)	0.6059 (0.0498)
Wine	**0.9618 (0.0347)**	0.7366 (0.1141)	0.7481 (0.1486)

Table 2. Rand Index (higher is better). Standard deviations in parentheses.

	PSOG	SOMM	KBTM
BalanceScale	**0.9170 (0.0593)**	0.7549 (0.0427)	0.7626 (0.0335)
Cloud	**1.0000 (0.0000)**	0.9991 (0.0027)	0.9899 (0.0082)
Dermatology	0.9173 (0.0815)	**0.9238 (0.0413)**	0.6867 (0.0349)
Liver	**0.5538 (0.0510)**	0.5018 (0.0220)	0.4977 (0.0122)
Vowel	0.9641 (0.0107)	**0.9641 (0.0065)**	0.8976 (0.0121)
Wine	**0.9433 (0.0509)**	0.7550 (0.0745)	0.7696 (0.1264)

Table 3. Hubert's gamma (higher is better). Standard deviations in parentheses.

	PSOG	SOMM	KBTM
BalanceScale	**0.8324 (0.1191)**	0.5004 (0.0930)	0.5058 (0.0721)
Cloud	**1.0000 (0.0000)**	0.9983 (0.0054)	0.9798 (0.0165)
Dermatology	0.7808 (0.1906)	**0.7509 (0.1456)**	0.1249 (0.1062)
Liver	**0.1069 (0.1012)**	0.0050 (0.0457)	–0.0054 (0.0277)
Vowel	**0.7837 (0.0665)**	0.7799 (0.0390)	0.4018 (0.0571)
Wine	**0.8735 (0.1137)**	0.4618 (0.1546)	0.5040 (0.2635)

Table 4. Overall cluster entropy (lower is better). Standard deviations in parentheses.

	PSOG	SOMM	KBTM
BalanceScale	**0.1748 (0.1129)**	0.3746 (0.0863)	0.3882 (0.0534)
Cloud	**0.0000 (0.0000)**	0.0025 (0.0078)	0.0262 (0.0204)
Dermatology	**0.3002 (0.2289)**	0.3009 (0.1444)	1.1353 (0.1811)
Liver	**0.6069 (0.0596)**	0.6446 (0.0485)	0.6331 (0.0487)
Vowel	0.3097 (0.0988)	**0.3013 (0.0462)**	0.8818 (0.0898)
Wine	**0.1127 (0.1028)**	0.4461 (0.1375)	0.4223 (0.1949)

Table 5. Overall class entropy (lower is better). Standard deviations in parentheses.

	PSOG	SOMM	KBTM
BalanceScale	**0.2235 (0.1609)**	0.5761 (0.0626)	0.5375 (0.0422)
Cloud	**0.0000 (0.0000)**	0.0025 (0.0077)	0.0262 (0.0202)
Dermatology	**0.1752 (0.1294)**	0.3151 (0.1624)	0.9835 (0.1515)
Liver	0.5899 (0.0683)	**0.5893 (0.0638)**	0.6005 (0.0492)
Vowel	0.3012 (0.0960)	**0.2986 (0.0482)**	0.8457 (0.0796)
Wine	**0.1058 (0.0964)**	0.4217 (0.1310)	0.3908 (0.1677)

Table 6. Overall entropy (lower is better). Standard deviations in parentheses.

	PSOG	SOMM	KBTM
BalanceScale	**0.1992 (0.1353)**	0.4753 (0.0679)	0.4628 (0.0449)
Cloud	**0.0000 (0.0000)**	0.0025 (0.0078)	0.0262 (0.0203)
Dermatology	**0.2377 (0.1771)**	0.3080 (0.1520)	1.0594 (0.1355)
Liver	**0.5984 (0.0571)**	0.6169 (0.0428)	0.6168 (0.0261)
Vowel	0.3055 (0.0972)	**0.2999 (0.0468)**	0.8638 (0.0837)
Wine	**0.1093 (0.0986)**	0.4339 (0.1303)	0.4065 (0.1764)

5 Conclusions

We have presented a new probabilistic self-organizing model which is able to learn the topology of the input data, so that a self-organizing graph is built. Hence, the underlying structure of the input distribution is revealed by the adjacency matrix of the graph, whose components have a probabilistic interpretation. We have shown the topology adaptation capabilities of the model, and its performance in standard classification benchmarks when compared with two well known probabilistic self-organizing map models. This reveals that our model is well suited for real multidimensional data, where the input structure can not be expected to have a simple topology, unlike traditional self-organizing network models assume.

Acknowledgements

This work is partially supported by Junta de Andalucía (Spain) under contract TIC-01615, project name Intelligent Remote Sensing Systems.

References

1. Fukushima, K.: Self-organization of shift-invariant receptive fields. Neural Networks 12(6), 791–802 (1999)
2. Kohonen, T.: The Self-Organizing Map. Proc. IEEE 78, 1464–1480 (1990)
3. López-Rubio, E., Muñoz-Pérez, J., Gómez-Ruiz, J.A.: Self-Organizing Dynamic Graphs. Neural Processing Letters 16, 93–109 (2002)
4. Yin, H.: ViSOM—a novel method for multivariate data projection and structure visualization. IEEE Trans. Neural Networks 13(1), 237–243 (2002)
5. Huang, D., Yi, Z.: Shape recovery by a generalized topology preserving SOM. Neurocomputing 72(1-3), 573–580 (2008)
6. Van Hulle, M.M.: Joint Entropy Maximization in Kernel-Based Topographic Maps. Neural Computation 14(8), 1887–1906 (2002)
7. Verbeek, J.J., Vlassis, N., Krose, B.J.A.: Self-organizing mixture models. Neurocomputing 63, 99–123 (2005)
8. Heskes, T.: Self-organizing maps, vector quantization, and mixture modeling. IEEE Transactions on Neural Networks 12(6), 1299–1305 (2001)
9. Sato, M., Ishii, S.: On-line EM Algorithm for the Normalized Gaussian Network. Neural Computation 12(2), 407–432 (2000)
10. Kushner, H.J., Yin, G.G.: Stochastic approximation and Recursive Algorithms and Applications, 2nd edn. Springer, New York (2003)
11. Lakany, H.: Extracting a diagnostic gait signature. Pattern Recognition 41(5), 1644–1654 (2008)
12. Wang, B., Fujinaka, T., Omatu, S., Abe, T.: Automatic inspection of transmission devices using acoustic data. IEEE Transactions on Automation Science and Engineering 5(2), 361–367 (2008)
13. Wong, H.-S., Ma, B., Sha, Y., Ip, H.H.S.: 3D head model retrieval in kernel feature space using HSOM. Pattern Recognition 41(2), 468–483 (2008)
14. Powers, S.T., He, J.: A hybrid artificial immune system and Self Organising Map for network intrusion detection. Information Sciences 178(15), 3024–3042 (2008)
15. Alvarez-Guerra, M., González-Piñuela, C., Andrés, A., Galán, B., Viguri, J.R.: Assessment of Self-Organizing Map artificial neural networks for the classification of sediment quality. Environment International 34(6), 782–790 (2008)
16. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, Department of Information and Computer Science, Irvine, CA (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
17. Moschou, V., Ververidis, D., Kotropoulos, C.: Assessment of self-organizing map variants for clustering with application to redistribution of emotional speech patterns. Neurocomputing 71(1-3), 147–156 (2007)

Spicules for Unsupervised Learning

J.A. Gómez-Ruiz, J. Muñoz-Perez, and M.A. García-Bernal

Dept. of Languages and Computer Science, University of Málaga, 29071 Málaga, Spain
`{janto,munozp,magb}@lcc.uma.es`

Abstract. We present a new model of unsupervised competitive neural network, based on spicules. This model is capable of detecting topological information of an input space, determining its orientation and, in most case, its skeleton.

Keywords: competitive neural networks, unsupervised learning, shapes description.

1 Introduction

Neural network models based on self-organizing maps, as proposed by Willshaw and von der Malsburg [1], Grossberg [2] and Kohonen [3], generate maps with a determinate dimension coming from an input space with larger dimensionality. These maps obtain a certain topological structure that preserves a neighborhood relation, which have the property of representing the highest density areas of input space. This property is interesting in different disciplines like voice recognition [4], combinatorial optimization [5], pattern recognition [6] and data compression [7].

The fact that similar maps have been found in human and animal brains, denotes that this topological preservation is important, at least in "natural processing systems".

In Kohonen's self-organizing maps, it is important to emphasize that both, the number of neurons and neighborhood topology between them, must be established before the learning process. This fact implies a great limitation, since in many cases there is not previous information that indicate us an adequate election of such parameters and forces to carry out, therefore, different simulations to determine them.

Another solution consists of determining these parameters incrementally during the learning process. This solution has been adopted and developed by diverse authors, proposing different models of neuronal networks included in what is generically named as "ontogenic networks" [8]. Within this classification we found the growing cell structures, developed by Fritzke [9], and the neuronal networks based on competitive hebbian learning, developed by Martinetz [10].

The growing cell structures [9] are able to generate dimensionality reducing mappings which may be used, for example, for visualization of high-dimensional data or for clustering. In contrast to Kohonen's self-organizing feature maps, which serves similar purposes, neither the number of units nor the exact topology has to be predefined in this model. Instead, a growth process successively inserts units and connections. Moreover, there is no need to define a priori the number of adaptation steps, making possible to continue the growth process until a specific network size or until an applications-dependent performance criterion is fulfilled.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 188–195, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The purpose of the competitive hebbian learning [10] is to distribute a number of centers according to some probability distribution (neural gas) and to generate a topology among these centers which has a dimensionality which is equal to the local dimensionality of the data and may be different in parts of the input space.

In this line, we present a model of competitive and unsupervised neural network based on spicules capable of detecting the topological information of an input space.

The remainder of this paper is organized as follows. In Section 2, we describe the segment based neural network, which is needed for the construction of the new model. In Section 3, we present the new spicules-based competitive neural network. In sections 4, 5 and 6 the computation dynamics, the learning rule and the learning process of this network are respectively explained. In section 7 experimental results are shown and, finally, in section 8, the conclusions of this work are presented.

2 The Segments-Based Neural Network

The main contribution of the segments-based neural network is that the synaptic weights are composed of two reference vectors, which determine the extremes of a segment. In other words, every unit of the network has associated a segment that will evolve in the learning process. Since the prototype of every unit is a segment, instead of a single vector, additional information, for example the predominant direction of the classes, can be obtained.

The topology of the network is similar to the topology of the simple competitive neural network. The neural network is composed by N input sensors and K neurons, where N is the dimension of input patterns. All input sensors are connected with all neurons.

The synaptic weight of every neuron in the network is determined by a pair of vectors ($\mathbf{w}_{i1}$, $\mathbf{w}_{i2}$). The notation is the following:

$$\begin{aligned} \mathbf{w}_{i1} &= (w_{i1}^1, w_{i1}^2, w_{i1}^3, \ldots, w_{i1}^N),\ i = 1,2,\ldots K \\ \mathbf{w}_{i2} &= (w_{i2}^1, w_{i2}^2, w_{i2}^3, \ldots, w_{i2}^N),\ i = 1,2,\ldots K \end{aligned} \tag{1}$$

When an input pattern, **x,** is presented to the network, the synaptic potential associated to the i-th neuron is

$$h_i = \alpha_i(\mathbf{x})h_{i1} + \overline{\alpha}_i(\mathbf{x})h_{i2} + \frac{1}{2}\alpha_i(\mathbf{x})\overline{\alpha}_i(\mathbf{x})\left\|\mathbf{w}_{i1} - \mathbf{w}_{i2}\right\|^2,\quad i = 1,2,\ldots,K. \tag{2}$$

where h_{i1} and h_{i2} are real numbers determined by the following expressions:

$$h_{i1} = \mathbf{w}_{i1}^T\mathbf{x} - \frac{1}{2}\mathbf{w}_{i1}^T\mathbf{w}_{i1} \quad \text{and} \quad h_{i2} = \mathbf{w}_{i2}^T\mathbf{x} - \frac{1}{2}\mathbf{w}_{i2}^T\mathbf{w}_{i2},\qquad i = 1, 2, \ldots, K \tag{3}$$

and
$$\alpha_i(\mathbf{x}) = \frac{(\mathbf{w}_{i1} - \mathbf{w}_{i2})^T(\mathbf{x} - \mathbf{w}_{i2})}{\left\|\mathbf{w}_{i1} - \mathbf{w}_{i2}\right\|^2},\quad \alpha_i(\mathbf{x}) + \overline{\alpha}_i(\mathbf{x}) = 1 \tag{4}$$

Now, the segment composed by the pair of reference vectors ($\mathbf{w}_{i1}$,$\mathbf{w}_{i2}$) can be defined as:

$$S_{\mathbf{w}_{i1}\mathbf{w}_{i2}} = \left\{ \mathbf{s} = \alpha_i(\mathbf{x})\mathbf{w}_{i1} + (1-\alpha_i(\mathbf{x}))\mathbf{w}_{i2} \ , \alpha_i(\mathbf{x}) \in [0,1] \right\} \tag{5}$$

This segment is called *structural segment* and can also be denoted by S_i. The distance, *D*, from an input pattern **x** to a structural segment is evaluated by the expression:

$$D(\mathbf{x}, S_i) = \left\| \mathbf{x} - S_{\mathbf{w}_{i1}\mathbf{w}_{i2}} \right\| = \begin{cases} \left\| \mathbf{x} - (\alpha_i(\mathbf{x})\mathbf{w}_{i1} + (1-\alpha_i(\mathbf{x}))\mathbf{w}_{i2}) \right\| & \text{if } \alpha_i(\mathbf{x}) \in (0,1) \\ \left\| \mathbf{x} - \mathbf{w}_{i1} \right\| & \text{if } \alpha_i(\mathbf{x}) \geq 1 \\ \left\| \mathbf{x} - \mathbf{w}_{i2} \right\| & \text{if } \alpha_i(\mathbf{x}) \leq 0 \end{cases} \tag{6}$$

If $\alpha_i(\mathbf{x}) \in (0,1)$, then the point $\mathbf{w}_{i0} = \alpha_i(\mathbf{x})\mathbf{w}_{i1} + (1-\alpha_i(\mathbf{x}))\mathbf{w}_{i2}$ is the point of the segment $S_{\mathbf{w}_{i1}\mathbf{w}_{i2}}$ nearest to the input pattern **x** (see figure 1).

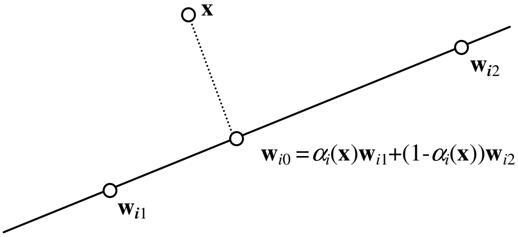


Fig. 1. Distance from an input pattern **x** to the segment $S_{\mathbf{w}_{i1}\mathbf{w}_{i2}}$

When an input pattern **x** is presented to the network only one neuron is activated, the neuron whose structural segment is nearest to the input pattern **x**. This neuron *r* is called the *winning neuron* and satisfies the following expression:

$$\left\| \mathbf{x} - S_r \right\| = \min_{1 \leq i \leq K} \left\| \mathbf{x} - S_i \right\| \tag{7}$$

In this sense, the neuron activated is the neuron with the highest synaptic potential, as is proved by the following theorem:

Theorem 1

$$h_r > h_i \Leftrightarrow \quad \left\| \mathbf{x} - S_r \right\| < \left\| \mathbf{x} - S_i \right\| \ , \forall\, r \neq i,\ i = 1,2,\ldots,K \tag{8}$$

Proof: using the structural segment definition we have:

$$\left\| \mathbf{x} - S_i \right\|^2 = \left\| \mathbf{x} - \left(\alpha_i(\mathbf{x}) w_{i1} + \overline{\alpha}_i(\mathbf{x}) w_{i2} \right) \right\|^2$$

so that:

$$\begin{aligned}
&\left\|x-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\right\|^2=\left(\mathbf{x}-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\right)^T\left(\mathbf{x}-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\right)\\
&=\mathbf{x}^T\mathbf{x}-\mathbf{x}^T\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)^T-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)^T\mathbf{x}\\
&\qquad+\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)^T\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\\
&=\mathbf{x}^T\mathbf{x}-\alpha_i(\mathbf{x})\mathbf{x}^T\mathbf{w}_{i1}-\bar{\alpha}_i(\mathbf{x})\mathbf{x}^T\mathbf{w}_{i2}-\alpha_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{x}-\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}{}^T\mathbf{x}+\alpha_i^2(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}\\
&\qquad+2\alpha_i(\mathbf{x})\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2}+\bar{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}\\
&=\mathbf{x}^T\mathbf{x}-2\alpha_i(\mathbf{x})\mathbf{x}^T\mathbf{w}_{i1}-2\bar{\alpha}_i(\mathbf{x})\mathbf{x}^T\mathbf{w}_{i2}+\alpha_i^2(\mathbf{x})\mathbf{w}_{i1}{}^T\mathbf{w}_{i1}+2\alpha_i(\mathbf{x})\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}{}^T\mathbf{w}_{i2}+\bar{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i1}{}^T\mathbf{w}_{i2}
\end{aligned}$$

From expression (3) we obtain that

$$\mathbf{w}_{i1}{}^T\mathbf{x}=h_{i1}+\frac{1}{2}\mathbf{w}_{i1}{}^T\mathbf{w}_{i1}\ ,\ \mathbf{w}_{i2}{}^T\mathbf{x}=h_{i2}+\frac{1}{2}\mathbf{w}_{i2}{}^T\mathbf{w}_{i2}$$

and now

$$\begin{aligned}
&\left\|x-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\right\|^2=\\
&=\mathbf{x}^T\mathbf{x}-2\alpha_i(\mathbf{x})\left(h_{ii}+\frac{1}{2}\mathbf{w}_{i1}^T\mathbf{w}_{i1}\right)-2\bar{\alpha}_i(\mathbf{x})\left(h_{i2}+\frac{1}{2}\mathbf{w}_{i2}^T\mathbf{w}_{i2}\right)+\alpha_i^2(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}\\
&\qquad+2\alpha_i(\mathbf{x})\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2}+\bar{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}\\
&=\mathbf{x}^T\mathbf{x}-2\alpha_i(\mathbf{x})h_{i1}-\alpha_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}-2\bar{\alpha}_i(\mathbf{x})h_{i2}-\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}+\alpha_i^2(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}\\
&\qquad+2\alpha_i(\mathbf{x})\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2}+\bar{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}\\
&=\mathbf{x}^T\mathbf{x}-2\alpha_i(\mathbf{x})h_{i1}-2\bar{\alpha}_i(\mathbf{x})h_{i2}-\alpha_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}+\alpha_i^2(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}-\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}\\
&\qquad+2\alpha_i(\mathbf{x})\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2}+\bar{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}\\
&=\mathbf{x}^T\mathbf{x}-2\alpha_i(\mathbf{x})h_{i1}-2\bar{\alpha}_i(\mathbf{x})h_{i2}-\alpha_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}\left(1-\alpha_i(\mathbf{x})\right)-\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}\left(1-\bar{\alpha}_i(\mathbf{x})\right)\\
&\qquad+2\alpha_i(\mathbf{x})\bar{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2}
\end{aligned}$$

Using the expression in (2) for the synaptic potential of the structural segment, the following expression is obtained:

$$\left\|\mathbf{x}-\left(\alpha_r(\mathbf{x})w_{r1}+\bar{\alpha}_r(\mathbf{x})w_{r2}\right)\right\|^2=\mathbf{x}^T\mathbf{x}-2h_i$$

Since in hypothesis it is supposed that $h_r > h_i$, then $\mathbf{x}^T\mathbf{x}-2h_r <\ \mathbf{x}^T\mathbf{x}-2h_i$. That is,

$$\left\|\mathbf{x}-\left(\alpha_r(\mathbf{x})w_{r1}+\bar{\alpha}_r(\mathbf{x})w_{r2}\right)\right\|^2<\left\|\mathbf{x}-\left(\alpha_i(\mathbf{x})w_{i1}+\bar{\alpha}_i(\mathbf{x})w_{i2}\right)\right\|^2$$

$$\left\|\mathbf{x}-S_r\right\|<\left\|\mathbf{x}-S_i\right\|\ ,\ \forall i\neq r\ ,\ \ i=1,2,\ldots,K \qquad \Box$$

3 The Spicules-Based Competitive Neural Network

We name *spherule* to a structure constituted by a central point, called *nucleus*, and a set of segments, called *spicules*, that emanate from it. The nucleus has a synaptic weight denoted by $\mathbf{w_0}$ and the synaptic weights of peripheral points are denoted by $\mathbf{w_1}$, $\mathbf{w_2}$,…, $\mathbf{w}_L$, being L the number of spicules that constitute the spherule. An example of spherule with 3 spicules is shown in figure 2.

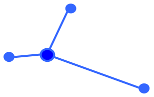


Fig. 2. Spherule with three spicules

The topology of the network is similar to the topology of the simple competitive neural network: N input sensors and L+1 neurons, where N is the dimension of input patterns and L is the number of spicules of the spherule.

Formally, a spherule E with L spicules is defined by the following set:

$$E = \bigcup_{i=1}^{L} \left\{ e_i \in \Re^N : e_i = \alpha \cdot \mathbf{w}_0 + (\mathbf{1} - \alpha) \cdot \mathbf{w}_i , \forall \alpha \in [0,1] \right\} \tag{9}$$

Due to the fact that spicules are segments, the definition of the synaptic potential, h_i, of every spicule, e_i, is identical to the shown one in the expression (2). The distance D from an input pattern **x** to a spicule e_i is evaluated by the expression:

$$D(\mathbf{x}, e_i) = \|\mathbf{x} - e_i\| = \begin{cases} \|\mathbf{x} - (\alpha_i(\mathbf{x})\mathbf{w}_0 + (1 - \alpha_i(\mathbf{x}))\mathbf{w}_i)\| & \text{if} \quad \alpha_i(\mathbf{x}) \in (0,1) \\ \|\mathbf{x} - \mathbf{w}_0\| & \text{if} \quad \alpha_i(\mathbf{x}) \geq 1 \\ \|\mathbf{x} - \mathbf{w}_i\| & \text{if} \quad \alpha_i(\mathbf{x}) \leq 0 \end{cases} \tag{10}$$

where

$$\alpha_i(\mathbf{x}) = \frac{(\mathbf{w}_0 - \mathbf{w}_i)^T (\mathbf{x} - \mathbf{w}_i)}{\|\mathbf{w}_0 - \mathbf{w}_i\|^2} \tag{11}$$

Finally, the distance between an input pattern **x** and a spherule E is given by

$$D(\mathbf{x}, E) = \|\mathbf{x} - E\| = \min_{1 \leq i \leq L} \|\mathbf{x} - e_i\| \tag{12}$$

4 Computation Dynamics

When an input pattern **x** is presented to the network, only one spicule is activated, the spicule e_r that is nearest to the input, satisfying the following expression:

$$\|\mathbf{x} - e_r\| = \min_{1 \leq i \leq L} \|\mathbf{x} - e_i\| \tag{13}$$

In this sense, the spicule activated is the one with the highest synaptic potential, as is proved by the following theorem:

Theorem 2

$$h_r > h_i \quad \Leftrightarrow \quad \|\mathbf{x} - e_r\| < \|\mathbf{x} - e_i\| \ , \quad \forall i \neq r \ , \quad i = 1,2,\ldots,L \tag{14}$$

Proof: due to the fact that spicules are segments, the proof is identical to the proof of theorem 1 in expression (8).

5 The Learning Rule

Let us consider an input space constituted by p patterns (vectors). For the determination of the learning rule, we used the principle of the minimum empirical risk, given by the following distortion function:

$$E_{distorsion} = \sum_{\mu=1}^{p} \sum_{i=1}^{L} M_i^{\mu}(\mathbf{x}) \left\| \mathbf{x}^{\mu} - e_i \right\|^2 , \quad M_i^{\mu}(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \left\| \mathbf{x} - e_i \right\| = \min\limits_{j=1,2,\ldots,L} \left\| \mathbf{x} - e_j \right\| \\ 0 & \text{in another case} \end{cases} \tag{15}$$

where

$$e_i = \left\{ e \in \Re^N : e = \alpha \cdot \mathbf{w}_0 + (\mathbf{1} - \alpha) \cdot \mathbf{w}_i , \forall \alpha \in [0,1] \right\} , \quad i = 1,2,\ldots,L \tag{16}$$

That is, the problem consists of determining the L spicules of the spherule that diminish the distortion error expressed in (15). The learning rule is deduced by the gradient descent method of a distortion function similar to expression (15), obtaining the following:

$$\begin{cases} \text{- if } \; M_i^{\mu}(\mathbf{x}) = 1 & \mathbf{w}_0(k+1) = \eta(k) \cdot \mathbf{x}(k) + (1 - \eta(k)) \cdot \mathbf{w}_0(k) \\ & \mathbf{w}_i(k+1) = \eta(k) \cdot \mathbf{x}(k) + (1 - \eta(k)) \cdot \mathbf{w}_i(k) \\ \text{- if } \; M_i^{\mu}(\mathbf{x}) = 0 & \mathbf{w}_i(k+1) = \mathbf{w}_i(k) \end{cases} \tag{17}$$

where k is the iteration in course, and $\eta(k)$ is the learning parameter. This parameter is a monotonous decreasing function that depends on the number of iterations. Let us consider $\eta(k)$ as

$$\eta(k) = \eta_0 (1 - k / T) , \quad k = 1,2,\ldots,T \tag{18}$$

being T the total number of iterations and $\eta_0 \in [0,1]$ the initial value.

From the learning rule exposed in (17), it can be deduced that when an input pattern **x** is presented to the network, only the end of the winning spicule is modified, but, since the nucleus of the spherule is common to all the spicules, it will be always modified in every step of the learning process.

6 The Learning Process

Similarly to Kohonen's self-organizing maps [3], the learning process is divided in two phases: the *ordering phase* and the *convergence phase*:

- Ordering phase: during this phase the adaptive process takes place in which the nucleus of the spherule is positioned in the gravity centre of input space and spicules are distributed according to the topological structure of this input space. As much the synaptic weights of the nucleus as all the weights of the spicules ends, are modified with the same learning parameter, $\eta(k)$.
- Convergence phase: this phase is needed to fine-tune the spherule structure and, therefore, provide an accurate statistical quantification of the input space. The synaptic weights are modified with two different learning parameters: $\eta_a(k)$ for the synaptic weight of the nucleus and $\eta_b(k)$ for the synaptic weights of the spicules ends.

In the ordering phase, the initial value for the learning parameter, η_0, will be near to one whereas in the convergence phase, the initial values for the learning parameters, η_{a0} and η_{b0}, will be inferior that the previous one, fulfilling in addition that $\eta_{b0} \geq \eta_{a0}$. Moreover, the number of necessary iterations for the convergence phase must be minor that in the ordering phase.

7 Experimental Results

Different shapes in digital images have been considered using, for all of them, the same configuration: in order to obtain the input patterns, 800 points have been randomly chosen within such figures (p = 800); the ordering phase has 4 iterations with η_0 = 0.9 and the convergence phase has 2 iterations with η_{a0} = 0.1 and η_{b0} = 0.3.

In figure 3, the result of the network is shown from three different convex shapes (a spherule with 3 spicules for the triangle and with 4 spicules for the rectangle and rhombus). It can be appreciated that spicules are topologically organized, allowing that the resulting spherule constitutes the skeleton of the different input spaces and, therefore, giving very valuable information that could be used for a possible automatic recognition of shapes. In figure 4, the results of the network from a circle shape with different number of spicules are shown. An interesting result can be appreciated, as spicules are added to the spherule, these are organized radially, so that the circle is partitioned in portions of similar size. In figure 5, the results from a non-convex shape are shown. As spicules are added to the network, these are distributed along the shape, looking for its topological characteristics.

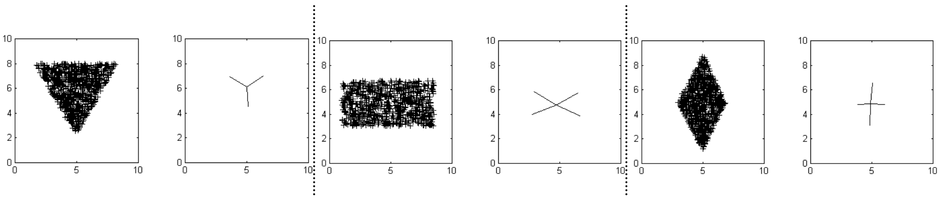


Fig. 3. Results from three different convex shapes

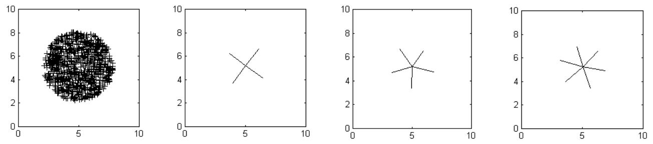


Fig. 4. Results from a circle shape with different number of spicules

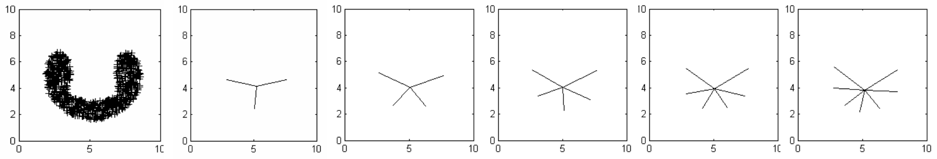


Fig. 5. Results from a non-convex shape with different number of spicules

In all simulations, it can be observed that the nucleus of spherule is positioned in the gravity centre of input spaces, but not necessarily over input patterns like in figure 5, whereas spicules ends are always distributed over them.

8 Conclusions

We present a new model of unsupervised competitive neural network based on spicules in which a new hierarchic structure constituted by association of several segments is developed. This hierarchic structure is called *spherule* and it is constituted by a central point, called *nucleus*, and a set of segments, called *spicules*, that emanate from it. As it is presented in experimental results, this model is capable of detecting topological information of an input space, determining its orientation and, in most case, its skeleton.

At the moment, we are working in providing to the network, on the one hand, with a mechanism that allows to identify the optimal number of spicules and, by another one, of a process of replication or multiplication of spherules in the assumption in which the input space is constituted by more than one group.

Acknowledgements

This work has been partially supported by TIN-7362 project (MEC).

References

1. Willshaw, D.J., von der Malsburg, C.: How patterned neural connections can be set up by self-organization. Proc. of the Royal Society of London B 194, 431–445 (1976)
2. Grossberg, S.: Adaptative pattern classification and universal recording: parallel development and coding of neural detectors. Biol. Cyber. 23, 121–134 (1976)
3. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, 59–69 (1982)
4. Kohonen, T.: The neural phonetic typewriter. IEEE Computer 21, 11–22 (1988)
5. Favata, F., Walker, R.: A study of the application of Kohonen-type neural networks to the traveling salesman problem. Biological Cybernetics 64, 463–468 (1991)
6. López-Rubio, E., Muñoz-Pérez, J., Gómez-Ruiz, J.A.: Invariant pattern identification by self-organizing networks. Pattern Recognition Letters 22, 983–990 (2001)
7. Gómez-Ruiz, J.A., López Rubio, E., Muñoz Pérez, J.: Expansive competitive learning for colour imagen compression. In: Proceedings of the II international symposium on neural computation, Berlin, Germany, pp. 522–527 (2000)
8. Fiesler, E., Beale, R.: Handbook of Neural Computation. Oxford University Press, Oxford (1997)
9. Fritzke, B.: Growing cell structures: a self-organizing network for unsupervised and supervised learning. Neural Networks 7(9), 1441–1460 (1994)
10. Martinetz, T.M.: Competitive hebbian learning rule forms perfectly topology preserving maps. In: Proceedings of the International Conference on Artificial Neural Networks, Amsterdam, Holland, pp. 427–434 (1993)

Topology Preserving Visualization Methods for Growing Self-Organizing Maps

Soledad Delgado[1], Consuelo Gonzalo[2], Estibaliz Martinez[2], and Agueda Arquero[2]

[1] Department of Organization and Structure of Information, Technical University of Madrid,
Ctra. Valencia, Km. 7, 28031 Madrid, Spain
sole@eui.upm.es

[2] Department of Architecture and Technology of Computer Systems, Technical University of Madrid, Campus de Montegancedo, 28660 Madrid, Spain
{chelo,emartinez,aarquero}@fi.upm.es

Abstract. Self-organizing map (SOM) is a neural network model widely used in high dimensional data visualization processes. A trained SOM provides a simplified data model as well as a projection of the multidimensional input data into a bi-dimensional plane that reflects the relationships involving the training patters. Visualization methods based in SOM explore different characteristics related to the data learned by the network. It is necessary to find methods to determine the goodness of a trained network in order to evaluate the quality of the high dimensional data visualizations generated using the SOM simplified model. The degree of topology preservation is the most common concept used to implement this measure. Several qualitative and quantitative methods have been proposed for measuring the degree of SOM topology preservation, in particular using Kohonen model. In this work, two measuring topology preservation methods for Growing Cell Structures (GCS) model are proposed: the topographic function and the topology preserving map.

Keywords: topology preserving, self-organizing map, growing cell structures, visualization methods, Delaunay triangulation.

1 Introduction

At present there are multitude of investigation areas in which is necessary to work with large volumes of information of high dimension nature. In the last past years, great advances have been generated in the development of data mining tools that facilitate this work. Data mining first stage usually consist of building simplified global overviews of dates sets, generally in graphical form [1]. Self-Organizing Map (SOM) fit well in the exploratory data analysis since its main purpose is the visualization and analysis of nonlinear relations between multidimensional data [2]. SOM is a model that allows obtaining simplified models of data sets, normally of high dimension. In addition, it has the characteristic of generating a projection of this high-dimensional input space in a two-dimensional one that reflects the existing relations in the training patterns. These two characteristics have made SOM an ideal model for exploitation in diverse visualization techniques of multidimensional information [3-6]. All of them are based on exploring different characteristics related to the data learned by the network.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 196–203, 2009.
© Springer-Verlag Berlin Heidelberg 2009

SOM generate a mapping of the input space (data manifold M), usually multidimensional, to the two-dimensional plane of the output layer of the network (lattice A of N neural units). These networks use an unsupervised learning in which the weights of the network are adapted associating each training pattern with a neuron of the two-dimensional output map (the one that best matches with the pattern) and moving the synaptic vector of this one closer to that input. The aim of this process is that similar input patterns are mapped by neighbor output units, as well as neighbor output units map only similar input patterns. This is what could be defined informally as preservation of the topology [7], which reflects the quality of the simplified model as well as the degree of success in the choice of the neighborhood connections architecture of the output layer neurons.

Generally, SOM multidimensional information visualization techniques are based on generating graphs of the two-dimensional output layer of the network in which knowledge acquired during the learning process is included. If a trained SOM has a low degree of topology preservation, the graphs generated using the information of the network will not be reliable. Therefore, it is essential to create tools to evaluate the quality of trained networks in order to validate the visualizations based on it.

Several qualitative and quantitative methods have been proposed for measuring the degree of SOM topology preservation, in particular using Kohonen model [2, 8].Many of them are based on an assessment of the position of the neurons in the output lattice and, on to other hand, in evaluating the position of their synaptic vectors, without considering the data manifold in the measurement. When nonlinear data manifold are used, these measures give incorrect results, usually. Several methods that explicitly take into account the structure of data manifold have been proposed [9-10] and has been proved using Kohonen SOM.

In Kohonen SOM the lattice of the output layer has to be pre-specified at the beginning of the training process and lets static during all the adaptation steps. In many cases it is impossible to determine a neighborhood connection architecture a priori, that fits well with the structure of the data manifold, thus providing Kohonen SOM fails providing perfectly topology preserving maps. Growing Cell Structures [11] is a SOM model with a dynamic architecture of the two-dimensional output layer (for a neighbor connection factor of k=2), adding and eliminating neurons during adaptation process of the network. The two-dimensionality of the output layer offers a way to make use of this model in high dimension information visualization techniques [12]. The flexibility of the output layer lattice normally provides GCS networks obtaining degrees of topology preservation better than Kohonen networks.

In this work, two measuring topology preservation methods for Growing Cell Structures (GCS) are proposed: topographic function and topology preserving map.

2 The Topographic Function in GCS

The output layer in GCS network is organized by neurons that exhibit neighborhood connections forming k-dimensional structures. Normally, factor k takes value 2, so these structures have the shape of a triangle. As in the Kohonen model, each output neuron i has associated a synaptic vector, w_i, with the same dimension that the input patterns ($M \subset \mathbb{R}^d$ and $w_i \in \mathbb{R}^d$). In a GCS trained network, the synaptic vectors of the N output layer neurons (A) are prototype vectors of the training patterns dataset.

Each neuron $i \in$ A has a receptive field for which is the best matching unit, *bmu*, known as the Voronoi region (V_i), where the synaptic vector w_i is the kernel point. Each neuron represents those training patterns that are within its Voronoi region.

The neighborhood connections of the GCS output layer define the connectivity matrix $C_{ij}^{(A)}$ of the *A* network:

$$C_{ij}^{(A)} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are connected} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

$C_{ij}^{(A)}$ defines the lattice of *A* and is used to obtain the matrix D_A, that will keep the distance between each pair of neurons *i* and *j*, understood as the minimum number of neighborhood connections that separate *i* and *j* according to $C_{ij}^{(A)}$. As a result of the removal of neurons during the weight adaptation process of the network, the neighborhood lattice of the output layer in GCS network can result divided in two or more isolated meshes. In these cases, when calculating D_A, the distance between two neurons that belong to two different meshes is theoretically ∞.

In addition, using training patterns a new connectivity matrix can be defined based on the characteristics of the data manifold. Each input pattern has associated a *bmu* and a second *bmu*, being this last the neuron of the output layer whose synaptic vector is the second most similar to the pattern (using Euclidean distance). With the concept of *bmu* and second *bmu* a new connectivity matrix $C_{ij}^{(M)}$ can be generated, that will reflect how the output neurons should be connected using the knowledge inherent to *M*:

$$C_{ij}^{(M)} = \begin{cases} 1, & \text{if } i \text{ is bmu and } j \text{ is second bmu for some input pattern} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

This new connectivity matrix reflects what is known as the induced Delaunay triangulation. $C_{ij}^{(M)}$ matrix is used to calculate D_M, which will keep the distance between each pair of neurons *i* and *j* according to the lattice established by *M*.

Using D_M y D_A distance matrices the topographic function Φ_A^M [9] can be defined for GCS network:

$$\Phi_A^M(k) = \begin{cases} \frac{1}{N} \sum_{j \in A} f_j(k) & k > 0 \\ \Phi_A^M(1) + \Phi_A^M(-1) & k = 0 \\ \frac{1}{N} \sum_{j \in A} f_j(k) & k < 0 \end{cases} \tag{3}$$

where:

$$\begin{aligned} f_i(k) &= \#\{ j \mid D_A(i,j) > k \,;\, D_M(i,j) = 1 \} \\ f_i(-k) &= \#\{ j \mid D_A(i,j) = 1 \,;\, D_M(i,j) > k \} \end{aligned} \tag{4}$$

The expression #{.} denotes de cardinality of the set, and *k* is defined in the range [1..N-1]. A GCS network is perfectly topology preserving if $\Phi_A^M(0) = 0$. In the range of positive values of *k*, Φ_A^M function calculates the mean number of neurons that should exhibit a direct neighbor connection in *A*, but presents a distance greater than *k*. On the other hand, in the range of negative values ok *k*, Φ_A^M function determines the mean number of neurons that are direct neighbors in *A*, but should be to a distance greater than *k*. The topographic function can be painted in the (-N..N) range, giving a tool of fast evaluation to measure the topology preserving quality in a GCS network.

3 GCS Topology Preserving Map

When *k*=2 architecture factor is used, the GCS output layer neurons are organized in groups of interconnected triangles, forming a bi-dimensional mesh that can be projected in the plane [11]. The resulting graph is known as topographic map, and can be used to implement different visualization methods [12].

The GCS topographic map and the data manifold can be used to generate a graph that reflects the degree of topology preservation, similar to that one proposed in [10] for Kohonen networks. This map will reflect the neurons of the output layer that should be direct neighbors on *A*, according to the data manifold structure *M*, as well as the strength of these connections. In order to be able to detect violations of the topology preservation, it is necessary in parallel to generate the topographic map of the network to visualize the connections that really exist in the lattice *A*.

First of all, the cumulative adjacent matrix *CADJ* is calculated, where *CADJ(i,j)* will keep the number of training patterns for which *i* is de *bmu* and *j* is the second *bmu*. In a second step, the connectivity strength matrix, *CONN*, can be obtained as *CONN(i,j) = CADJ(i,j)+CADJ(j,i)*, that shows how strongly two neurons must be connected in the basis of the data manifold structure.

In GCS topology preserving map neurons are placed in the coordinates calculated for the topographic map of the network [12]. Between each pair of neurons *i* and *j* with *CONN(i,j)> 0*, a line connecting them in the topology preserving map is included. The strength of connections is reflected by the thickness of the line, and is calculated as:

$$line\ width\,(i,j) = \begin{cases} 1 & \mu - \sigma > CONN(i,j) > 0 \\ 2 & \mu > CONN(i,j) > \mu - \sigma \\ 3 & \mu + \sigma > CONN(i,j) > \mu \\ 4 & CONN(i,j) > \mu + \sigma. \end{cases} \tag{5}$$

where μ represents the mean value and σ the variance of the strengths of all connections in *CONN*. Value 1 is associated with the weak and 4 with the strongest. In addition, 5 levels of gray are used that reflect the local strength of the connections for each neuron, associating dark tones with high values and clear degrees with the low ones. A dark line between two neurons *i* and *j* will indicate that exists a high volume of data for which *i* is *bmu* and *j* the second *bmu* or vice versa.

Lines connecting units in topology preserving map which are not direct neighbors in the lattice defined by *A* are considered topology violations. Usually, local violations tend to arise, that is, fine and clear gray lines connecting units that belong to the same isolated cluster of neurons. Strong violations will be represented by strong end dark lines connecting units of separate clusters according to *A*.

4 Experiments and Discussion

In all the experiments, the modification of the training algorithm for GCS networks described in [13] has been used, in order to achieve a feasible interpretation of the removal of units learning parameters. In topographic function graphs, a normalization of X-axis coordinates has been realized, being range (- N..N) standardized to values in the range (-1..1).

In this section, the results of applying the visualizations previously described to two data sets are exposed. First of them is a data group of simulated bi-dimensional points in the plane, distributed in 11 clusters. The second set is a real multidimensional group of data selected from a scene registered by the ETM+ sensor (Landsat 7). The input signals are defined by the six ETM+ spectral bands with the same spatial resolution: ETM+1 to ETM+5, and ETM+7. The input data set has a total number of 1800 pixels, 1500 carefully chosen from the original scene and 300 randomly selected. The input vectors are associated to six land cover categories.

For both data sets two types of training have been performed: with and without removal of neurons. In GCS network without removal, the topology preserving map can be analyzed jointly with some of the topographic maps that make available the visualization of clusters in the data manifold (for example U-map). With the conclusions of this analysis, another GCS network with removal of units can be trained, in which the ending condition of the training takes place when it has obtained at least a desired number of isolated clusters of neurons. The definition, description and values of the common GCS training parameters used in all experiments can be obtained in [12]. For GCS networks without removal of units, a neuron removal factor μ=0 (relative normalized probability density estimation value) has been used, and μ=0.0006 for networks trained with cluster ending condition.

Fig. 1 shows all the information related to the GCS network trained without removal of units using the bi-dimensional data distributed in 11 clusters. This network has been trained until obtaining 100 neurons in the output layer. The topology preserving map (Fig. 1 e) shows clearly the presence of several clusters (groups of dark and heavy lines). U-map (Fig. 1 c) confirms this fact. The topographic function (Fig. 1 d) reveal that exists very few neurons that must be direct neighbors and that are to a small distance (it quickly falls to 0 for positive values), whereas the asymptote that appears for negative values indicates that there are neurons that are direct neighbors who never would have to be it (those that does not identify to any pattern). The value of the topographic function for k=0 indicates that this network

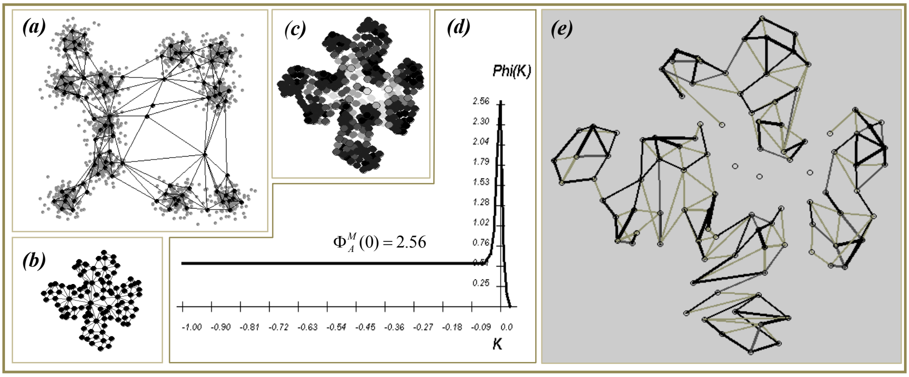


Fig. 1. Eleven separate regions in the bi-dimensional plane. GCS network without removal of units. (a) Visualization of the input data (gray dots) and the network projection (black dots and neighbor connection lines) using the two component values of the synaptic vectors. (b) Topographic map. (c) U-map. (d) Topographic function and topology preserving value for k=0. (e) Topology preserving map.

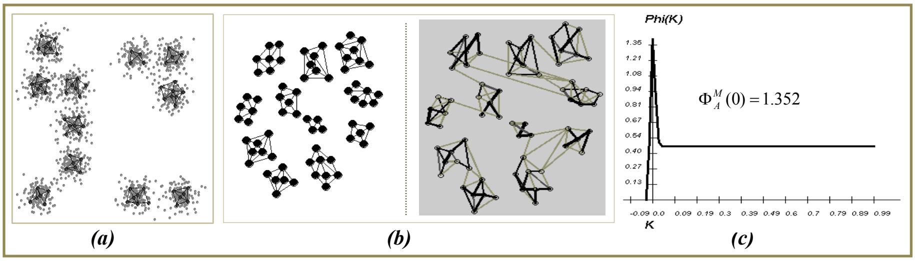


Fig. 2. Eleven separate regions in the bi-dimensional plane. GCS network with removal of units (11 isolated clusters). (a) Visualization of the input data (gray dots) and the network projection (black dots and neighbor connection lines) using the two component values of the synaptic vectors. (b) Topographic map (left) and topology preserving map (right). (c) Topographic function and topology preserving value for k=0.

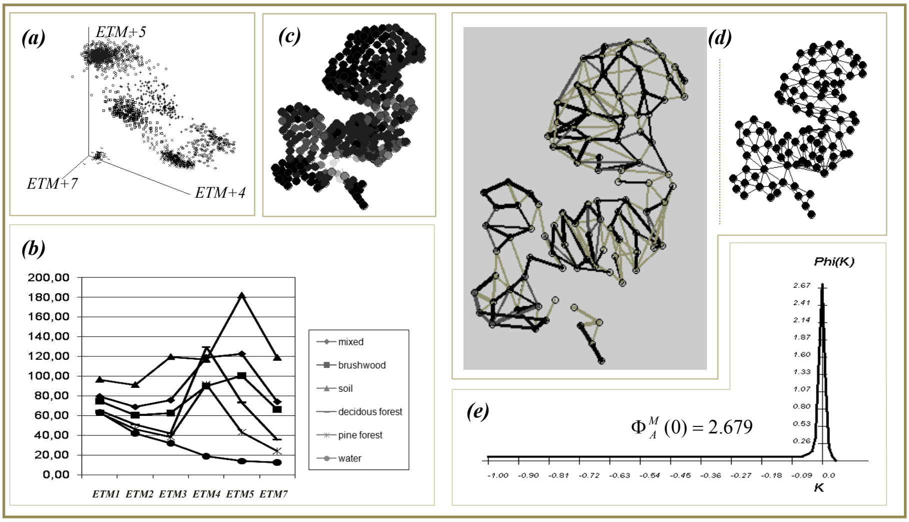


Fig. 3. Multidimensional data of satellite image. GCS network without removal of units. (a) Visualization of the input data using the 4, 5, and 6 component values (ETM+4, ETM+5 and ETM+7) of patterns. (b). Mean values for the six land cover categories present in data. (c) U-map. (d) Topographic map (right) and topology preserving map (left). (e) Topographic function and topology preserving value for k=0.

does not have a good degree of topology preservation. Fig. 2 shows the results obtained with the GCS trained with these data, using removal of neurons. The training ending condition was fulfilled when at least 11 clusters of neurons in the output layer was otained. In the topology preserving map (Fig 2 b) can be observed that there are only a few fine and clear lines (slight topology violations) connecting units that belong to different cluster of neurons. This fact is also reflected in the topographic function for positive values of *k*. In this case, the topographic function (Fig. 2 c) also reveals that there are very few direct neighbors that must be to a small distance

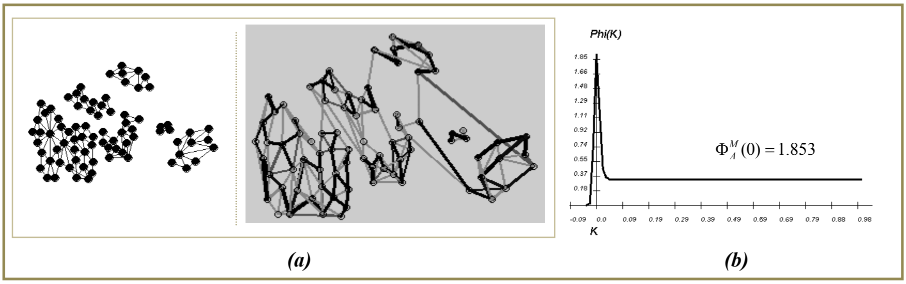


Fig. 4. Multidimensional data of satellite image. GCS network with removal of units (6 isolated clusters). (a) Topographic map (left) and topology preserving map (right). (b) Topographic function and topology preserving value for k=0.

(*k* negative values). Comparing this network with the previous one, the topographic function value for *k=0* shows a significant improvement.

Fig. 3 include some representations associated to de GCS network trained using the second data set, without removal of neurons. 100 output units in the output layer is the ending training condition, again. Topology preserving map (Fig. 3 d) and U-map (Fig. 3 c) show the presence of several clusters in the data manifold. Topographic function (Fig. 3 e) reveal that exists very few neurons that must be direct neighbors and that are to a small distance (it quickly falls to 0 for positive values), whereas the asymptote that appears for negative values indicates that there are very few neurons that are direct neighbors and never would have to be it (those that identify different land cover categories and appear in separate clusters in the bottom of the topology preserving map – Fig. 3 d). In this example, the topographic function value for *k*=0 has not a good quality grade.

Fig. 4 shows some results generated using the GCS network trained with the second data set, removal of units, and looking for at least 6 isolated clusters of neurons in the output layer. In this network, as in the previous data set, the topology preserving map and the topographic function display better quality topology preservation grade with respect the GCS network trained without removal of units.

5 Conclusions

The exposed topology preserving visualization methods for GCS are a powerful tool for a quickly evaluation of the trained network quality.

The topographic function gives a fast idea of the quality of the trained network, clearly separating the mean number of neurons that should be neighbors and they are not and the mean value of neighbor neurons that should not be.

On the other hand, topology preserving map used along with the topographic map allows distinguish the neurons that are generating violations, as well as the gravity of the same.

GCS network usually presents better topology preservation value when removal of units is used, since it eliminates those neurons located in areas of the input space with low probability density.

The average evaluation that offers the topographic function along with the detailed one of each output neuron of the network displayed in the topology preserving map causes that jointly both graphs offer a complete vision of the quality of a network with regard to the topology preservation.

References

1. Tukey, J.W.: Exploratory data analysis. Addison-Wesley, Reading (1977)
2. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Heidelberg (2001)
3. Ultsch, A., Siemon, H.P.: Kohonen self-organizing feature maps for exploratory data analysis. In: Proceedings of the International Neural Network, Dordrecht, The Netherlands (1990)
4. Kraaijveld, M.A., Mao, J., Jain, A.K.: A non linear projection method based on Kohonen's topology preserving maps. IEEE Trans. on Neural Networks 6(3), 548–559 (1995)
5. Merlk, D., Rauber, A.: Alternative ways for cluster visualization in self-organizing maps. In: WSOM 1997, Helsinki, Finland (1997)
6. Rubio, M., Giménez, V.: New methods for self-organizing map visual analysis. Neural Computation & Applications 12, 142–152 (2003)
7. Martinetz, T., Schulten, K.: Topology representing networks. Neural Networks 7(3), 507–522 (1994)
8. Bauer, H., Pawelzik, K.: Quantifying the neighborhood preservation of self-organizing feature maps. IEEE trans. Neural Networks 3, 570–579 (1992)
9. Villmann, T., Der, R., Herrmann, M., Martinetz, T.: Topology preservation in self-organizing feature maps: exact definition and measurement. IEEE trans. Neural Networks. 8(2), 256–266 (1997)
10. Tasdemir, K., Erzsebet, M.: Data topology visualization for the self-organizing map. In: ESANN, pp. 277–282 (2006)
11. Fritzke, B.: Growing Cell Structures – A self-organizing Network for Unsupervised and Supervised Learning. Neural Networks 7(9), 1441–1460 (1994)
12. Delgado, S., Gonzalo, C., Martínez, E., Arquero, A.: Visualizing High-Dimensional Input Data with Growing Self-Organizing Maps. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 580–587. Springer, Heidelberg (2007)
13. Delgado, S., Gonzalo, C., Martínez, E., Arquero, A.: Improvement of Self-Organizing Maps with Growing Capability for Goodness Evaluation of Multispectral Training Patterns. In: IEEE International Geoscience and Remote Sensing Symposium, vol. 1, pp. 564–567 (2004)

Making Standard SOM Invariant to the Initial Conditions

Soukeina Ben Chikha[1] and Kirmene Marzouki[2]

[1] Faculty of Science of Tunis (FST), Dept of Computer Science
Campus Universitaire El Manar 2092 Tunis, Tunisia
sokyana@yahoo.fr
[2] Sousse Institute of Applied Science and Technology, Dept of Computer Science
Cité Taffala (Ibn Khaldoun), 4003, Sousse, Tunisia
marzouki.kirmene@planet.tn

Abstract. In data clustering, the assessment of learning properties with respect to data is important for a reliable classification. However, in standard Self Organizing Map (SOM), weight vectors initialization is done randomly, leading to a different final feature map each time the initial conditions are changed. To cope with this issue, in this paper, we present a behavioral study of the first iterations of the learning process in standard SOM. After establishing the mathematical foundations of the first passage of input vectors, we show how to conclude a better initialization relatively to the data set, leading to the generation of a unique feature map.

Keywords: Clustering, Self Organizing Maps (SOM), Learning process.

1 Introduction

Kohonen formulated the Self Organizing Map algorithm as a mathematical model of the self-organization of topographic maps, which are found in brains of higher animals [1]. The key idea introduced by Kohonen is the concept of neighborhood [2]. Each unit has a set of neighbors on the map. After finding the best matching unit (BMU) to the input vector, not only its weight is adjusted, but those of its neighbors are also changed. SOM accomplishes two things: It reduces dimensions and displays similarities [3].

One of the limitations of the standard Self Organizing Map algorithm is that every SOM is different and finds different similarities among the sample vectors each time the initial conditions are changed. Thus, the output map provided by standard SOM depends highly on the initial conditions and the position of the winning unit on the feature map. Consequently, a lot of maps need to be constructed in order to get one final good map, which is a tough and time consuming task.

There are numbers of approaches in the literature which tried to deal with this problem [4-7].

We believe that a deeper understanding of the learning process, helps avoiding the above limitation of standard SOM.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 204–211, 2009.
© Springer-Verlag Berlin Heidelberg 2009

In this paper, we present a behavioral study of the feature map, with respect to the passage of the different input vectors during the very first iterations of the whole process. The resulting mathematical foundations, applied to color classification or Breast Cancer Classification, make the resultant mapping invariant to the initial conditions.

2 Behavioral Study of the Learning Process in Standard SOM

2.1 Observations and Motivation

The observations of the evolution of the state of the feature map during learning progress, iteration by iteration, are plotted in Figure 1.

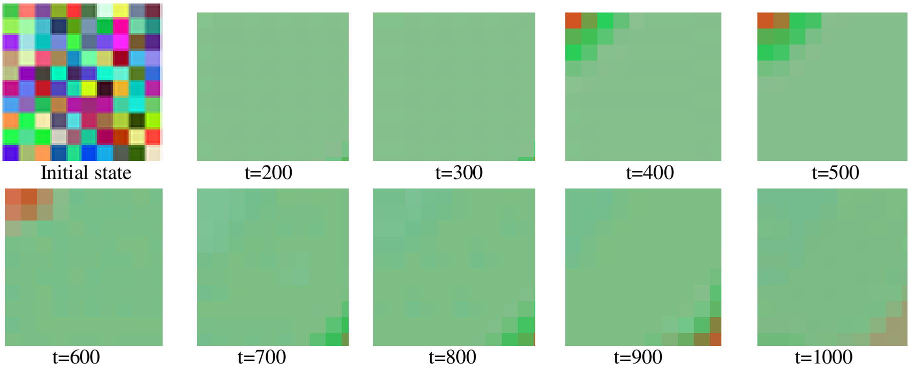


Fig. 1. Network evolution during the first 1000 iterations in standard SOM

From the above observations, we notice that the feature map, from the very first iterations, splits into two regions, one larger than the other. This is attributed to the effect of the neighborhood function, *Nc*, initially big, and its shrinking through time.

One other important observation is the dominance of the same color on the map. We found this change not only fast, but also presenting a big contrast with respect to the initial state.

The above observations are general for all simulations we've run. Only the partition of the two regions and their proportions are different from one simulation to another.

This leads us to conclude that the very first steps of learning are of big importance to the fate of the feature map. In order to well understand the "*hows*" of the inside mechanism of learning, especially the evolution of the very first steps of it, we suggest to perform a mathematical study of the phenomenon.

2.2 Mathematical Foundations

In the following, we consider the first time epoch of the learning process. We remind the reader that one epoch is the time during which all sample vectors are fed into the network.

For a unit *i* of the network, we traced its weight modifications due to the passage of each sample vector of the input space.

Unit *i* will see its weight modified according to Kohonen learning rule as follow:

$$W_i\,(t) = W_i\,(t\text{-}1)+\alpha(t\text{-}1)\ [x(t) - W_i\,(t\text{-}1)]$$

$$W_i\,(t) = (1\text{-}\ \alpha(t\text{-}1))\ W_i\,(t\text{-}1) + \alpha(t\text{-}1)\ x(t) \tag{1}$$

where $W_i\ (t)$ is the newly updated weight vector of the unit *i*, $\alpha(t)$ is learning rate, $x(t)$ is the input vector and *t* is the current time epoch.

We introduce *j* to represent the input vector rank. Thus (1) becomes:

$$W_i^j\,(t) = (1\text{-}\ \alpha(t\text{-}1))\ W_i^{j\text{-}1}\,(t\text{-}1) + \alpha(t\text{-}1)\ x^j(t) \tag{2}$$

where $W_i^j\ (t)$ is the weight vector of unit *i* after the passage of the j^{th} input vector $x^j(t)$ during time epoch *t*.

For the first time epoch, $t=1$, $\alpha(t\text{-}1) = \alpha(0) = \alpha_0$ and the first sample vector $j=1$, (2) becomes:

$$W_i^1\,(1) = \ (1\text{-}\ \alpha_0)\ W_i^0\,(0)+\alpha_0\ x^1(1)$$

where W_i^0 (0) is the initial value of the weight vector of unit *i*. Since input vectors are the same during all time epochs of the learning, $x^j(t)$ can be changed to x^j. Thus the last equation becomes.

$$W_i^1\,(1) = \ (1\text{-}\ \alpha_0)\ W_i^0\,(0)+\alpha_0\ x^1$$

Let $S_1= (1\text{-}\alpha_0)$ and $p_0= \alpha_0$

$$W_i^1(1) = S_1\ W_i^0\,(0)+p_0\ x^1 \tag{2'}$$

After the passage of the second sample vector (2) becomes:

$$W_i^2\,(1) = (1\text{-}\ \alpha_0)W_i^1\,(1)+\ \alpha_0\ x^2 = (1\text{-}\alpha_0)\ S_1\ W_i^0\,(0)+\ (1\text{-}\alpha_0)\ p_0x^1 +\ \alpha_0\ x^2$$

Let $S_2= (1\text{-}\alpha_0)\ S_1$ and $p_1= (1\text{-}\alpha_0)\ p_0$

$$W_i^2\,(1) = S_2\ W_i^0\,(0) + p_1\ x^1 + p_0\ x^2 \tag{2''}$$

After the passage of the third sample vector (2) becomes:

$$W_i^3\,(1) = (1-\alpha_0)W_i^2(1)+\alpha_0\, x^3 = (1-\alpha_0)\, S_2 W_i^0\,(0) + (1-\alpha_0)p_1\, x^1 + (1-\alpha_0)p_0\, x^2 + \alpha_0\, x^3$$

Let $S3= (1-\alpha_0)\, S2$ and $p_2= (1-\alpha_0)\, p_1$

$$W_i^3\,(1) = S_3\, W_i^0\,(0) + p_2\, x^1 + p_1\, x^2 + p_0\, x^3 \tag{2'''}$$

From equations (2'), (2") and (2'''), we notice that each time we feed an input vector x^j to the network, unit i will see its weight vector modified in a *geometric progression-like* manner.

At this point, it is crucial and very important to point out that we are not trying to reduce the learning process to a simple mathematical geometric progression as in [8] where they involve Markov chain, but we are just trying to understand the evolution of the weight vectors during the first time epoch of training.

Let *GP1* be a geometric progression with initial value $S_0=1$ and common ratio $q=(1-\alpha_0)$ with general form:

$$S_{n+1} = (1-\alpha_0)\, S_n \tag{3}$$

Also, let *GP2* be a geometric progression with initial value $p_0= \alpha_0$ and with the same common ratio $q=(1-\alpha_0)$ with general form:

$$p_{n+1} = (1-\alpha_0)\, p_n \tag{4}$$

After the learning of all the vectors of the dataset the weight vector of a unit i will be:

$$W_i^n\,(1) = S_n\, W_i^0\,(0) + p_{n-1}\, x^1 + \ldots + p_{n-i}\, x^i + \ldots + p_2\, x^{n-2} + p_1\, x^{n-1} + p_0\, x^n \tag{5}$$

As *GP1* and *GP2* are geometrical progressions with the same common ratio $|q=(1-\alpha_0)| < 1$, the different terms of the progressions converge to 0.

As n is the number of the training dataset, which is usually large, and as in any geometric progression, we assume that the $(n-3)$ first terms converge to zero, and only the three last ones are significant. Thus, equation (5) reduces to:

$$W_{ini} = p_2\, x^{n-2} + p_1\, x^{n-1} + p_0\, x^n \tag{6}$$

That is, during the first time epoch, where *Nc* still big and covers all the unit space, and after feeding all input vectors to the network, only the combination of the last three ones will affect the weight vector of unit *i*, which explains the appearance of the feature map on figure 1 during the very first time epochs.

In fact, this observation is also true for all time epochs till the end of training, but the degree of the influence of the last three training vectors gets minor from one epoch to another with respect to the evolution of the network.

Inspired from the above observations, and in order to eliminate the dependency of the final feature map on the initial conditions, we assume that it would be better to initialize the network according to equation (6), which depends on the training data set. Moreover, proceeding this way, we also may be able to gain on the training time.

2.3 Experimental Results

For a matter of clarity, the first chosen application to test the new algorithm consists of color mapping. As SOM produces a mapping of classified data, it is then easy to evaluate how good a map is and how strong the similarities between objects are. In the case of color classification, it is very easy to check the resulted map and make comparisons.

Simulations were performed with different network topologies with different number of training data and different iteration numbers. The training data and initial states were three-dimensional vectors on the RGB scale having the form $X_1=(R_1,G_1,B_1)$, $X_2=(R_2,G_2,B_2)$… $X_N=(R_n,G_n,B_n)$, where $R_m \in [0,255]$, $G_m \in [0,255]$ and $B_m \in [0,255]$.

The shown results were performed on a 10x10 net lattice using 1000 training data for 5000 iterations. Representative examples of the used initial data sets are represented in Figure 2.

The modified initial state makes the resultant mapping invariant to the initial conditions.

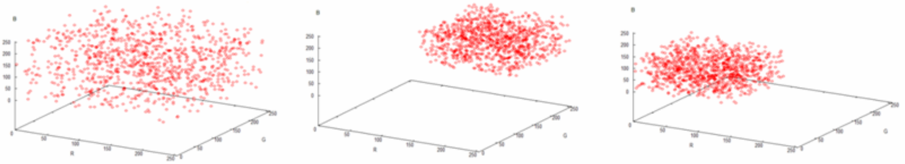


Fig. 2. Used different distributions of the training data set

We've run two series of simulations. In the first one, standard SOM, initial states were drawn randomly. In the second one, the initial states were drawn according to equation (6), deduced from the training dataset, separating thus, the dependency of the final result from the initial conditions as we assumed in section 2.2.

Final results are shown Figure 3 below.

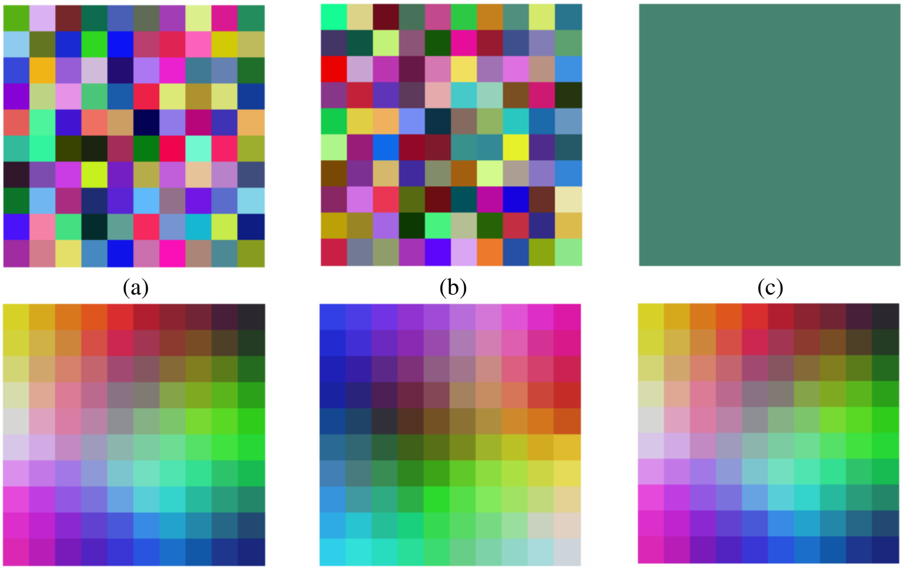

(a) (b) (c)
(d) (e) (f)

Fig. 3. Color Classification simulation: Results by the standard SOM (d) and (e) with the initial state (a) and (b), the same result obtained (f) with the introduced initial state (c)

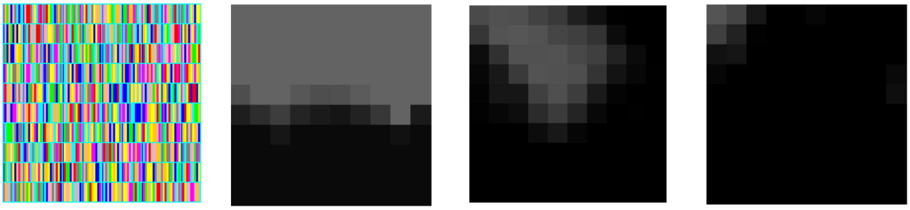
As expected, standard SOM gave different results each time we changed the initial conditions.

Simulations in which initial states were drawn according equation (6), gave a unique final feature map. In the case of Figure 3, we obtained the same final state as in standard SOM.

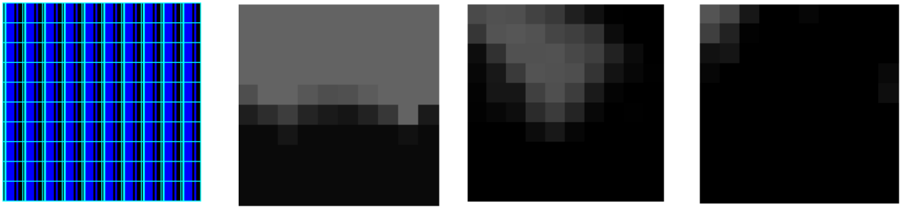
Adopting the equation (6) to draw the initial state we succeeded to separate the final result from the initial conditions. The introduced initial state was established according to the study of the internal progression of the learning process. It is very important to point out that learning is processed exactly like in standard SOM, and only initialization is different.

The main goal of the second cased study is to find out if our approach still working with high dimensional input vectors. For this purpose, we chose to perform breast cancer classification. The used datasets contains several numerical attributes and a single categorical attribute (class). An example of such dataset is the *Wisconsin Breast Cancer (WBC)* [9] dataset available from the UCI Machine Learning Repository. We first give brief introduction to *WBC* dataset. The *WBC* dataset has 10 numerical non-class attributes and one categorical class attribute. The class attribute has two categorical values, “1” and “10”. Out of 10 numerical attributes we consider only nine excluding ID. The 9 numerical attributes are: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitosis. Each of these attributes draws an integer value from the range 1 to 10 inclusive. Each sample is either benign or malignant. The training data and initial states were 10-dimensional vectors having the form $X_m=(x_1,x_2,\ldots,x_i,\ldots, x_{10})$, where $x_i \in [1,10]$.

We've run two series of simulations as in the first case study. After the learning process we color each neuron according with each component value in the code vector. Therefore, we will take a colored SOM for each variable (using grey scale) (Fig 4).



(a) SOM initial state (b) CP0 (c) CP7 (d) CP9



(e) Initial state (f) CP0 (g) CP7 (h) CP9

Fig. 4. Breast Cancer Classification simulation: Components planes for the attributes 1,7 and 9 of the weight vector. Results by the standard SOM (b),(c) and (d) with the initial state (a), the same result obtained (f), (g) and (h) with the introduced initial state (e).

Through these components planes (CP) we can realize emerging patterns of data distribution on SOM's grid, detect correlations among variables and the contribution of each one to the SOM differentiation only viewing the colored pattern for each Component Plane. Analyzing that map we identify that the attribute 0 is influent and that the last one do not define a cluster.

Finally, as expected, standard SOM gave different results each time we changed the initial conditions.

Simulations in which initial states were drawn according equation (6), gave a unique final feature map. In the case of Fig. 4, we obtained the same final state as standard SOM.

Moreover, proceeding this way, we could realize about 20% gain on the training time comparing to standard SOM, which is very considerable.

3 Conclusion

This paper addresses one of the limitations of standard SOM, that is, the dependency of the resulting feature map on the initial conditions.

We developed the mathematical foundations of the first time epoch of the learning process in standard SOM. We could express the update of the weight vectors in a form of a geometric progression. We found out that only the three last fed input vectors (training data set) dominate the update.

In order to make the final map independent from the initial conditions, we assumed that if we draw the initial conditions accordingly, the resulting map will be invariant.

For a matter of visualization, experiments were performed on an RGB color classification. Then, to find out if our approach still working with high dimensional input vectors, experiments were performed on breast cancer classification (10-dimensional vectors).

The obtained results were according our hypothesis. The resulting final feature map is unique and does not depend on the initial conditions. Moreover, proceeding this way, we could gain about 20% of the training time, which is considerable.

Finally, the results were according to our hypothesis but we still looking after an explanation of the observed phenomenon.

References

1. Kohonen, T.: Self-Organization and Associative memory, 3rd edn. Springer, Heidelberg (2001)
2. Kohonen, T.: Self Organizing Maps. Springer, Berlin (1995)
3. Kohonen, T.: The Self-Organization Map. Proceedings of the IEEE 78, 1464–1480 (1990)
4. Hecht-Nielsen, R.: Counterpropagation networks. Appl. Opt. 26, 4979–4984 (1987)
5. DeSieno, D.: Adding a conscience to competitive learning. In: IEEE Int. Conf. Neural Networks, San Diego, CA, vol. I, pp. 117–124 (1988)
6. Huerter, G.: Solution of the traveling salesman problem with an adaptive ring. In: IEEE Int. Conf. Neural Networks, San Diego, CA, pp. 85–92 (1988)
7. Marzouki, K., Yamakawa, T.: Novel Learning Algorithm Aiming at Generating a Unique Units Distribution in Standard SOM. In: Proceedings of the International Conference on Adaptive and Natural Computing Algorithms, Coimbra, Portugal, pp. 168–172 (2005)
8. Sadeghi, A.: Asymptotic Behavior of Self-Organizing Maps with Non- Uniform Stimuli Distribution. Annals of Applied Probability 8(1), 281–289 (1998)
9. `http://www.ics.uci.edu/~mlearn/MLRepository.html`

The Complexity of the Batch Neural Gas Extended to Local PCA

Iván Machón-González[1], Hilario López-García[1], and José Luís Calvo-Rolle[2]

[1] Universidad de Oviedo. Escuela Politécnica Superior de Ingeniería.
Departamento de Ingeniería Eléctrica, Electrónica de Computadores y Sistemas.
Edificio Departamental 2. Zona Oeste. Campus de Viesques s/n.
33204 Gijón/Xixón (Asturies). Spain
[2] Universidad de A Coruña. Escuela Universitaria Politécnica.
Departamento de Ingeniería Industrial. Avda.
19 de Febrero s/n. 15405 Ferrol (A Coruña). Spain

Abstract. The adaptation of a neural gas algorithm to local principal component analysis (NG-LPCA) is a useful technique in data compression, pattern recognition, classification or even in data estimation. However, the batch NG-LPCA becomes unfeasible when dealing with high dimensional data. In this paper, a regularization method is described in detail to prevent the batch NG-LPCA approach from instability. The proposed method is tested and the results seem to prove that it is a suitable tool for classifying tasks avoiding instability with high dimensional datasets.

1 Introduction

Principal component analysis (PCA) is a well known technique for dimension reduction by means of the first principal eigenvectors of the data covariance matrix. However, applying PCA to a whole data set will be inefficient since it is a linear method and the non-linear relationships between the variables are not taken into account. This can be corrected using a local PCA approach, where the whole dataset can be described by a set of locally linear models. In this way, non-linear data distributions can be represented [1].

Neural gas (NG) algorithm [2] is an unsupervised learning algorithm that considers the cooperation-competition computation. Both cooperation and competition between the map units allow the algorithm to be prevented from local minima problems [1]. In addition, the batch NG allows fast training so that the convergence is achieved in a small number of epochs [3].

There are several combinations of these two techniques (or using Self Organizing Map instead of NG) in the literature. The most similar techniques to this proposal are [4,5]. A combination of normalized Mahalanobis distance and the squared reconstruction error is used as a metric in [5]. Standard NG approaches are based on the Euclidean distance implying that the clusters defined by the prototypes are spherical. Ellipsoidal clusters can be searched using a normalized Mahalanobis distance instead of the Euclidean metric.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 212–219, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The complexity of this batch neural gas algorithm extended to work as local PCA method is to ensure the inversion of the covariance matrix. It is necessary that this matrix should have at least m linearly independent vectors, where m is the data dimensionality. Low dimensional data will not entail instability trouble. However, the datasets are usually composed of a high amount of variables and this fact can imply serious problems of numerical instability. In [5] a Gram-Schmidt step is employed to avoid the collapse of orthogonality, but [4] do not deal with the numerical instability problems.

In this paper, we expose in detail a regularization method based on [4,5] to prevent the batch NG-LPCA approach from matrix singularity. The computation assigns the same eigenvalue to all the minor principal components making a uniform partition of the variance in those minor directions.

The paper contains the learning rules of the batch NG-LPCA algorithm in Section 2, the procedure for training the network in order to be applied in classification tasks in Section 3 and the metric to evaluate the classifier in Section 4. The proposed technique is tested in Section 5.

2 Batch NG-LPCA Algorithm

The formulation of the Batch NG-LPCA algorithm appears in [4]. A neighborhood function (1) is necessary to implement the algorithm. In this case, there are no topological restrictions as in SOM algorithm [6]. Rank function $k(\mathbf{v}, \mathrm{w}_i) \in 0, ..., n-1$ represents the rank distance between prototype w_i and data vector $\mathbf{v}$. The minimum distance takes the value 0 and the rank for the maximum distance is equal to $n-1$, where n is the number of map units or prototypes and $\sigma(t)$ is the neighborhood radius .

$$h_\sigma(\mathbf{v}, \mathrm{w}_i) = \exp\left(-\frac{k(\mathbf{v}, \mathrm{w}_i)}{\sigma(t)}\right) \tag{1}$$

The neighborhood radius $\sigma(t)$ was chosen to decrease exponentially according to (2). The decrease goes from an initial positive value σ_{t_0} to a smaller final positive value $\sigma_{t_{max}}$.

$$\sigma(t) = \sigma_{t_0} \cdot \Big(\frac{\sigma_{t_{max}}}{\sigma_{t_0}}\Big)^{t/t_{max}}, \tag{2}$$

where t is the epoch step and t_{max} is the maximum number of epochs. In all the datasets, σ_{t_0} was chosen as the half number of map units. Also $\sigma_{t_{max}} = 0.001$ for all the datasets.

The generalized metric to measure the distance d_{Λ_i} between the data vector v and the prototype vector w_i searching ellipsoidal clusters is chosen as the Mahalanobis distance in (3)

$$d_\Lambda(\mathrm{v}, \mathrm{w}_i) = (\mathrm{v}, \mathrm{w}_i) \cdot \Lambda_i \cdot (\mathrm{v}, \mathrm{w}_i)^{\mathrm{t}}, \tag{3}$$

where $\Lambda_i \in \Re^{m \times m}$ is a symmetric positive definite matrix with $\det \Lambda_i = 1$ and m is the data dimensionality. The Euclidean metric implies Λ_i equal to the identity matrix.

The energy cost function E_{NG} according to the chosen metric is written in (4). The updating rules for the prototypes vectors and the matrices appear in (5) and (6), respectively. They are obtained in [4] by means of derivation of the energy function E_{NG} with respect to the prototype vectors $\left(\frac{\partial E_{\mathrm{NG}}}{\partial \mathrm{w}_i} = 0\right)$ and to the matrix Λ_i $\left(\frac{\partial E_{\mathrm{NG}}}{\partial \Lambda_i} = 0\right)$.

$$E_{\mathrm{NG}} = \sum_{ij} h_\sigma(\mathrm{v}_j, \mathrm{w}_i) \cdot d_\Lambda(\mathrm{v}_j, \mathrm{w}_i), \tag{4}$$

$$\mathrm{w}_i = \frac{\sum_j h_\sigma(\mathrm{v}_j, \mathrm{w}_i) \cdot \mathrm{v}_j}{\sum_j h_\sigma(\mathrm{v}_j, \mathrm{w}_i)} \tag{5}$$

$$\Lambda_i = S_i^{-1} \cdot (\det S_i)^{1/m}, \tag{6}$$

where matrix S_i is calculated in (7) and corresponds to the correlation of the data centered at prototype vector w_i and weighted by means of its distance from the prototype according to the neighborhood function (1).

$$S_i = \sum_j h_\sigma(\mathrm{v}_j, \mathrm{w}_i) \cdot (\mathrm{v}_j - \mathrm{w}_i)^{\mathrm{t}} \cdot (\mathrm{v}_j - \mathrm{w}_i) \tag{7}$$

Since matrix S_i must be invertible, there must be m linearly independent vectors $\mathrm{v}_j - \mathrm{w}_i$. The main complexity of the algorithm is the requirement to update the matrix Λ_i. Low dimensional datasets usually do not produce inversion problems of S_i, but numerical instabilities arise with high dimensional data since the variance can become very small in the minor principal components of the data. This means that some eigenvalues of S_i exactly 0 should be avoided. The eigenvalues are contained in the diagonal matrix D and the corresponding eigenvectors in the columns of the $m \times m$ matrix Ω related to S_i by (8).

$$S_i = \Omega \cdot D \cdot \Omega^{\mathrm{t}} \tag{8}$$

Using the algorithm with high dimensional data and full matrix adaptation, i.e., searching the eigenvalues in all the dimensions, can cause instabilities and the execution can become unfeasible. For this reason, a regularization scheme to reduce numerical instabilities is advisable. The approach computes only k major principal components and it scales all remaining dimensions uniformly. An estimation of the variance in the remaining $m - k$ directions is given by (9) in [5]. In this way, only k principal components need to be computed explicitely.

$$\begin{aligned}\sigma^2_{n-k} = \sum_j & (h_\sigma(\mathrm{v}_j, \mathrm{w}_i) \cdot (\mathrm{v}_j - \mathrm{w}_i) \cdot (\mathrm{v}_j - \mathrm{w}_i)^{\mathrm{t}} \\ & - h_\sigma(\mathrm{v}_j, \mathrm{w}_i) \cdot (\mathrm{v}_j - \mathrm{w}_i) \cdot \Omega_k \cdot \Omega_k^{\mathrm{t}} \cdot (\mathrm{v}_j - \mathrm{w}_i)^{\mathrm{t}}),\end{aligned} \tag{9}$$

where Ω_k is a $m \times k$ matrix containing the first k principal eigenvectors in its columns. The residual variance is evenly distributed to all the minor directions according to (9). In this way, the eigenvalue λ^* is the same for all the minor components.

$$\lambda^* = \frac{\sigma^2_{n-k}}{m-k} \tag{10}$$

For each iteration, matrix S_i can be recalculated using (8) by means of substitution of the new eigenvalues λ^* in the minor components of the matrix D. Finally, matrix Λ_i is computed according to (6).

3 Proposal of Algorithm Training

The initialization is randomly choosing values from a uniform distribution on the interval (0–1). The class variable is referred to the class label that defines the membership of a data vector v to a certain class and is available for all the datasets. Obviously, that information will not be present for the test datasets and, for this reason, the class variable will be removed from the k principal components.

```
begin initialize randomly the prototypes vectors $w_i$
initialize the matrices $\Lambda_i$ as the identity matrix
do for each epoch
    determine the ranks of the prototypes $k(\mathbf{v}, w_i)$
    update all the prototypes $w_i$ according to (5)
    do for each $i$ unit
        determine $S_i$ and its eigenvalues by means of (7) and (8), respectively
        if the class variable belongs to the $k$ principal components
        then { • the minor component with the highest eigenvalue
                 becomes a principal component
               • the class variable is removed from the $k$ major
                 components and is considered as a minor component
        determine the variance in the remaining $m - k$ directions by (9)
        calculate $\lambda^*$ using (10)
        replace the old eigenvalues of the minor components by $\lambda^*$ in $D$
        compute the new $S_i$ according to (8)
        update matrix $\Lambda_i$ using (6)
    until all the units are updated
until the maximum number of epochs
end
```

4 Classifier Evaluation

The prototypes must be assigned to a certain class since the algorithm has an unsupervised learning. The component that corresponds to the class variable determines the membership of each prototype to a certain class.

The $m \times m$ matrix Λ_i, which is trained using the training dataset, is reduced to a $(m-1) \times m-1)$ matrix to be able to operate with the test data vectors. The

Table 1. Datasets used for experimental testing

Dataset	Variables	Negative samples (training)	Positive samples (training)	Negative samples (test)	Positive samples (test)	% Oversampling
glass7	9	130	21	55	8	400
hepatitis	18	66	14	27	5	200
cancer	9	311	168	133	71	100
vowel0	12	630	63	270	27	800
vehicle1	18	444	149	190	63	200
diabetes	8	350	188	150	80	200

dimensionality of these vectors is $m-1$ because the class variable is removed from them to correctly test the algorithm. Then, matrix Λ_i is normalized in order to obtain $\det \Lambda_i = 1$.

The classification of the samples is according the g-means metric [7] defined as $g = \sqrt{acc^{+} \cdot acc^{-}}$, where $acc^{+} = TP/(TP+FN)$ and $acc^{-} = TN/(TN+FP)$ are commonly known as sensitivity and specificity, respectively. The sensitivity is related to the accuracy on the positive classes, whereas the specificity shows the accuracy on the estimation of the negative classes. TP and TN are the cases where the classification is correct dealing with positive and negative samples, respectively. FN is the case of a sample which is estimated as negative when it is actually positive. Following the same reasoning, FP corresponds to the case of an actual negative sample which is predicted as positive. The aim of g-means metric is to maximize the accuracy on each of the two classes by means of keeping them well balanced.

5 Experimental Testing

Six datasets were used to check the algorithm and all of them were extracted from UCI repository of machine learning databases [8]. In all the cases, the comparison was according to the g-means metric.

All the datasets have been randomly divided into two subsets, training and testing data. 70% of the original samples are reserved for the training dataset and the rest of them (30%) belong to the testing dataset. The imbalance ratio of each dataset (ratio of negative to positive samples) has been preserved in both training and testing sets. Table 1 shows all these details. The number after the dataset name indicates the class considered as positive in those datasets that have more than 2 classes. An oversampling of the positive class was made in order to keep the sensitivity of the classifier well balanced.

The original data vectors $\mathbf{v}$ were normalized according $\mathbf{v_n}_j = (\mathrm{v}_j - \mu_j)/\sigma_j$ to obtain a normalized set of data vectors $\mathrm{v}_\mathbf{n}$; where $\mathbf{v}_j$, μ_j, σ_j are the original data, the mean value and the standard deviation of variable j, respectively. This data transformation leads to a normalized distribution with zero mean and unitary

Table 2. The results indicate the average of the g-means and the standard deviation over 10 runs. The training was for 50 epochs and the number of prototypes was equal to 2 units

Dataset	Principal components	G-means	Dataset	Principal components	G-means
glass7	0	0.927±0.000	vowel0	0	0.637±0.087
glass7	1	0.947±0.032	vowel0	1	0.501±0.364
glass7	2	0.968±0.042	vowel0	2	0.429±0.309
glass7	3	0.877±0.308	vowel0	3	0.070±0.222
glass7	4	0.867±0.308	vowel0	4	0.247±0.334
glass7	5	0.899±0.045	vowel0	5	0.135±0.297
glass7	6	0.897±0.081	vowel0	6	0.113±0.238
glass7	7	0.839±0.301	vowel0	7	0.312±0.342
glass7	8	0.811±0.291			
glass7	9	0.527±0.369	diabetes	0	0.651±0.025
			diabetes	1	0.668±0.023
hepatitis	0	0.699±0.246	diabetes	2	0.685±0.019
hepatitis	1	0.844±0.029	diabetes	3	0.668±0.021
hepatitis	2	0.737±0.263	diabetes	4	0.676±0.022
hepatitis	3	0.821±0.035	diabetes	5	0.651±0.024
hepatitis	4	0.588±0.407	diabetes	6	0.567±0.201
hepatitis	5	0.759±0.271	diabetes	7	0.558±0.198
hepatitis	6	0.847±0.016	diabetes	8	0.491±0.268
hepatitis	7	0.762±0.271			
hepatitis	8	0.681±0.360	vehicle1	0	0.619±0.000
hepatitis	9	0.597±0.415	vehicle1	1	0.623±0.006
hepatitis	10	0.684±0.362	vehicle1	2	0.622±0.002
hepatitis	11	0.499±0.431	vehicle1	3	0.623±0.006
hepatitis	12	0.607±0.328	vehicle1	4	0.627±0.003
hepatitis	13	0.385±0.412	vehicle1	5	0.628±0.002
hepatitis	14	0.297±0.386	vehicle1	6	0.629±0.004
hepatitis	15	0.238±0.384	vehicle1	7	0.629±0.005
			vehicle1	8	0.630±0.005
cancer	0	0.968±0.000	vehicle1	9	0.628±0.005
cancer	1	0.975±0.000	vehicle1	10	0.628±0.008
cancer	2	0.975±0.000	vehicle1	11	0.624±0.002
cancer	3	0.975±0.000	vehicle1	12	0.616±0.010
cancer	4	0.975±0.000	vehicle1	13	0.615±0.024
cancer	5	0.975±0.000	vehicle1	14	0.494±0.261
cancer	6	0.975±0.000	vehicle1	15	0.503±0.267
cancer	7	0.943±0.029	vehicle1	16	0.571±0.201
cancer	8	0.906±0.030	vehicle1	17	0.442±0.306
			vehicle1	18	0.320±0.338

variance. In this way, all the variables are treated by the neural network with the same importance.

The results are presented in Table 2. The algorithm is evaluated according to g-means metric. The results without principal components correspond to a

Euclidean metric. The approach has a higher g-means metric computing with principal components in all the datasets, except for *vowel0* dataset. As the number of principal components rise, the instability appears.

Moreover, good results were achieved in three datasets (*glass7*, *hepatitis* and *cancer*). In these datasets, the g-means is equal or greater than in similar works with supervised algorithms for classification of imbalanced datasets [9,10,11]. Although results using more than two map units are not shown, the same conclusions can be made for them as above.

6 Conclusions

Batch NG-LPCA can improve the clustering performance compared to standard approaches which rely on the Euclidean metric searching hyperspherical clusters. The improvement is well known and corroborated by several works of the topic literature. However, the algorithm becomes unfeasible dealing with high dimensional data and serious problems of numerical instability can take place. The complexity is to ensure the inversion of the covariance matrix. Although adding a small value to the diagonal elements of the covariance matrix is a standard method for preventing singularities, it does not guarantee the stability of the algorithm according to our experiments.

In this paper, a regularization method based on [4,5] is described in detail to prevent the batch NG-LPCA approach from matrix singularity. The computation assigns the same eigenvalue to all the minor principal components making a uniform partition of the variance in those minor directions. The proposed method is tested and the results seem to prove that instability is avoided. Moreover, it seems to be a suitable tool for classification tasks and good results were achieved in several datasets, even outperforming some supervised algorithms for classification of imbalanced datasets.

Acknowledgments

Our warmest thanks are expressed to the following for their financial support: The European Coal and Steel Community (ECSC), supporting SensorControlPilot project "Implementation of sensor based on-line control of pickling lines" with RFSP-CT-2007-00046 as agreement number.

References

1. Tipping, M.E., Bishop, C.M.: Mixtures of probabilistic principal component analyzers. Neural Computation 11(2), 443–482 (1999)
2. Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: Neural-gas network for vector quantization and its application to time-series prediction. IEEE Transactions on Neural Networks 4(4), 558–569 (1993)
3. Cottrell, M., Hammer, B., Hasenfuss, A., Villmann, T.: Batch and median neural gas. Neural Networks 19(6-7), 762–771 (2006)

4. Arnonkijpanich, B., Hammer, B., Hasenfuss, A., Lursinsap, C.: Matrix Learning for Topographic Neural Maps. In: Koutník, J., Kůrková, V., Neruda, R. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 572–582. Springer, Heidelberg (2008)
5. Möller, R., Hoffmann, H.: An Extension of Neural Gas to Local PCA. Neurocomputing 62, 305–326 (2004)
6. Kohonen, T.: Self-organizing maps, 3rd extended edn., 2001. Springer, Berlin (1995)
7. Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-sided Selection. In: Proc. 14th Int. Conf. on Machine Learning, pp. 179–186 (1997)
8. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998), `http://mlearn.ics.uci.edu/MLSummary.html`
9. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
10. Cantador, I., Dorronsoro, J.: Parallel Perceptrons, Activation Margins and Imbalanced Training Set Pruning. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3523, pp. 43–50. Springer, Heidelberg (2005)
11. Machón, I., López, H.: FLSOM with Different Rates for Classification in Imbalanced Datasets. In: Koutník, J., Kůrková, V., Neruda, R. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 642–651. Springer, Heidelberg (2008)

Self Organized Dynamic Tree Neural Network

Juan F. De Paz, Sara Rodríguez, Javier Bajo, Juan M. Corchado,
and Vivian López

Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, España
`{fcofds,srg,jbajope,corchado,vivian}@usal.es`
Department of Computer Science and Automation, University of Salamanca Plaza de la
Merced s/n, 37008, Salamanca, Spain

Abstract. Cluster analysis is a technique used in a variety of fields. There are currently various algorithms used for grouping elements that are based on different methods including partitional, hierarchical, density studies, probabilistic, etc. This article will present the SODTNN, which can perform clustering by integrating hierarchical and density-based methods. The network incorporates the behavior of self-organizing maps and does not specify the number of existing clusters in order to create the various groups.

Keywords: Clustering, SOM, hierarchical clustering, PAM, Dendrogram.

1 Introduction

The assignment of a set of objects into clusters is a widely spread problem that has been the object of investigation in various scientific branches including bioinformatics [10], surveillance [15], [16], [17]. Although occasionally the number of groups is known beforehand, clustering data requires an additional step for identifying the existing groups. There are currently different methods for creating clusters, most notably those based on partitioning, such as k-means [11], and PAM [9] (Partition around medoids), which work by minimizing the error function. Other widely accepted methods are the hierarchical methods which include dendrograms [7], agnes [9], and Diana [9]. In addition to the hierarchical methods, there are others that use density-based models, or probabilistic-based models such as EM [8] (Expectation-maximization) and fanny [9].

This research presents the new Self Organized Dynamic tree neural network which allows data to be grouped automatically, without having to specify the number of existing clusters. The SODTNN uses algorithms to detect low density zones and graph theory procedures in order to establish a connection between elements. This would allow connections to be established dynamically, thus avoiding the need for the network to expand and adjust the data surface. Additionally, the connections would continue to adapt throughout the learning process, reducing the high density neuron areas and separating them from the low density areas.

The SODTNN integrates techniques from hierarchical and density-based models that allow the grouping and division of clusters according to the changes in the

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 220–227, 2009.
© Springer-Verlag Berlin Heidelberg 2009

densities that are detected. The hierarchical process is based on the Kruskal algorithm that creates a minimum spanning tree containing data for the problem at hand. Based on the information obtained from the minimum spanning tree, low density areas are detected by using a distance matrix for each cluster. The low density areas will allow the clusters to be separated iteratively. Furthermore, the minimum spanning tree determines the network structure and connections so that learning can take place according to the tree's distribution.

This article is divided as follows: section 2 describes different clustering alternatives, section 3 describes the SODTNN, and section 4 presents the results and conclusions.

2 Clustering Techniques

The problem of clustering is far reaching, and there have been various proposals for its resolution: i) Partition based methods have the disadvantage of requiring the number of clusters up front [8]. The k-means algorithm presents problems with atypical points. The PAM method resolves this problem by assigning an existing element as the centroid. ii) Hierarchical methods such as dendrograms [7] do not require a number of clusters up front since they use a graphical representation to determine the number. iii) Probability based methods such as EM define an algorithm of probabilities that determines the probability that a point belongs to a cluster. iv) Finally, there are the methods that use changes in density in order to separate clusters. Included in these methods are the artificial neural networks (ANN) [18], [19], which estimate the surface of the point distribution by using a mesh of neurons that can be automatically adjusted to the surface. There are also other networks such as ART [5] (Adaptive Resonance Theory) that can form clusters, although it does not function based on meshes. Our research will concentrate on the mesh-based neural networks.

The self-organized Kohonen maps (SOM) [2], have variants of learning methods that base their behaviour on methods similar to the Neural Gas (NG) [4]. They create a mesh that is adjusted automatically to a specific area. The greatest disadvantage, however, is that both the number of neurons that are distributed over the surface and the degree of proximity are set beforehand. Growing Cell Structure (GCS) [3] do not set the number of neurons or the degree of connectivity, but they do establish the dimensionality of each mesh. This complicates the separation phase between groups once it is distributed evenly across the surface. There are other ANN such as SOINN [6] and ESOINN [1] (Enhanced self-organizing incremental neural network). Unlike the SOINN, ESOINN consists of a single layer, so it is not necessary to determine the manner in which the training of the first layer changes to the second.

3 SODTNN

This study proposes the SODTNN, which can detect the number of existing groups or classes and, by using the Kruskal algorithm [12], create clusters based on the connections taken from the minimum spanning tree. As opposed to the ESOINN or GCS networks, the SODTNN does not distinguish between the original data and the neurons—during the initial training phase, the latter correspond to the position for each element. This makes it possible to eliminate the expansion phase for a NG to

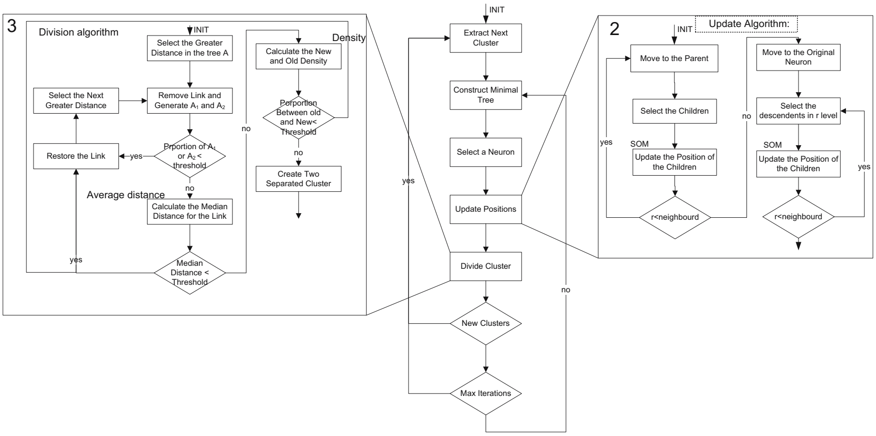

3
Division algorithm
INIT
Select the Greater Distance in the tree A
Select the Next Greater Distance
Remove Link and Generate A₁ and A₂
Prportion of A₁ or A₂ < threshold
Restore the Link
yes
no
Average distance
Calculate the Median Distance for the Link
Median Distance < Threshold
yes
no
Density
Calculate the New and Old Density
Porportion Between old and New< Threshold
no
Create Two Separated Cluster
INIT
Extract Next Cluster
Construct Minimal Tree
Select a Neuron
Update Positions
Divide Cluster
New Clusters
Max Iterations
yes
no
2
Update Algorithm:
INIT
Move to the Parent
Select the Children
SOM
Update the Position of the Children
r<neighbourd
yes
no
Move to the Original Neuron
Select the descendents in r level
SOM
Update the Position of the Children
r<neighbourd
yes

Fig. 1. Algorithm for the SODTNN

adjust to the surface. As each neuron is updated, it can draw closer to neighboring neurons, thus facilitating the detection of clusters and the separation from other elements. The learning phase for the network is illustrated in Figure 1: the main loop of the learning phase is shown in the center, block 2 is the algorithm that updates the position of the neurons, and block 3 is in charge of locating the low density areas and separating the clusters in order to create new groups.

The following sections describe the steps that are carried out during the learning phase for the ANN. The nomenclature defines T as the set of neurons to be classified, A as the minimum spanning tree that contains all of the nodes from T where matrix C defines the connections between the nodes where element $c_{ij}=1$ if node $i \in$ T is connected with element $j \in T$., D the distance matrix for T.

3.1 Density: Block 3

One of the main problems when assigning individuals into groups is knowing which divisions cause a significant rise in the density of the resulting clusters. ANN such as SOINN or ESOINN study the length of the links in order to determine if the length is different within the subgroup for each individual. This process requires the creation of subclasses within each cluster, which is done by using a set of functions that determines the threshold on which the creation of the subclasses is based. The SODTNN searches for cut-off points in areas that produce a significant rise in density. It does so by using the relationship between the total distance calculated from the distance matrix, and the distance from the minimum spanning tree.

1. Distance from tree $f^{A}(C,D)=\sum_{i,j} d_{ij}\ where\, c_{ij}=1\ , c_{ij}\in C, d_{ij}\in D$
2. Distance between neurons in the tree $f^{T}(D)=\sum_{i,j} d_{ij}, d_{ij}\in D$
3. Calculate the final density $f^{D}(C,D)=f^{T}(D)/f^{A}(C,D)$

3.2 Average Distance: Block 3

Selecting the links for finding low density areas can be done by considering the distance of one neuron with respect to its parent in tree *A*, and the average distance surrounding the neuron. The calculation of the latter distance is based on the distance that exists for each link of the subtree, where the depth is equal to the surrounding distance and centered on the neuron in question, and the number of neurons that exist in the subtree. Figure 2 illustrates the subtree, highlighted in gray to indicate the neuron that falls within 2 links.

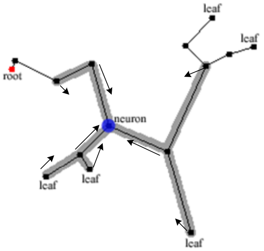


Fig. 2. Subtree for neuron falling within 2 links

The algorithm is described as follows:

1. Given a_i is the neuron for the tree for which the average distance needs to be calculated, with $i \in T$ where $f^p(a_i)$ is the function that determines the parent node for a_i, that is defined by
   $$f^p : A \to A \qquad \text{Where } c_{si} = 1 \text{ and } c_{si} \in C$$
   $$a_i \to f^p(a_i) = a_s$$
2. Apply f^p recursively and select the root node a_r from subtree $a_r = \overbrace{(f^p \circ \ldots \circ f^p)}^{e}(a_i)$
3. Establish the group of nodes within the surrounding area of a_i $A^e_{a_i} \subseteq A$ as follows $A^e_{a_i} = \{a_j \in A / \exists r \in \mathrm{N}, a_r = \overbrace{(f^p \circ \ldots \circ f^p)}^{r \le 2e}(a_j)\}$
4. Calculate the average distance for the node a_i $f^m(A^e_{a_i}, D) = \dfrac{\sum_{a_j \in A^e_{a_i}} d_{sj} \cdot c_{sj}}{\# A^e_{a_i}}$

3.3 Division Algorithm: Block 3

The division algorithm is responsible for finding the connections between the low density neurons in order to separate the cluster. It considers the distance between the neurons and the resulting changes in density for the potential divisions. The process is described as follows:

1. Determine the cut-off point for the elements α, and the cut-off points for distance β
2. Initiate $i = 1$
3. Select the greatest distance *i* for $d_{jk} \in D / c_{jk} = 1$ and remove the node from the tree $a_k \in A$
4. Given A_1, A_2 are the remaining trees alter eliminating a_k and the connection with the parent node $f^p(a_k)$ where $T_1 = \{s \in T / a_s \in A_1\}$ and $T_2 = \{s \in T / a_s \in A_2\}$ with $T = T_1 \cup T_2$, $T_1 \cap T_2 = \phi$, C_1, C_2, D_1, D_2 for the corresponding link and distance matrixes.
5. If $\#T_1 / \#T$ or $\#T_2 / \#T$ is less than α go to step 13
6. Calculate the average distance from the node for the tree a_k following the average distance algorithm $d^m_{a_k} = f^m(A^e_{a_i}, D)$
7. Determine if the distance from tree node a_k and its parent is less than the average distance $d_{sk} \le d^m_{a_k} \cdot \beta$ where $s \in T$ and $a_s = f^p(a_k)$ go to step 13
8. the density for T, T_1 and T_2 following the density algorithm $f^D(C, D)$, $f^D(C_1, D_1)$, $f^D(C_2, D_2)$
9. Calculate the new density threshold $\delta(t+1) = f^D(C_1, D_1) + f^D(C_2, D_2)$ and the previous $\delta(t) = f^D(C, D)$
10. If the value $\delta(t) / \delta(t+1) < 1/(\delta(0)/\delta(1) \cdot \rho)$ where ρ is constant, go to 12
11. Finish
12. Re-establish the connection a_k with its parent node
13. If $i < \#T$ calculate the value of $i = i + 1$ and go to step 2

3.4 Update Algorithm: Block 2

The neurons from the network that define the clusters are periodically updated in a way similar to the kohonen SOM. By updating automatically, the positions and connections of the neurons can be readjusted in order complete the division of the clusters. The network randomly selects an initial neuron and brings neighboring neurons closer in. The neuron is updated according to the hierarchy of the tree. The arrows in figure 2 indicate the direction and strength with which the neurons are brought closer to the selected neuron. The magnitude of the vector and the direction depend on the distance and neighborhood as indicated in the following algorithm:

1. Given $k \in T$ with $a_k \in A$ is the selected neuron, set the value of the neighboring radius r
2. Begin $i = 1$, $a_s = a_k$
3. Calculate the parent node from the current node $a_t = f^p(a_s)$, obtain all the sons from a_t which are defined as $A^1_{a_t}$
4. For each instance $a_j \in A^1_{a_t}$ update the coordinates for the neuron by following the equation for self-organizing maps

   $$x_j(t+1) = x_j(t) + \eta(t) \cdot g(i,t) \cdot (x_s(t) - x_j(t))$$

   Where $g(i,t)$ represents the neighboring function $\eta(t)$ the learning rate [13].

   $$g(i,t) = Exp\left[-\frac{i}{N} \frac{\sqrt{(x_{j1} - x_{s1})^2 + \ldots + (x_{jn} - n_{sn})^2}}{\underset{i,j}{Max}\{d_{ij}\}} - \lambda \frac{i \cdot t}{\beta N} \right]$$

   $$\eta(t) = Exp\left[-\sqrt[4]{\frac{t}{\beta N}} \right]$$

   Where t is the iteration, N the number of elements from group $\# A$, n is the dimension of the coordinates, x_{ij} coordinate j for the neuron $i \in T$, with $a_i \in A$, λ and β the constants established for 1 and 5 respectively.
5. If $i < r$ set $a_s = a_t$ and increase i
6. Use the same procedure to update the descendents $a_k \in A$ until reading depth r, $A^1_{a_k} \ldots A^r_{a_k}$

4 Results and Conclusions

In order to conclude the tests, both real data and fictional self-generating data were used. Additionally, graphic representations are in 2D in order to facilitate the interpretation of the results. In order to confirm that the proposed SODTNN functioned properly, the clustering process was compared with other statistical techniques traditionally use for unsupervised clustering. In the first case, we selected a test case generated with fictional data. Figure 3 illustrates the classification process that was carried out for the given test. We can see the data and the sequence of the divisions in different colors that were made with the SODTNN until the algorithm finalize, where the last image shows the results obtained after implementing the algorithm.

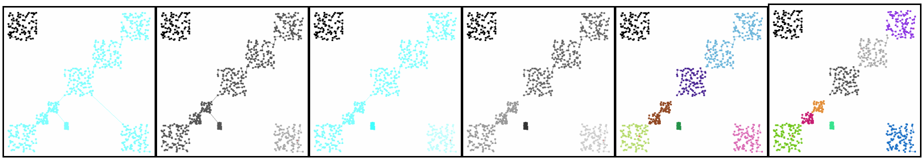


Fig. 3. Sequence of clusters generated by the ANN. Total number of points 774. Each colour represents a different cluster that was found.

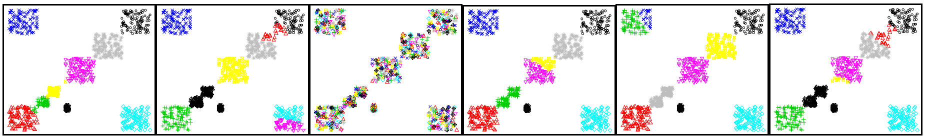


Fig. 4. Sequence of clusters generated by PAM, dendrograms, k-means, fanny, agnes, dian in this order

Figure 4 illustrates the final classification obtained by the PAM, dendograms, k-means, fanny, agnes and diana clustering methods. For each of these methods, it was necessary to determine the number of clusters that we wanted to obtain, for which it was necessary to provide more information than with the SODTNN. Additionally, it is easy to see how only the PAM method is capable of obtaining results comparable to our network, although the green group includes elements from other clusters. The rest of the processes generate classifications that could be considered as erroneous given the distribution of the data.

Secondly, we studied a real case from the UC Irvine Machine Learning Repository [14] regarding data for wines. The data within the range [0-1] was normalized in order to eliminate the scale factor and units. The classification process was then carried out. The percentage of success was as follows: 91,01%, 90,45%, 93,26%, 94.94%, 33.71%, 71.35% for the SODTNN, PAM, dendrogram, k-means, agnes and diana respectively. Fanny did not produce any results since it included data approaching 0. As we can see, the network provided results similar to the PAM, dendograms and k-means methods, while the others provided worse results. In order to analyze the elements that the ANN and PAM classified incorrectly as compared to the dendograms and k-means, we created a 3D representation and applied a multidimensional scaling process to reduce the dimensionality. The results demonstrated that the errors were atypical elements that were located outside of both clusters that would have been eliminated with a filtering phase.

The results obtained with the SODTNN are promising. Nevertheless, we have detected several deficiencies in the case of elements that are distributed along very close parallel lines. Occasionally, the SODTNN is incapable of calculating the correct cut-off point for dividing clusters, thus functioning as a hierarchical algorithm for which the user must interpret the results. The results can be interpreted according to the distances from the cut-off points and the changes in density. In order to resolve this problem, we are working on defining criteria for a cut-off point based on the calculation of the densities of the clusters.

Acknowledgements. This development has been supported by the projects SA071A08 and SIAAD-TSI-020100-2008-307.

References

[1] Furao, S., Ogura, T., Hasegawa, O.: An enhanced self-organizing incremental neural network for online unsupervised learning. Neural Networks 20, 893–903 (2007)

[2] Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics, 59–69 (1982)

[3] Fritzke, B.: A growing neural gas network learns topologies. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 625–632 (1995)

[4] Martinetz, T., Schulten, K.: A neural-gas network learns topologies. Artificial Neural Networks 1, 397–402 (1991)

[5] Carpenter, G.A., Grossberg, S.: The ART of adaptive pattern recognition by a self-organizing neural network. IEEE Trans. Computer, 77–88 (1987)

[6] Shen, F.: An algorithm for incremental unsupervised learning and topology representation. Ph.D. thesis. Tokyo Institute of Technology (2006)

[7] Saitou, N., Nie, M.: The neighbor-joining method: A new method for reconstructing phylogenetic trees. Mol. Biol. 4, 406–425 (1987)

[8] Xu, L.: Bayesian Ying–Yang machine, clustering and number of clusters. Pattern Recognition Letters 18(11-13), 1167–1178 (1997)

[9] Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)

[10] Corchado, J.M., De Paz, J.F., Rodríguez, S., Bajo, J.: Model of experts for decision support in the diagnosis of leukemia patients. Artificial Intelligence in Medicine (in Press)

[11] Hartigan, J.A., Wong, M.A.: A K-means clustering algorithm. Applied Statistics 28, 100–108 (1979)

[12] Campos, R., Ricardo, M.: A fast algorithm for computing minimum routing cost spanning trees 52(17), 3229–3247 (2008)

[13] Bajo, J., De Paz, J.F., De Paz, Y., Corchado, J.M.: Integrating case-based planning and RPTW neural networks to construct an intelligent environment for health care. Expert Systems with Applications 36(3) (2009)

[14] UC Irvine Machine Learning Repository, `http://archive.ics.uci.edu/`

[15] Patricio, M.A., Carbó, J., Pérez, O., García, J., Molina, J.M.: Multi-Agent Framework in Visual Sensor Networks. EURASIP Journal on Advances in Signal Processing, special issue on Visual Sensor Networks, 21 (2007)

[16] Carbó, J., Molina, J.M., Dávila, J.: Fuzzy Referral based Cooperation in Social Networks of Agents. AI Communications 18(1), 1–13 (2005)

[17] García, J., Berlanga, A., Molina, J.M., Casar, J.R.: Methods for Operations Planning in Airport Decision Support Systems. Applied Intelligence 22(3), 183–206 (2005)

[18] Pavón, J., Arroyo, M., Hassan, S., Sansores, C.: Agent-based modelling and simulation for the analysis of social patterns. Pattern Recognition Letters 29, 1039–1048 (2008)

[19] Pavón, J., Gómez, J., Fernández, A., Valencia, J.: Development of intelligent multi-sensor surveillance systems with agents. Robotics and Autonomous Systems 55(12), 892–903 (2007)

Development of Neural Network Structure with Biological Mechanisms

Samuel Johnson, Joaquín Marro, Jorge F. Mejias, and Joaquín J. Torres

Instituto Carlos I de Física Teórica y Computacional,
and Departmento de Electromagnetismo y Física de la Materia,
University of Granada, Spain
`samuel@onsager.ugr.es`
`http://ergodic.ugr.es/neurofisica/neurofisica.html`

Abstract. We present an evolving neural network model in which synapses appear and disappear stochastically according to bio-inspired probabilities. These are in general nonlinear functions of the local fields felt by neurons—akin to electrical stimulation—and of the global average field—representing total energy consumption. We find that initial degree distributions then evolve towards stationary states which can either be fairly homogeneous or highly heterogeneous, depending on parameters. The critical cases—which can result in scale-free distributions—are shown to correspond, under a mean-field approximation, to nonlinear drift-diffusion equations. We show how appropriate choices of parameters yield good quantitative agreement with published experimental data concerning synaptic densities during brain development (synaptic pruning).

Keywords: Neural networks, Brain development, synaptic pruning.

1 Introduction

Neural systems, whether natural or artificial, are paradigmatic cases of complex systems consisting of many interacting dynamical elements [1,2]. The phenomenology that ensues depends not only on the nature of the neurons and synapses [3], but also on the topology of the underlying network, with the result that structure influences function [4]. In the case of biological neural networks, the inverse also appears to be true: topology is dynamic and related to neural activity. This relation is probably very complex; however, we will try to account for the some of main mechanisms described in the biological literature that lead to the forming or elimination of synapses.

One of the most striking features of brain development is the systematic and relatively rapid net reduction in the density of synapses undergone as of a very early age—resulting, in humans, in adult brains with about half the synaptic density of newborns [5]. Another interesting feature is synaptogenesis, which has been related to various interacting local influences such as the concentration

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 228–235, 2009.
© Springer-Verlag Berlin Heidelberg 2009

of different neurotransmitters and electrical stimulation [6,7]. On the other hand, synapses can suffer atrophy and die, probably if not potentiated enough by use.

Chechik *et al.* have proposed a model for synaptic pruning based on Hebbian learning in which the weakest synapses are systematically removed [8,9], and show how this scheme allows for a reduction in synaptic density while preserving near-optimal properties. However, this research is not concerned with topological aspects of network evolution. Meanwhile, in the field of complex networks, many authors have studied a variety of evolving network models (for a review, see for example Boccaletti *et al.* [10]), in which nodes and/or edges appear or disappear according to local rules such as preferential attachment. In line with this, we here endeavour to show how a rather general evolving network model which mimics biological mechanisms can result in topological features which are in good accord with experimental data.

2 The Model

We consider a neural network with N neurons. The adjacency matrix $\epsilon_{ij} = \{1, 0\}$ defines whether there is some form of synaptic interaction between neurons i and j (if $\epsilon_{ij} = 1$, we say i and j are *neighbours*). Each neuron can then be characterised by its *degree*, $k_i = \sum_j \epsilon_{ij}$. The strength of the interaction is determined by the weights ω_{ij} which typically will store information via the application of some learning rule [11,12]. The state of neuron i is described at each time t by its *activity* s_i, which for concreteness we will consider to be a binary variable as in a Hopfield model [1]. The dynamics for the activity of neuron i responds to the total incoming *field* $h_i \equiv \sum_j \epsilon_{ij}\omega_{ij}s_j$ it receives at each time from its neighbours. One can then assume particular transition rates for the activities, s_i, as functions of the fields, h_i. However, for the purposes of this study we need not assume any particular form for this dependence, since the conclusions are valid independently of the rates used.

The adjacency matrix also has a dynamics, based on a combination of local and global rules for the emergence and disappearance of edges—representing, for example, growth and death of synapses. Initially, we will have a random network where the degrees k_i follow some distribution $p(k, t = 0)$—i.e., edges are placed randomly among the nodes until this distribution is achieved, implying a total of $\langle k\rangle N/2$ edges, where $\langle\cdot\rangle \equiv N^{-1}\sum_i(\cdot)$ stands for an average over neurons. Every time step, each neuron has a probability, to be defined, of gaining a new synapse, P_i^{gain}, to a random neuron. It also has a probability of loosing a (randomly chosen) synapse, P_i^{lose}. To define these probabilities, we will take into account the two mechanisms which are widely thought to define synaptic growth and death in biological neural networks. Firstly, Chechik *et al.* [8,9] have proposed that the phenomenon known as *synaptic pruning*, whereby the mean synaptic density in the brain during development drops considerably, could be related to energy conservation requirements. In our model, the total energy E (or total current) can be identified with the mean value of the field, $E = \sum_i h_i$. The second mechanism is that by which synaptic growth is stimulated by local electrical activity [6,7]. This would correspond here, for a given neuron i, to the field it feels, h_i. Taking both these consiedrations into account, we will assume that the probabilities factorise as follows:

$$P_i^{\text{gain}} = u(E)\pi(h_i)$$

and

$$P_i^{\text{lose}} = d(E)\sigma(h_i),$$

where both E and h_i are time-dependent, although for clarity this is not explicitated. The functions $\pi(h_i)$ and $\sigma(h_i)$ can be any (in general, nonlinear) functions of the field of neuron i. The terms $u(E)$ and $d(E)$ can also be arbitrary functions.

In a mean-field approximation, we can treat the dynamics for the synapses independently of the activity, and therefore render our results very general, simply by assuming that the field of a node is proportional to its degree. This can be derived formally as follows. The expected value of the adjacency matrix for a network in the configuration ensemble[1] is $[\epsilon_{ij}] = k_i k_j / (\langle k \rangle N)$, where the expected value operator $[\cdot]$ can be interpreted as an average over all configurations in the ensemble. Inserting this value in the definition of h_i yields $h_i = k_i \mu_i$, where for large networks the term $\mu_i \equiv (\langle k \rangle N)^{-1} \sum_j \omega_{ij} k_j s_j$ can be considered independent of i as long as the weights ω_{ij} are statistically independent of k_i and k_j [4]. By the same reasoning, the total energy is proportional to the mean degree of the network, $E \propto \langle k \rangle$ (though see note[2]).

Under this approximation, we can now write the probabilities for growth and death as

$$P_i^{\text{gain}} = u(\langle k \rangle)\pi(k_i)$$

and

$$P_i^{\text{lose}} = d(\langle k \rangle)\sigma(k_i),$$

The local probabilities $\pi(k)$ and $\sigma(k)$ correspond to preferential attachment and detachment, similar to those used by Barabási and Albert for their evolving network model [13] and later implemented in many models. For example, we have already studied the case of nonlinear preferential rewiring of one edge at a time [14] while maintaining the number of nodes and edges in the network fixed (equivalent, over large enough times, to keeping $P_i^{\text{gain}} = P_i^{\text{lose}} = 1$). Here we report on the main results concerning this more general scenario. A more detailed and extensive analysis is underway [15].

3 General Results

The probabilities P_i^{gain} and P_i^{lose} a given neuron i has, at each time step, of increasing or decreasing its degree can be interpreted as transition probabilities between states. Furthermore, for each synapse that is withdrawn form the

---

[1] This is the collection of all possible network configurations which respect a given degree sequence $\{k_1, ...k_N\}$ but are otherwise randomly wired.

[2] Note that this is possible because we are considering N to remain constant. It is known that in reality neurons can die and also be replenished. However, in this simplified model we are neglecting this effect.

network, two neurons decrease in degree. One is neuron i chosen according to $\sigma(k_i)$, the other, say neuron j, is a random neighbour of i's; therefore, there is an added effective probability of loss $k_j/(\langle k\rangle N)$. Similarly, for each synapse placed in the network, not only neuron l chosen according to $\pi(k_l)$ increases its degree; a random neuron m will also gain, with the consequent effective probability of (approximately) N^{-1}. Thus, by summing over all these probabilities, we can obtain an equation for the expected value of the increment in a given $p(k,t)$ at each time step, $\Delta p(k,t)$, which we will equate with a temporal partial derivative:

$$\frac{\partial p(k,t)}{\partial t} = u(\langle k\rangle)\left[\pi(k-1)+\frac{1}{N}\right]p(k-1,t)+d(\langle k\rangle)\left[\sigma(k+1)+\frac{k+1}{\langle k\rangle N}\right]p(k+1,t)- -\left\{u(\langle k\rangle)\left[\pi(k)+\frac{1}{N}\right]+d(\langle k\rangle)\left[\sigma(k)+\frac{k}{\langle k\rangle N}\right]\right\}p(k,t) \quad (1)$$

Assuming that $p(k,t)$ evolves towards a stationary distribution, $p_{st}(k)$, we can set Eq. (1) equal to zero and, after again substituting a difference for a partial derivative, obtain a condition for stationarity:

$$\frac{\partial p_{st}(k)}{\partial k} = \frac{u(\langle k\rangle)}{d(\langle k\rangle)}\left[\frac{\pi(k)+\frac{1}{N}}{\sigma(k+1)+\frac{k+1}{\langle k\rangle N}}-1\right]p_{st}(k). \quad (2)$$

In fact, $p_{st}(k)$ must also be such that $u(\langle k\rangle)=d(\langle k\rangle)$ (since the total number of synapses must then be conserved) with $\langle k\rangle=\sum_k kp_{st}(k)$. From Eq. (2) we can see that $p_{st}(k)$ will have an extremum at some value k_e if it satisfies $\pi(k_e)+\frac{1}{N}=\sigma(k_e+1)+\frac{k_e+1}{\langle k\rangle N}$. Assuming, for example, that there is one and only one such k_e, then, depending on the concavity of $p_{st}(k)$ at this point, it will correspond to a maximum (implying a relatively homogeneous distribution) or a minimum (with the result that $p_{st}(k)$ will be split in two, and therefore highly heterogeneous). The critical case separating these two regimes occurs when $\pi(k)$ and $\sigma(k)$ are such that $\pi(k)+N^{-1}=\sigma(k)+k/(\langle k\rangle N)$, $\forall k$. For this critical choice, Eq. (1) can be shown [15] to reduce to a nonlinear drift-diffusion equation, with a non-uniform velocity $c=u(\langle k\rangle)-d(\langle k\rangle)$ in the increasing k direction.

To the best of our knowledge, this is the first dynamic network model to be proposed and studied in which the rewiring actions respond to completely general nonlinear functions of local degrees and the global mean degree.

4 The Effects of Drift: Application to Synaptic Pruning

Let us define $\langle k\rangle \equiv \kappa(t)$ and assume the following linear forms for $u(\langle k\rangle)$ and $d(\langle k\rangle)$:

$$u(t)=\frac{n}{N}\left(1-\frac{\kappa(t)}{\kappa_m}\right), \quad (3)$$

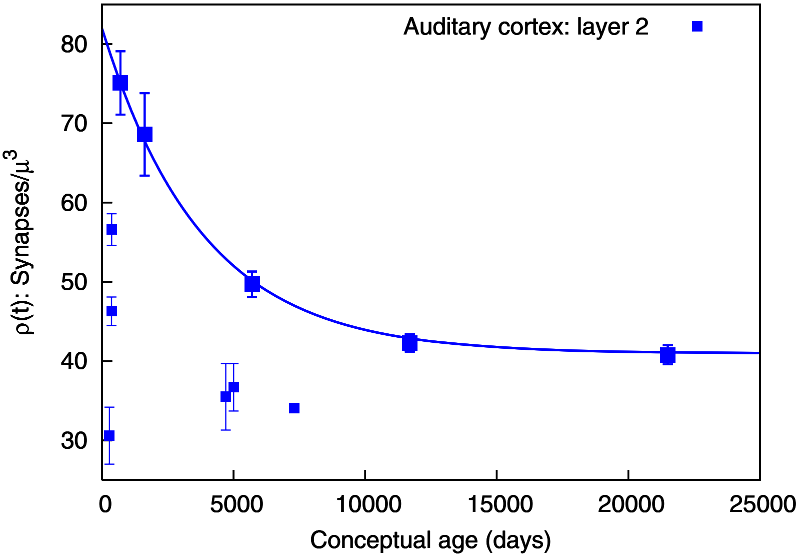


Fig. 1. Synaptic pruning in cortical layer 2 of the human auditory cortex. Experimental data from Huttenlocher and Dabholkar [5]. The line follows the best fit to Eq. (6), where parameters are $\rho(0) = 81.93$ and $\nu = 0.0107$, measured in synapses/μm^3 and synapses/μm^3/day, respectively. Correlation was $r^2 = 0.906$. Data pertaining to the rapid overgrowth of the first year after conception were omitted from the fit, as were those measured at days 4700, 5000 7300—which, for unknown reasons, fall abnormally far below the line marking the general tendency. These data points are plotted with smaller squares than the rest.

$$d(t) = \frac{n}{N}\frac{\kappa(t)}{\kappa_m}, \tag{4}$$

where the parameter n can be interpreted as the expected value of the number of additions and deletions of synapses per MCS and κ_m is the maximum value the mean degree can have. This choice describes a situation in which the higher the density of synapses, the less likely new synapses are to “sprout” and the more likely existing synapses are to “atrophy“. The increment in $\kappa(t)$ is

$$\frac{d\kappa(t)}{dt} = 2[u(t) - d(t)] = \frac{2n}{N}\left[1 - \frac{2\kappa(t)}{\kappa_m}\right], \tag{5}$$

where the factor 2 after the first equality appears because for each addition or deletion of a synapse, the degrees of two neurons are modified. We have, therefore, that temporal evolution of the mean degree will increase or decrease exponentially from $\kappa(0)$ to $\kappa_m/2$. Defining the mean synaptic density in some volume V, $\rho(t) \equiv \kappa(t)N/(2V)$ (which is the magnitude usually measured experimentally), and assuming for simplicity that $\kappa(0) = \kappa_m$ (or $\rho(0) = \rho_m$), the solution of Eq. (5) is

$$\rho(t) = \frac{1}{2}\rho(0)\left[1 + \exp\left(-\frac{2\nu}{\rho(0)}t\right)\right]. \tag{6}$$

where $\nu \equiv n/V$ (the only parameter) is the number of synapses modified (added plus deleted) per unit volume per unit time.

This equation is fitted to experimental data on layer 2 of the human auditory cortex, obtained by Huttenlocher and Dabholkar [5], and shown in Fig. 1. Time is measured in days since conception and synaptic density is in synapses per cubic micron. We assume that the initial overgrowth is governed by other factors and use Eq. (6) only as of the onset of synaptic pruning. In this way we can estimate a value of $\nu \simeq 0.01$ synapses per μ^3 per day. The data for three particular days does not seem to fit the general tendency very well. We do not know what the source of these deviations is, but it is unlikely that densities actually fluctuate to that extent within one individual. Rather, it seems more probable that they correspond to data from subjects with inherently lower synaptic density (it is important to note that data points corresponding to different times were taken from different subjects).

5 The Effects of Diffusion: Heterogeneous Topologies

Figure 2 shows typical stationary degree distributions obtained in the three regimes that emerge: in the subcritical regime, the distribution remains relatively homogeneous; in the supercritical regime, a phenomenon akin to gelation in polymers occurs, in which a small number of neurons is connected to most of the network. In the critical case, $p(k)$ is seen to evolve towards a scale-free stationary state, $p_{st}(k) \sim k^{-2}$, as is characteristic of second order phase transitions. Interestingly, the functional topology, as defined by correlated activity between clusters of neurons, in the human cortex during cognitive tasks has also been found to acquire a scale-free distribution with exponent $\gamma \simeq -2$ [16]. The same authors have argued that the reason for this is that the brain maximises its performance in a complex world by becoming critical. Recent theoretical work [4] also suggests that random topologies with distribution $p_{st}(k) \sim k^{-2}$ can result in optimal performance for neural networks executing dynamical tasks.

It is still not clear what kind of degree distribution the structural topology of the brain follows. However, it seems that function reflects structure at least to some extent [17]. Furthermore, it has been suggested, based on indirect methods, that the structural connectivity of cat and macaque brains, at the level of brain areas, may indeed be scale free [18] — and in any case displays significantly higher heterogeneity than that of, say, Erdős-Rényi random graphs.

6 Discussion

We have presented a very general evolving neural network model in which local and global rules for synaptic growth and death are coupled to the local fields and

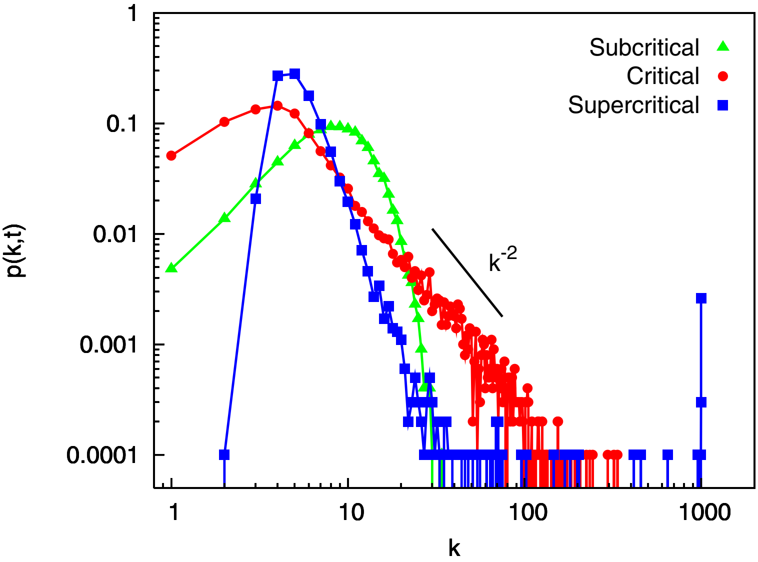


Fig. 2. Stationary distributions $p_{st}(k)$ obtained from MC simulations of the model after 10^5 MCS. The networks all started as regular random graphs with $\kappa(0) = 20$. Local probabilities were $\sigma(k) = k/(\langle k \rangle N)$ in all cases and $\pi(k) = \sigma(k)$, $\pi(k) = 2\sigma(k) - N^{-1}$ and $\pi(k) = k^{1.5}/(\langle k^{1.5} \rangle N)$ for the subcritical (triangles), critical (circles) and supercritical (squares) cases, respectively. Global probabilities $u(\langle k \rangle)$ and $d(\langle k \rangle)$ were as in the example in the main text, with $n = 10$ and $\kappa_m = 20$. $N = 1000$.

total energy consumption, respectively. Under a mean-field approximation, the situation can be reduced to a nonlinear preferential rewiring model, similar to the one studied in Ref. [14] but more general. We derive analytical expressions which can be compared to experimental data. In particular, our results are in good quantitative agreement with results for synaptic pruning. Furthermore, they show how scale-free stationary degree distributions can be obtained with biologically inspired mechanisms. To the best of our knowledge, this is the first attempt to model emergent topological properties of the brain from this kind of microscopic considerations.

Acknowledgements. We wish to thank V.M. Eguíluz for comments concerning the experimental data and S. de Franciscis for technical advice. This work was financed by Junta de Andalucía project FQM-01505 and by MEC-FEDER project FIS2005-00791.

References

1. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79, 2554–2558 (1982)
2. Amit, D.J.: Modeling Brain Function. Cambridge Univ. Press, Cambridge (1989)

3. Mejias, J.F., Torres, J.J.: The role of synaptic facilitation in coincidence spike detection. J. Comp. Neurosci. 24, 222–234 (2008)
4. Johnson, S., Marro, J., Torres, J.J.: Functional optimization in complex excitable networks. EPL 83, 46006 (2008)
5. Huttenlocher, P.R., Dabholkar, A.S.: Regional differences in synaptogenesis in human cerebral cortex. Joural of Comparative Neurology 387, 167–178 (1997)
6. De Roo, M., Klauser, P., Mendez, P., Poglia, L., Muller, D.: Activity-dependent PSD formation and stabilization of newly formed spines in hippocampal slice cultures. Cerebral Cortex 18, 151–161 (2008)
7. Klintsova, A.Y., Greenough, W.T.: Synaptic plasticity in cortical systems. Current Opinion in Neurobiology 9, 203–208 (1999)
8. Chechik, G., Meilijson, I., Ruppin, E.: Synaptic pruning in development: A computational account. Neural Comput. 10(7), 1759–1777 (1998)
9. Chechik, G., Meilijson, I., Ruppin, E.: Neuronal regulation: A mechanism for synaptic pruning during brain maturation. Neural Comp. 11(8), 2061–2080 (1999)
10. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U.: Complex networks: Structure and dynamics. Phys. Rep. 424, 175 (2006)
11. Hebb, D.O.: The organization of behavior. Wiley, New York
12. Mejias, J.F., Torres, J.J.: Maximum memory capacity on neural networks with short-term synaptic depression and facilitation. Neural Comp. 21(3), 851–871 (2009)
13. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
14. Johnson, S., Marro, J., Torres, J.J.: Nonlinear preferential rewiring in fixed-size networks as a diffusion process (submitted)
15. Johnson, S., Marro, J., Torres, J.J.: A nonlinear evolving network model, and its application to brain development (submitted)
16. Eguíluz, V.M., Chialvo, D.R., Cecchi, G.A., Baliki, M., Apkarian, A.V.: Scale-free brain functional networks. Phys. Rev. Lett. 94, 018102 (2005)
17. Zhou, C., Zemanova, L., Zamora, G., Hilgetag, C.C., Kurths, J.: Hierarchical organization unveiled by functional connectivity in complex brain networks. Phys. Rev. Lett. 97, 238103 (2006)
18. Kaiser, M., Martin, R., Andras, P., Young, M.P.: Simulation of robustness against lesions of cortical networks. Eur. J. Neurosci. 25 (2007)

Fuzzy Logic, Soft Computing, and Applications*

Inma P. Cabrera, Pablo Cordero, and Manuel Ojeda-Aciego

Dept. Matemática Aplicada, Univ. de Málaga. Spain
{ipcabrera,pcordero,aciego}@uma.es

Abstract. We survey on the theoretical and practical developments of the theory of fuzzy logic and soft computing. Specifically, we briefly review the history and main milestones of fuzzy logic (in the wide sense), the more recent development of soft computing, and finalise by presenting a panoramic view of applications: from the most abstract to the most practical ones.

1 Fuzzy Logic

Although there are authors that establish the origin of *Fuzzy Logic* in the introduction of multiple-valued logics by Jan Łukasiewicz in 1920, it is commonly accepted that it emerged from the theory of fuzzy sets introduced by Lotfi Zadeh in [49]. In this paper, fuzzy subsets were used as a formalization of vagueness (in the sense of a predicate that applies to a certain degree, not in absolute terms), with the underlying notion of soft membership, in which objects might neither belong nor not belong to a set and that there may be borderline cases.

Joseph Goguen also participated in the initial developments of fuzzy logic with his "logic of inexact concepts." Since then, the theory of fuzzy logic has been constantly growing and, moreover, has been applied to very diverse fields from control theory to artificial intelligence. Although statisticians tend to prefer probabilistic logic, and control engineers prefer traditional two valued logic, nowadays, it cannot be discussed that fuzziness has become a mainstream theory and, as we will notice in next section, a mainstay of *Soft Computing.*

Now, two important question arise: what is really fuzzy logic? and what is not fuzzy logic? Related to these questions, Zadeh recently [52] wrote

> Fuzzy logic is not fuzzy. Basically, fuzzy logic is a precise logic of imprecision and approximate reasoning [...] Paradoxically, one of the principal contributions of fuzzy logic is its high power of precisiation of what is imprecise. This capability of fuzzy logic suggests that it may find important applications in the realms of economics, linguistics, law and other human-centric fields.

A typical source of confusion is the duality of meaning of the term *fuzzy logic.* In a wide sense, it is much more than a logical system, and different branches can

---

* Partially supported by projects TIN2006-15455-C03-01, TIN2007-65819 (Science Ministry of Spain) and P06-FQM-02049 (Junta de Andalucía).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 236–244, 2009.
© Springer-Verlag Berlin Heidelberg 2009

be observed: (i) Logical systems aiming at formalizing approximate reasoning, that is *fuzzy logic in narrow sense*, mathematical fuzzy logic systems such as those in the monographs [5,16,18,39] have become standard in this research line; (ii) *General theories of fuzzy sets* such as L-fuzzy sets, bipolar fuzzy sets, intuitionistic fuzzy sets, research on t-norms and/or copulas as adequate extensions of the intersection of fuzzy sets; (iii) The *epistemic approach*, knowledge representation and reasoning using fuzzy logic, natural language, information systems, fuzzy databases, etc. [11]; (iv) The *relational approach*, or the study of fuzzy relations and, more generally, fuzzy dependencies [48,2]. This aspect focuses on the representation and manipulation of imprecisely defined functions and relations. It is this facet of fuzzy logic that plays a pivotal role in its applications to system analysis and control.

Several notions in crisp mathematics can be translated into the corresponding notions in fuzzy mathematics in a uniform way by Zadeh Extension Principle [50]. A natural question is: Given a crisp deduction rule, does there exist a canonical way to extend it in a fuzzy context? Deduction in fuzzy logic is governed by a collection of rules of deduction which, in the main, are rules that govern propagation and counterpropagation of imprecise constraints. The principal rule is the compositional rule of inference [14]. Concerning reasoning and knowledge representation, it is worth to refer to the several generalizations of logic programming to a fuzzy framework, such as [17,36,35,7,27,34].

2 Soft Computing

Most of the attempts for defining the evolving term *Soft Computing* coincide in that it is a collection of techniques which uses the human mind as a model and aims at formalizing our cognitive processes. These methods are meant to operate in an environment that is subject to uncertainty and imprecision. The objective is to study, model and analyse complex phenomena for which more conventional methods have not yielded low cost, analytic, and complete solutions. According to Zadeh [51] the guiding principle of soft computing is to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost. The notion soft computing is to computational intelligence as traditional hard computing is to artificial intelligence, and usually is viewed as a foundation component for the emerging field of conceptual intelligence.

Soft computing replaces the traditional time-consuming and complex techniques of hard computing with more intelligent processing techniques. The key aspect for moving from hard to soft computing is the observation that the computational effort required by conventional approaches which makes in many cases the problem almost infeasible, is a cost paid to gain a precision that in many applications is not really needed or, at least, can be relaxed without a significant effect on the solution. A basic difference between perceptions and measurements is that, in general, measurements are crisp whereas perceptions are fuzzy.

According to scientific folklore, the name Computational Intelligence is chosen to indicate the link to and the difference with Artificial Intelligence. Artificial Intelligence techniques are top-down whereas Computational Intelligence techniques are generally bottom-up, with order and structure emerging from an unstructured beginning.

It is widely accepted that the main components of Soft Computing are Fuzzy Logic, Probabilistic Reasoning, Neural Computing and Genetic Algorithms. This four constituents share common features and they are considered complementary instead of competitive. The mentioned technologies can be combined in models which exploit their best characteristics. As an important consequence, some real problems can be solved most effectively by using hybrid systems what is increasing the interest on them. The first and probably the most successful hybrid approach till now are the so-called neurofuzzy systems [43], although some other hybridations are being developed with great success as, for instance, the genetic fuzzy systems [6].

In the partnership of the mentioned collection of computational techniques that compose soft computing, fuzzy logic, which has been widely referred in previous section, is concerned with imprecision, approximate reasoning and representation of aspects that are only qualitatively known; Probabilistic Reasoning such as Bayesian Belief Networks, with uncertainty and belief propagation; the main characteristic is its ability to update previous outcome estimates by conditioning them with newly available evidence [41]; Neural Computing focuses on the understanding of neural networks and learning systems, self-organising structures, and the implementation of models from available data [44,15]; last but not least, Genetic and Evolutionary Computing [19,45,13] provide approaches to computing based on analogues of natural selection, such as, the optimisation methods based on ant colonies [10,46] or on particle swarms [25,42].

3 Applications

The new wave of theoretical contributions and practical applications that followed the seminal works by Zadeh has had a remarkable inspirational effect on numerous disciplines. Activities in soft computing have increased since the field started. They do not only focus on theoretical descriptions, but also provide a collection of real-world problems and techniques that are used to solve them.

Industry has benefited from adopting these techniques to address a variety of problems that can be seen also by the diverse range of products developed. Lately, it has been noticed that publications tend to combine the different subfields which seems to indicate that there are much more applications to come.

Last but not least, it is important to note the number of books and special issues of journal dedicated to applications of fuzzy logic and soft computing that have been recently published [3,28,26,20,38,22,47,8,12,4,9,1,30,21].

The applications range from the purely theoretical ones, those which develop new lines in abstract mathematics or logic, passing across the areas of multimedia, preference modelling, information retrieval, hybrid intelligent systems, image processing, etc., to practical applications domains such as robotics and manufacturing, actuarial science, nuclear or medical engineering.

Pure and Applied Mathematics. Theoretical foundations of soft computing techniques stem from purely mathematical concepts. The basic mathematical formalisms of fuzzy logic and soft computing have triggered a renewed interest in some old theories, such as that of residuated lattices or the theory of t-norms and copulas, and have initiated a complete redesign of well-established areas such as the theory of differential equations (with the addition of fuzziness), topology (including similarity spaces, tolerance spaces, approximation spaces), development and algebraic study of new logical systems for dealing with vagueness, imprecision and uncertainty, etc [33].

Extended tools for fuzzy and similarity-based reasoning. Existing tools for knowledge representation and reasoning, such as Prolog-based implementations, are being extended to the framework of fuzzy logic or, even, lattice-valued logics. In this sense, we can cite the works [37,24]. Some other approaches also include the adaptation of enhancements and specific optimization methods, such as the tabulation (or tabling) methods for logic programming.

Case-based reasoning. This model of reasoning incorporates problem solving, understanding and learning, and integrates all of them with memory processes. It involves adapting old solutions to meet new demands, using old cases to explain new situations or to justify new solutions, and reasoning from precedents to interpret a new situation. Recent research is demonstrating the role of soft computing tools, both individually and in combination, for performing different tasks of case based reasoning with real life applications [32].

Multimedia processing. Due to their strong learning and cognitive ability, soft computing techniques have found applications in multimedia processing and, nowadays, there is a wide range of research areas of soft computing in multimedia processing including video sequence, color quantization, image retrieval, meeting video, document image analysis, image segmentation and biometric application. The increased possibilities to capture and analyze images have contributed to create the new scientific field of image processing that has numerous commercial, scientific, industrial and military applications [29].

Preference modelling and decision making. Although standard approaches to decision-making problems assumed by default that all the information is expressed in the same preference representation format, in real practice this is hardly possible. As a result, new fuzzy approaches to integrating different preference representation formats in decision-making are of great importance. Moreover, missing information poses additional difficulties that have to be addressed when dealing with real decision-making problems, which leads to topics that are naturally included within the boundaries of fuzzy logic and soft computing. In this respect, theoretical studies on areas such as extensions of fuzzy sets (type-2 fuzzy sets, L-fuzzy sets, interval-valued fuzzy sets, fuzzy rough sets) or aggregation operators (fuzzy measures, linguistic aggregators, inter-valued aggregators) are specially useful. Some specific application domains of preferences modelling are the following: data-base theory, classification and data mining, information retrieval, non-monotonic reasoning, recommendation systems, etc [23].

Knowledge engineering applications. With the advent of artificial intelligence, the emphasis on knowledge engineering moved from social and philosophical concepts to the problem of knowledge representation in computers. The inherent synergy of the different methods of soft computing allows to incorporate human knowledge effectively, deal with imprecision and uncertainty, and learn to adapt to unknown or changing environments for better performance. One can see applications to several areas related to management of knowledge, such as knowledge representation, knowledge acquisition, knowledge-based inference, modeling and developing knowledge-based systems, knowledge integration, and knowledge discovery.

Ontologies and the semantic web. When analysing information on the web one has to note the difference between information produced primarily for human consumption and that produced mainly for machines; on the other hand, one has to keep track of information uncertainty. The increasing interest in ontology-based, standard representations of belief-based, possibilistic and probabilistic information, as well as other types of uncertainty, is bringing soft computing techniques for uncertainty representation and processing to the forefront of semantic web research. In the last few years, a number of seminal workshops and seminars have spread the interest for these issues within both the Semantic Web and the fuzzy logic or soft computing communities. Fuzzy logic has been used to bridge the gap among intuitive knowledge and machine-readable knowledge systems. Much research is also being done on techniques for extracting incomplete, partial or uncertain knowledge, as well as on handling uncertainty when representing extracted information using ontologies, e.g. to achieve semantic interoperability among heterogeneous systems. Semantic Web demands the management of large amounts of fuzzy data and the extraction of fuzzy information. Therefore, automatic tools for reasoning about fuzzy dependencies are necessary, in this line we can cite [40].

Business and economics. Soft computing methods can be used in an uncertain economic decision environment to deal with the vagueness of human thought and the difficulties in estimating inputs. There is a plethora of applications of soft computing in business and economics, which range from marketing (analysis of customer's purchasing attitudes, fraud detection, service quality), to finance (stock market predicting schemes, portfolio selection, risk management, loan assessment systems), electronic business (e-commerce decisions, personalization, risk analysis in e-commerce), etc.

Medical engineering. Successful diagnoses and surgical outcomes depend on the experience and skill of examiners and surgeons, but dependence on the subjective abilities of these healthcare professionals carries with it the risk of failure. Teaching these feelings to beginners is a very difficult task, because the skill of diagnose the feelings is based on subjective evaluation. Thus, the medical industry requires new engineering technologies, such as soft computing techniques, to assess information objectively. While recent developments in medical engineering have been achieved by state-of-the-art of intelligent computing techniques, including computer-aided diagnosis, computer-aided radiography, computer-assisted surgery, developments in soft computing, including information processing, signal/image processing, and data mining seems to be specially promising in this field.

Information retrieval. Information retrieval aims at defining systems able to provide a fast and effective content-based access to a large amount of stored information. Currently, soft computing techniques are being used to model subjectivity and partiality in order to provide a adaptative environment of information retrieval, one which learns the user's concept of relevance. The modelling is performed by the knowledge representation components of SC such as fuzzy logic, probabilistic reasoning, and rough sets. This way, the application of soft computing techniques can be of help to obtain greater flexibility in IR systems.

Fuzzy control applications. The first application of fuzzy logic to control systems was the design of a fuzzy algorithm for regulating a steam engine by given Mamdani and Assilian [31]. After this starting point, the research and applications of fuzzy control progressed rapidly. Hard computing methodologies are not useful for the construction of the robot control systems of acceptable cost, it is the use of soft computing techniques what allows to overcome the problem of complexity of control systems and, in addition, provides them with abilities of tolerance for imprecise data, and high efficiency and performance.

Robotics. This field has a number of subareas which can profit from soft computing techniques. For instance, the drive control of a robot is often performed by a neuro-fuzzy system that generates action commands to the motors, the input of this systems comes from the surrounding information, in terms of data obtained by the vision subsystem and the goal identifying device. Then, fuzzy inference mechanisms are usually provided by neural networks. Moreover, the systems are taught how to behave by means of adjusting its knowledge base by a neural network learning technique.

4 Conclusions

We have briefly recalled the basics of fuzzy logic and soft computing, and have surveyed a range of applications of these fields ranging from the purely theoretical to the most practical ones.

References

1. Aliev, R.A., Fazlollahi, B., Aliev, R.R.: Soft computing and its applications in business and economics. Springer, Heidelberg (2004)
2. Belohlávek, R., Vychodil, V.: Attribute implications in a fuzzy setting. In: Missaoui, R., Schmidt, J. (eds.) Formal Concept Analysis. LNCS, vol. 3874, pp. 45–60. Springer, Heidelberg (2006)

3. Castillo, O., Melin, P., Kacprzyk, J., Pedrycz, W. (eds.): Soft Computing for Hybrid Intelligent Systems. Studies in Computational Intelligence, vol. 154. Springer, Heidelberg (2008)
4. Chiclana, F., Herrera-Viedma, E., Alonso, S., Herrera, F.: Special issue on fuzzy approaches in preference modelling, decision making and applications. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 16(suppl. 2) (2008)
5. Cignoli, R., D'Ottaviano, I.M., Mundici, D.: Algebraic Foundations of Many-Valued Reasoning. Kluwer, Dordrecht (1999)
6. Cordón, O., Alcalá, R., Alcalá-Fernandez, J., Rojas, I.: Genetic fuzzy systems: What's next? an introduction to the special section. IEEE Trans. Fuzzy Systems 15(4), 533–535 (2007)
7. Damásio, C.V., Pereira, L.M.: Hybrid probabilistic logic programs as residuated logic programs. Studia Logica 72(1), 113–138 (2002)
8. Dillon, T.S., Shiu, S.C., Pal, S.K.: Soft computing in case based reasoning. Applied Intelligence 21(3), 231–232 (2004)
9. Ding, L.: A New Paradigm of Knowledge Engineering by Soft Computing. World Scientific, Singapore (2001)
10. Dorigo, M., Gambardella, L.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)
11. Dubois, D., Prade, H.: Epistemic entrenchment and possibilistic logic. Artificial Intelligence 50(2), 223–239 (1991)
12. Egiazarian, K., Hassanien, A.E.: Special issue on soft computing in multimedia processing. Informatica 29 (2005)
13. Fernández, A., del Jesús, M.J., Herrera, F.: Improving the performance of fuzzy rule based classification systems for highly imbalanced data-sets using an evolutionary adaptive inference system. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 462–465. Springer, Heidelberg (2009)
14. Fullér, R., Zimmermann, H.J.: On computation of the compositional rule of inference under triangular norms. Fuzzy Sets and Systems 51(3), 267–275 (1992)
15. Galushkin, A.I.: Neural Networks Theory. Springer, Heidelberg (2007)
16. Gottwald, S.: Fuzzy Sets and Fuzzy Logic: The foundations of application from a Mathematical Point of View. Vieweg-Verlag (1993)
17. Guadarrama, S., Muñoz, S., Vaucheret, C.: Fuzzy prolog: a new approach using soft constraints propagation. Fuzzy Sets and Systems 144(1), 127–150 (2004)
18. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
19. Herrera, F., Verdegay, J.L. (eds.): Genetic Algorithms and Soft Computing. Physica-Verlag (1996)
20. Herrera-Viedma, E., Cordón, O.: Special issue on soft computing applications to intelligent information retrieval on the internet. International Journal of Approximate Reasoning 34(2), 89–95 (2003)
21. Hou, Z.-G., Polycarpou, M.M., He, H.: Special issue on neural networks for pattern recognition and data mining. Soft Computing 12(7) (2008)
22. Jamshidi, M.: Special issue on neural networks and fuzzy logic: theory and applications in robotics and manufacturing. Comput. Electr. Eng. 19(4) (1993)
23. Julián, P., Moreno, G., Penabad, J.: On the declarative semantics of multi-adjoint logic programs. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 253–260. Springer, Heidelberg (2009)

24. Julián, P., Rubio-Manzano, C.: A similarity-based wam for bousi-prolog. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 245–252. Springer, Heidelberg (2009)
25. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
26. Kercel, S.W.: Industrial applications of soft computing. IEEE Transactions on Systems, Man and Cybernetics 36(4), 450–452 (2006)
27. Krajči, S., Lencses, R., Vojtáš, P.: A comparison of fuzzy and annotated logic programming. Fuzzy Sets and Systems 144(1), 173–192 (2004)
28. Langari, R., Yen, J.: Fuzzy logic at the turn of the millennium. IEEE Transactions on Fuzzy Systems 9(4), 481–482 (2001)
29. Lopez-Molina, C., Barrenechea, E., Bustince, H., Couto, P., Baets, B.D., Fernández, J.: Edges detection based on gravitational forces. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 1029–1036. Springer, Heidelberg (2009)
30. Ma, Z. (ed.): Soft Computing in Ontologies and Semantic Web. Springer, Heidelberg (2006)
31. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man-Machine Studies 7, 1–13 (1975)
32. Martins-Bede, F.T., Godo, L., Sandri, S., Freitas, C.C., Dutra, L.V., Guimarães, R.J.P.S., Amaral, R.S., Carvalho, O.S.: Classification of schistosomiasis prevalence using fuzzy case-based reasoning. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 1053–1060. Springer, Heidelberg (2009)
33. Medina, J.: Overcoming non-commutativity in multi-adjoint concept lattices. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 278–285. Springer, Heidelberg (2009)
34. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: Multi-adjoint logic programming with continuous semantics. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, pp. 351–364. Springer, Heidelberg (2001)
35. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: A procedural semantics for multi-adjoint logic programming. In: Brazdil, P.B., Jorge, A.M. (eds.) EPIA 2001. LNCS (LNAI), vol. 2258, pp. 290–297. Springer, Heidelberg (2001)
36. Mukaidono, M., Shen, Z., Ding, L.: Fundamentals of fuzzy Prolog. International Journal of Approximate Reasoning 3(2), 179–193 (1989)
37. Munoz-Hernandez, S., Pablos Ceruelo, V., Strass, H.: Rfuzzy: an expressive simple fuzzy compiler. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 270–277. Springer, Heidelberg (2009)
38. Nachtegael, M., Kerre, E., Damas, S., der Weken, D.V.: Special issue on recent advances in soft computing in image processing. International Journal of Approximate Reasoning 50(1), 1–2 (2009)
39. Novák, V., Perfilieva, I., Močkoř, J.: Mathematical Principles of Fuzzy Logic. Kluwer, Dordrecht (2000)
40. Cordero, P., Enciso, M., Mora, A., de Guzmán, I.: A complete logic for fuzzy functional dependencies over domains with similarity relations. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 261–269. Springer, Heidelberg (2009)
41. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
42. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm Intelligence 1(1), 33–57 (2007)

43. Prasad, B. (ed.): Introduction to Neuro-Fuzzy Systems. Advances in Soft Computing Series, vol. 226. Springer, Heidelberg (2000)
44. Rutkowski, L., Kacprzyk, J. (eds.): Neural networks and soft computing. Springer, Heidelberg (2003)
45. Sanchez, E., Shibata, T., Zadeh, L.A. (eds.): Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives. World Scientific, Singapore (1997)
46. Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L.: Ant-based load balancing in telecommunication networks. Adaptive Behaviour 5(2), 169–207 (1997)
47. Torra, V., Narukawa, Y.: Special issue on soft computing methods in artificial intelligence. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 16(suppl. 1) (2008)
48. Yager, R.R., Zadeh, L.A. (eds.): An Introduction to Fuzzy Logic Applications in Intelligent Systems. Kluwer Academic Publishers, Dordrecht (1992)
49. Zadeh, L.A.: Fuzzy sets. Information Control 8, 338–353 (1965)
50. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning (parts I, II, III). Information Sciences 8, 199–249, 8, 301–357, 9, 43–80 (1975)
51. Zadeh, L.A.: Soft computing and fuzzy logic. Software, IEEE 11(6), 48–56 (1994)
52. Zadeh, L.A.: Is there a need for fuzzy logic? Information Sciences 178(13), 2751–2779 (2008)

A Similarity-Based WAM for Bousi∼Prolog*

Pascual Julián-Iranzo and Clemente Rubio-Manzano

Department of Information Technologies and Systems,
University of Castilla-La Mancha
{Pascual.Julian}@uclm.es, {Clemente.Rubio}@alu.uclm.es

Abstract. Bousi∼Prolog is an extension of the standard Prolog language with an operational semantics which is an adaptation of the SLD resolution principle where classical unification has been replaced by a fuzzy unification algorithm based on proximity relations defined on a syntactic domain. In this paper we present the structure and main features of a low level implementation for Bousi∼Prolog. It consists in a compiler and an extension of the **W**arren **A**bstract **M**achine (WAM) able to incorporate fuzzy unification.

Keywords: Fuzzy Logic Programming, Fuzzy Prolog, Unification by Similarity, Warren Abstract Machine.

1 Introduction

The programming language we call Bousi∼Prolog (BPL for short) [4] is an extension of the standard Prolog language with an operational semantics which is a variant of the SLD resolution procedure where the classical unification algorithm has been replaced by a weak unification algorithm based on proximity relations (fuzzy relations that fulfill the reflexive and symmetric properties). Hence, the operational mechanism is a generalization of the similarity-based SLD resolution principle [9]. Informally, this weak unification algorithm states that two terms $f(t_1, \ldots, t_n)$ and $g(s_1, \ldots, s_n)$ weakly unify if the root symbols f and g are approximate and each of their arguments t_i and s_i weakly unify. Therefore, the weak unification algorithm does not produce a failure when there is a clash of two syntactical distinct symbols, whenever they are approximate, but a success with a certain approximation degree. Hence, Bousi∼Prolog computes answers as well as approximation degrees.

The syntax of BPL is mainly the Prolog syntax but enriched with a built-in symbol "∼" used for describing proximity/similarity relations by means of *similarity equations* of the form: `<symbol> ~ <symbol> = <degree>`. Although, a similarity equation represents an arbitrary fuzzy binary relation, its intuitive reading is that two constants, n-ary function symbols or n-ary predicate symbols

---

* This work has been partially supported by FEDER and the Spanish Science and Education Ministry under grants TIN 2004-07943-C04-03 and TIN 2007-65749 and by the Castilla-La Mancha Regional Administration under grant PII1I09-0117-4481.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 245–252, 2009.
© Springer-Verlag Berlin Heidelberg 2009

are approximate or similar with a certain degree. That is, a similarity equation $a \sim b = \alpha$ can be understood in both directions: a is approximate/similar to b and b is approximate/similar to a with degree α. Therefore, a Bousi∼Prolog program is a sequence of Prolog facts and rules followed by a sequence of similarity equations.

Example 1. Suppose a fragment of a database that stores a semantic network with information about people's names and hair color, as well as the approximate relation between black, brown and blond hair[1].

```
% BPL directive
:- transitivity(no).

% FACTS                                                    % SIMILARITY EQUATIONS
is_a(john, person).     hair_color(john,black).              black~brown=0.6.
is_a(peter, person).    hair_color(peter,brown).             black~blond=0.3.
is_a(mary, person).     hair_color(mary,blond).              blond~brown=0.6.
```

In a standard Prolog system, if we ask about whether `peter`'s hair is `blond`, "`?- hair_color(peter, blond)`", the system fails. However BPL allows us to obtain the answer "`Yes with 0.6`".

To obtain this answer, the BPL system operates as follows: i) First it generates the reflexive, symmetric closure of the fuzzy relation $\{\mathcal{R}(\mathit{black}, \mathit{brown}) = 0.6,\ \mathcal{R}(\mathit{black}, \mathit{blond}) = 0.3,\ \mathcal{R}(\mathit{blond}, \mathit{brown}) = 0.6\}$, constructing a proximity relation. ii) Then it tries to unify the goal `hair_color(peter, blond)` and (eventually) the fact `hair_color(peter,brown)`. Because there exists the entry $\mathcal{R}(\mathit{brown}, \mathit{blond}) = 0.6$ in the constructed proximity relation (that is, `brown` is approximate to `blond`), the unification process succeeds with approximation degree 0.6.

The above example serves to illustrate both the syntax and the operational semantics of the language. Also, it is important to note that, in the last example, to inhibit the construction of the transitive closure (and therefore the construction of a similarity relation) has been crucial to model the information properly and to obtain a convenient result. This effect is obtained by the BPL directive `transitivity` which disables or enables the construction of the transitive closure of a fuzzy relation during the compilation process[2]. If a similarity relation would be generated, the system would constructed the entries $\mathcal{R}(\mathit{brown}, \mathit{blond}) = 0.3$ and $\mathcal{R}(\mathit{blond}, \mathit{brown}) = 0.3$, overlapping the initial approximation degree provided by the user and leading to a wrong information modeling[3].

As it was shown in [4], there are exist several practical applications for a language with the aforementioned characteristics: flexible query answering;

---

[1] For the sake of simplicity we only consider programs without variables in the examples. Of course, BPL permits programs ans goals containing variables.

[2] In the BPL system, `transitivity(yes)` is the current default option.

[3] Because a person with brown hair should be closer to a person with blond hair than a person with black hair is to a person with blond hair.

advanced pattern matching; information retrieval where textual information is selected or analyzed using an ontology; text cataloging and analysis; etc.

Bousi∼Prolog is publicly available and can be found at the URL address `http://www.inf-cr.uclm.es/www/pjulian/bousi.html`. Currently it is delivered in two implementations formats: a high level and a low level implementation. The high level implementation [4] is written in Prolog through a meta-interpreter. Hence it may present efficiency problems. In order to solve them, we have investigated how to incorporate the weak unification algorithm into the Warren Abstract Machine. Some preliminary results for a pure subset of the Prolog syntax augmented with similarity equations can be find in [6]. In this paper, carrying this line of work one step further, we present a low level implementation of the Bousi∼Prolog language, which consists of an architecture compounded by a compiler and an enlargement of the Warren Abstract Machine able to execute BPL programs efficiently. In the following sections we develop this architecture.

2 Related Work

During the last decade, several fuzzy extensions of the resolution rule relying in the replacement of the classical syntactic unification algorithm by a similarity-based unification algorithm, have been proposed. We can distinguish two main lines of research: The first one is represented by the theoretical work [3], whose main practical realization is the fuzzy logic language LIKELOG [2] (an interpreter implemented in Prolog using rather direct techniques and cumbersome concepts). The second line of research is the closest to ours. It is based on [9], where the concepts of weak unification and weak SLD resolution were developed. In [7], a similarity-based logic programming language, named SiLog, is presented. SiLog is an interpreter written in Java.

Neither LIKELOG nor SiLog are publicly available and therefore a practical comparison is impossible. However, we can describe the following important features of the Bousi∼Prolog language that distinguish it from these other proposals:

1. Automatic support for the user. The user is responsible of providing a subset of similarity equations, defining a fuzzy relation, but the system will automatically generate its reflexive, symmetric and/or transitive closure, in order to obtain a proximity or a similarity relation. In the case of the construction of a similarity this feature gives a substantial support for the programmer, since it is not easy to define a similarity relation due to the transitivity constrains, which may contradict the initial similarity values.
2. A more flexible operational mechanism and additional expressive power. Since, Bousi∼Prolog uses proximity relations (as well as similarity relations), its operational semantics is more general and flexible than the ones used by LIKELOG and SiLog which are exclusively based on similarity relations. The removal of the transitivity restriction also provides programmers with more freedom to express data structures and model information [10]. Furthermore, as it was commented in [10] and Example 1 confirms, similarity relations sometimes represent fuzzy information incorrectly. Hence, to allow the use of

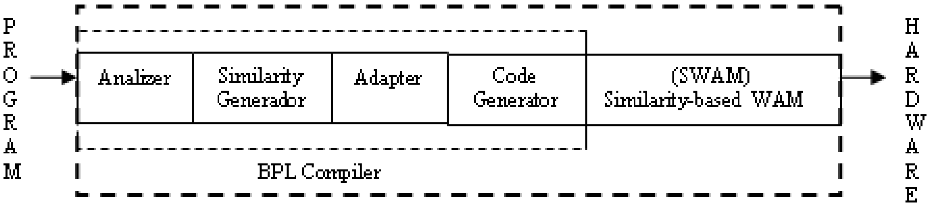
proximity relations, not only increases the expressive power of the language, but it is critical in order to solve certain problems.

3. A true extension of Prolog. Bousi∼Prolog is a true Prolog extension and not a simple interpreter able to execute the weak SLD resolution principle. It covers the main features of the Prolog language, such as, arithmetic, lists or the cut operator "!".

3 Bousi∼Prolog Architecture

Standard implementations of the Prolog programming language consists of a compiler, which transforms Prolog code into object code, and a **W**arren **A**bstract **M**achine (WAM) [11] that runs it. This is the *de facto* standard architecture for building Prolog systems. The WAM is a virtual computer that aids in the compilation and implementation of Prolog and offers techniques for compiling symbolic languages that can be generalized beyond Prolog. A tutorial reconstruction for the WAM can be found in [1].

Because we want to construct a system comparable to other Prolog systems in efficiency and robustness, we have chosen to develop this kind of low level implementation. Therefore, the BPL architecture will be an extension of these two components (the compiler and the WAM). In what follows, we will call **S**imilarity-based **WAM** (SWAM), the extension of the WAM able of handling proximity or similarity relations[4].



BPL Arquitecture

The BPL architecture is a multi-layer architecture (with three layers: GUI, compiler, abstract machine), consisting of over 6500 code lines, divided into 27 classes. It has been implemented in Java, since this is an object oriented language that possess facilities to deploy the BPL system on the web.

3.1 The Compiler

The compiler is responsible for translating BPL programs into SWAM code. To cope with this task, it is divided into four components having each one a different function.

---

[4] We maintain the denomination "Similarity-based Warren Abstract Machine" by historical reasons, although the current implementation allows the treatment of proximity relations.

The Analizer. It verifies that the programs are syntactically correct and obtains the syntactic tree which is the basis for later code generation.

The Similarity Generator. It calculates a proximity relation or a similarity relation (if the transitivity option is enabled) starting from the similarity equations provided by the programmer. For the construction of the similarity relation, it uses a three step algorithm. The first step computes the reflexive closure of the initial relation; the second the symmetric closure. The third step is an extension of well-know Warshall's algorithm for computing the transitive closure of a binary relation, where the classical meet and joint operators on the set 0,1 have been changed by the maximum (MAX) and the minimum (MIN) operators on the real interval [0,1], respectively. The algorithm is as follows:

```
for ( k = 0 ; k < nTotal ; k++ ) {
   for ( i = 0 ; i < nTotal ; i++ ) {
      for ( j = 0 ; j < nTotal ; j++ ) {
         dMatrix[i][j] = MAX ( dMatrix [i][j] ,
                         MIN ( dMatrix [i][k]  , dMatrix[k][j] ));}}}
```

See [5] to know about the formal properties of this algorithm.

The Adapter. It takes the syntactic tree and the proximity or similarity relation and constructs an intermediate representation which is used by the Code Generator to obtain the object code. Conceptually it defines a program transformation. The next paragraph summarizes the main ideas involved in the design of the Adapter.

In our context, a clause defining the predicate p can also be considered as defining each predicate q which is approximate or similar to p. On the other hand, the structure of the WAM is designed to test a “crisp” matching of predicate symbols. Therefore, if we want a “flexible” matching of predicate symbols without forcing the structure of the WAM, given a clause defining a predicate p, we need to introduce a new clause for each predicate q which is similar to p. We do it in order to simulate a “flexible” matching with a “crisp” technique.

To help in the specification of the program transformation performed by the Adapter, we need to introduce an extended language obtained by adding to the object language alphabet the elements of the lattice $[0, 1]$. Clauses in this extended language contain bodies with literals which are approximation degrees. Then, given a program $\mathcal{R}$ and a proximity or similarity relation on the syntactic domain generated by Π, the transformed program $\Pi' = \{q(t_1, \ldots, t_n) \leftarrow \alpha, \mathcal{Q} \mid (p(t_1, \ldots, t_n) \leftarrow \mathcal{Q}) \in \Pi$ and $\mathcal{R}(p, q) = \alpha > 0\}$.

We reproduce the effect of this transformation in the following example.

Example 2. Program `P'` is the intermediate code (a set of extended clauses) for program `P`.

```
%%PROGRAM  P              %%PROGRAM  P'(e-clauses)
young~middle=0.8.         young(michael):-1.0.
                          young(robbie):-0.8.
young(michael).           midle(robbie):-1.0.
middle(robbie).           middle(michael):-0.8
```

The clauses of `P'` are obtained when the above transformation is applied by the Adapter to all clauses of the original program `P`. For instance, the first clause "`young(michael)`" is transformed into "`young(michael):-1.0`" and "`middle(michael):-0.8`", because there exist the entries $\mathcal{R}(young, young) = 1$ and $\mathcal{R}(young, middle) = 0.8$ in the proximity relation $\mathcal{R}$ (generated starting from the similarity equations).

The Code Generator. The compilation of the intermediate code of the transformed program to machine code is done using standard techniques. In essence, clauses of a transformed program are translated into the same machine instruction set a standard implementation would have produced. The only difference is that similarity degrees, in the body of transformed clauses, are "stored" in a multi-label field of the `try_me_else`, `retry_me_else` and `trust_me` machine instructions. The values in the multi-label field will be used during the computation of the unification degree in a weak SDL resolution step.

We illustrate the code generation phase by means of following example.

Example 3. Object code for the program `P` of Example 2. The starting point in the code generation phase is the program `P'`. Each machine instruction consists in one or more memory words that may contain a possibly empty label field and an operation code followed by operands.

```
 0 : young   1.0  : try_me_else young#2       %% young(michael):-1.0.
 1 :              : get_constant michael, A0
 2 :              : proceed
 3 : middle 1.0   : try_me_else middle#2      %% middle(robbie):-1.0.
 4 :              : get_constant robbie, A0
 5 :              : proceed
 6 : middle#2 0.8 : trust_me
 7 :              : get_constant michael, A0   %% middle(michael):-0.8.
 8 :              : proceed
 9 : young#2 0.8  : trust_me
10 :              : get_constant robbie, A0    %% young(robbie):-0.8.
11 :              : proceed
```

3.2 Similarity-Based Warren Abstract Machine

The Similarity-based WAM is an extension of the WAM that allows the execution of BPL programs. It modifies the two parts of which the original machine consists: *the memory layout* and *the instruction set.*

Extension of the memory layout. The main changes are related to the incorporation of data structures that allow us to manage proximity or similarity relations. We add the so called *similarity matrix area*, which stores an adjacency matrix representation of a proximity or similarity relation. Two new specific registers: the *Aproximation Degree register* (`AD`), which stores the current computed approximation degree of a derivation; and the *Lambda-Cut register* (`LC`) which stores the lower bound for the approximation degree in a derivation, if the approximation degree computed in a step of a derivation is less than the value stored in the `LC` register, the derivation is cut[5] and is treated as a failing derivation. Finally, we need to modify the standard choice point frame[6] structure by adding a new field, `D`, to save the value stored in the `AD` register, prior to the creation of a choice point; this is because, when the computation backtracks and the next clause in an alternative is taken, we need to restart the computation (of the approximation degree) at the point it was left before the former clause was try.

Extension of the instruction set. As the choice point frame structure has been modified, the machine instructions that work in combination with it need also to be modified. That is, we have to modify the instructions: `try_me_else`; `retry_me_else`; `trust_me` and `allocate`. The first three machine instructions essentially update the value of the `AD` register (currently, this is done as the minimum of its current value and the similarity degree "stored" at the multilabel field of the instruction). The procedure `backtrack` helps to recover the old approximation degree value after a failure.

On the other hand, also it is necessary to modify some machine instructions involved in the process of argument unification, such as: `get_structure` and `get_constant`, which control the similarity of constant and function symbols of the terms in predicate arguments; `unify_value` and `unify_variable`, which play an special role in the implementation of a distributed fuzzy unification algorithm used by Bousi∼Prolog. Therefore, these instructions carry out the computation of the approximation degree after each step of the argument unification sequence.

Bousi∼Prolog implements a weak unification operator, denoted by "∼", which is the fuzzy counterpart of the syntactical unification operator "=" of standard Prolog. It can be used, in the source language, to construct expressions like "`Term1 ~ Term2 =:= Degree`" which is interpreted as follows: The expression is true if `Term1` and `Term2` are unifiable by proximity/similarity with approximation degree `AD` equal to `Degree`. In general, we can construct expressions "`Term1 ~ Term2 <op> Degree`" where "`<op>`" is a comparison arithmetic operator (that is, an operator in the set `{=:=, =\=, >, <, >=, =<}`). Also it is possible the following construction: `Term1 ~ Term2 = Degree` which succeeds if `Term1` and `Term2` are weak unifiable with approximation degree `Degree`; otherwise fails. These expressions may be introduced in a query as well as in the body of a clause, increasing the expresiveness of the language substantially. To implement this built-in operator, we must to add a set of new machine instructions: `unify_sim_greater`, `unify_sim_gEq`, `unify_sim_less`, `unify_sim_lEq`, and `unify_sim_eq`.

Finally, the machine instruction `unify_sim_cut` deals with the lambda-cut limitation.

---

[5] By default, the `LC` register is set to zero, i.e. no limitation is assumed. However, the `LC` register can be modified by the directive `cut(N)`, where `N` is a real number between zero and one.

[6] A choice point frame is a data structure used to save the computation state, that is, all the information required to continue the computation upon backtracking.

4 Conclusions and Future Work

Recently, in [6], we built a prototype low level implementation for a pure subset of Prolog with similarity relations. In this paper, following this initial path, we present a low level implementation for Bousi∼Prolog consisting in a compiler and a similarity-based WAM. The new implementation covers the main features of Prolog (arithmetic, lists, cut operator, etc.). Also it has a built-in proximity/similarity operator "∼", which increases the expressiveness of the BPL language. Moreover it offers an improved development environment and an improved graphical user interface. Additionally, Bousi∼Prolog incorporates an automatic treatment of proximity/similarity relations by means of a procedure able to generate the reflexive, symmetric and/or transitive closure of a fuzzy relation. Finally, this implementation improves the efficiency problems of a BPL high level implementation such as the one presented in [4].

As a matter of future work it is important to study the formal properties of proximity relations in combination with the SLD resolution principle. Also we want to study how to combine, in our setting, the weak resolution rule with a concrete instance of the multi-adjoint logic programming framework described in [8].

References

1. Aït-Kaci, H.: Warren's Abstract Machine: A Tutorial Reconstruction. MIT Press, Cambridge (1991)
2. Fontana, F.A., Formato, F.: Likelog: A logic programming language for flexible data retrieval. In: Proc. of the ACM SAC, pp. 260–267 (1999)
3. Fontana, F., Formato, F.: A similarity-based resolution rule. Int. J. Intell. Syst. 17(9), 853–872 (2002)
4. Julián, P., Rubio, C., Gallardo, J.: Bousi∼Prolog: a Prolog extension language for flexible query answering. In: ENTCS, p. 16. Elsevier, Amsterdam (in press, 2009)
5. Julián-Iranzo, P.: A procedure for the construction of a similarity relation. In: Málaga, U. (ed.) Proceedings of IPMU, pp. 489–496 (2008)
6. Julián, P., Rubio, C.: A WAM implementation for flexible query answering. In: Proc. of the 10th IASTED ASC, pp. 262–267. ACTA Press (2006)
7. Loia, V., Senatore, S., Sessa, M.: Similarity-based SLD resolution and its implementation in an extended prolog system. In: FUZZ-IEEE, pp. 650–653 (2001)
8. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: Similarity-based unification: a multi-adjoint approach. Fuzzy Sets and Systems 146(1), 43–62 (2004)
9. Sessa, M.I.: Approximate reasoning by similarity-based SLD resolution. Theoretical Computer Science 275(1-2), 389–426 (2002)
10. Shenoi, S., Melton, A.: Proximity relations in the fuzzy relational database model. Fuzzy Sets and Systems 100, 51–62 (1999)
11. Warren, D.H.D.: An Abstract Prolog Instruction Set. Technical note 309, SRI International, Menlo Park, CA (October 1983)

On the Declarative Semantics of Multi-Adjoint Logic Programs*

P. Julián[1], G. Moreno[2], and J. Penabad[3]

[1] Dep. Information Technologies and Systems, UCLM, 13071 Ciudad Real, Spain
[2] Dep. Computing Systems, UCLM, 02071 Albacete, Spain
[3] Dep. Mathematics, UCLM, 02071 Albacete, Spain
{Pascual.Julian,Gines.Moreno,Jaime.Penabad}@uclm.es

Abstract. The notion of least Herbrand model has been traditionally accepted as the declarative semantics for programs in the context of pure logic programming. Some adaptations of this concept, using model-theory, were made for a few number of fuzzy logic programming frameworks in the recent years. Unfortunately, this is not the case of multi-adjoint logic programming, one of the most expressive, powerful approaches for fuzzifying logic programming. To fulfill this gap, in this paper we propose a declarative semantics for such kind of fuzzy logic programs based on the so-called least fuzzy Herbrand model. We prove and important "minimality" property of our construction which can not trivially be inherited from pure logic programming. Moreover, apart from relating our notion with other existing procedural and fix-point semantics (what is also instrumental to prove its properties), we provide evident cases where our construction exists even when the rest of the aforementioned fuzzy semantics remain undefined.

Keywords: Fuzzy Logic Programming, Declarative Semantics, Herbrand Model.

1 Introduction

Soft computing and declarative programming represent two wide, powerful research areas with several links and influences between themselves. In this sense, although pure *logic programming* [5] has been largely used for problem solving and knowledge representation in the past, it has not directly taken into account the essential vagueness of human thinking. So, since the logical treatment of uncertainty is of increasing importance in soft computing, a considerable number of fuzzy extensions of logic programming systems have been developed to overcome this problem, as occurs with [1,2,4,8,10] and the so-called multi-adjoint logic programming approach of [6,7], one of the most expressive, well-formalized and promising languages in this field.

---

* This work has been partially supported by the EU, under FEDER, and the Spanish Science and Innovation Ministry (MICIN) under grant TIN 2007-65749 and by the Castilla-La Mancha Administration under grant PII1I09-0117-4481.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 253–260, 2009.
© Springer-Verlag Berlin Heidelberg 2009

In what follows, we present a short summary of the main features of our language (we refer the reader to [7] for a complete formulation). We work with a first order language, $\mathcal{L}$, containing variables, function symbols, predicate symbols, constants, quantifiers ($\forall$ and $\exists$), and several (arbitrary) connectives to increase language expressiveness. In our fuzzy setting, we use implication connectives ($\leftarrow_1, \leftarrow_2, \ldots, \leftarrow_m$) and also other connectives which are grouped under the name of "aggregators" or "aggregation operators". They are used to combine/propagate truth values through the rules. The general definition of aggregation operators subsumes conjunctive operators (denoted by $\&_1, \&_2, \ldots, \&_k$), disjunctive operators ($\vee_1, \vee_2, \ldots, \vee_l$), and average and hybrid operators (usually denoted by $@_1, @_2, \ldots, @_n$). Although the connectives $\&_i$, $\vee_i$ and $@_i$ are binary operators, we usually generalize them as functions with an arbitrary number of arguments. By definition, the truth function for an n-ary aggregation operator $\dot{@} : L^n \to L$ is required to be monotone and fulfills $\dot{@}(\top, \ldots, \top) = \top$, $\dot{@}(\bot, \ldots, \bot) = \bot$. Additionally, our language $\mathcal{L}$ contains the elements of a multi-adjoint lattice, $(L, \preceq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$, equipped with a collection of adjoint pairs $(\leftarrow_i, \&_i)$, where each $\&_i$ is a conjunctor intended to the evaluation of *modus ponens*. In general, the set of truth values L may be the carrier of any complete (bounded) lattice.

A *rule* is a formula $A \leftarrow_i \mathcal{B}$, where A is an atomic formula (usually called the *head*) and $\mathcal{B}$ (which is called the *body*) is a formula built from atomic formulas $B_1, \ldots, B_n$ ($n \geq 0$), truth values of L and conjunctions, disjunctions and aggregations. Rules with an empty body are called *facts*. A *goal* is a body submitted as a query to the system. Variables in a rule are assumed to be governed by universal quantifiers. Roughly speaking, a multi-adjoint logic program is a set of pairs $\langle \mathcal{R}; v \rangle$, where $\mathcal{R}$ is a rule and v is a *truth degree* (a value of L) expressing the confidence which the user of the system has in the truth of the rule $\mathcal{R}$.

In order to describe the procedural semantics of the multi-adjoint logic language, in the following we denote by $\mathcal{C}[A]$ a formula where A is a sub-expression (usually an atom) which occurs in the –possibly empty– context $\mathcal{C}[]$ whereas $\mathcal{C}[A/A']$ means the replacement of A by A' in context $\mathcal{C}[]$. Moreover, $\mathcal{V}ar(s)$ denotes the set of distinct variables occurring in the syntactic object s, $\theta[\mathcal{V}ar(s)]$ refers to the substitution obtained from θ by restricting its domain to $\mathcal{V}ar(s)$ and $mgu(E)$ denotes the *most general unifier* of an equation set E. In the next definition, we always consider that A is the selected atom in goal $\mathcal{Q}$ and L is the multi-adjoint lattice associated to $\mathcal{P}$.

Definition 1 (Admissible Step). *Let $\mathcal{Q}$ be a goal and let σ be a substitution. The pair $\langle \mathcal{Q}; \sigma \rangle$ is a* state. *Given a program $\mathcal{P}$, an* admissible computation *is formalized as a state transition system, whose transition relation $\to_{AS}$ is the smallest relation satisfying the following* admissible rules*:*

1) $\langle \mathcal{Q}[A]; \sigma \rangle \to_{AS} \langle (\mathcal{Q}[A/v\&_i\mathcal{B}])\theta; \sigma\theta \rangle$ *if* $\theta = mgu(\{A' = A\})$, $\langle A' \leftarrow_i \mathcal{B}; v \rangle$ *in* $\mathcal{P}$ *and $\mathcal{B}$ is not empty.*
2) $\langle \mathcal{Q}[A]; \sigma \rangle \to_{AS} \langle (\mathcal{Q}[A/v])\theta; \sigma\theta \rangle$ *if* $\theta = mgu(\{A' = A\})$, *and* $\langle A' \leftarrow_i ; v \rangle$ *in* $\mathcal{P}$.
3) $\langle \mathcal{Q}[A]; \sigma \rangle \to_{AS} \langle (\mathcal{Q}[A/\bot]); \sigma \rangle$ *if there is no rule in $\mathcal{P}$ whose head unifies with A (this case copes with possible unsuccessful branches).*

Definition 2 (Admissible Derivation). *Let $\mathcal{P}$ be a program with an associated multi-adjoint lattice $(L, \leq)$ and let $\mathcal{Q}$ be a goal. An* admissible derivation *is a sequence $\langle \mathcal{Q}; id\rangle \rightarrow^*_{AS} \langle \mathcal{Q}'; \theta\rangle$. When $\mathcal{Q}'$ is a formula not containing atoms and $r \in L$ is the result of interpreting $\mathcal{Q}'$ in $(L, \leq)$, the pairs $\langle \mathcal{Q}'; \sigma\rangle$ and $\langle r; \sigma\rangle$, where $\sigma = \theta[\mathcal{V}ar(\mathcal{Q})]$, are called* admissible computed answer *(a.c.a.) and* fuzzy computed answer *(f.c.a.), respectively (see [3] for details).*

Formulas are interpreted on a given multi-adjoint lattice and it is sufficient to consider (fuzzy) Herbrand interpretations [7]. A *fuzzy Herbrand interpretation* (or simply interpretation, for short) is a mapping $\mathcal{I} : B_{\mathcal{P}} \longrightarrow L$, where $B_{\mathcal{P}}$ is the Herbrand base of a program $\mathcal{P}$ and $(L, \leq)$ is the multi-adjoint lattice associated to $\mathcal{P}$. The truth value of a ground atom $A \in B_{\mathcal{L}}$ is $\mathcal{I}(A)$. The mapping $\mathcal{I}$ can be extended in a natural way to the set of ground formulas by structural induction. Given an assignment ϑ from terms into elements of the Herbrand universe $U_{\mathcal{L}}$, the valuation of a formula in an interpretation is:

$$\begin{aligned}
\mathcal{I}(p(t_1, \ldots, t_n))[\vartheta] &= \mathcal{I}(p(t_1\vartheta, \ldots, t_n\vartheta)),\\
\mathcal{I}(@(A_1, \ldots, A_n))[\vartheta] &= \dot{@}(\mathcal{I}(A_1)[\vartheta], \ldots, \mathcal{I}(A_n)[\vartheta]),\\
\mathcal{I}(A\leftarrow\mathcal{B})[\vartheta] &= \mathcal{I}(A)[\vartheta]\dot{\leftarrow}\mathcal{I}(\mathcal{B})[\vartheta],\\
\mathcal{I}((\forall x)\mathcal{A})[\vartheta] &= \inf\{\mathcal{I}(\mathcal{A})[\vartheta'] \mid \vartheta' \ \ x\text{–equivalent to } \ \vartheta\},
\end{aligned}$$

where p is a predicate symbol, @ an arbitrary aggregator, A and A_i atomic formulas, $\mathcal{B}$ any body, $\mathcal{A}$ any formula and we denote the truth value function of a connective @ by $\dot{@}$. An assignment ϑ' is x–equivalent to ϑ when $z[\vartheta'] = z[\vartheta]$ for all variable $z \neq x$ of $\mathcal{L}$. When the assignment would not be relevant, we shall omit it during the valuation of a formula. Moreover, an interpretation $\mathcal{I}$ satisfies a rule $\langle A\leftarrow_i\mathcal{B}; v\rangle$ if, and only if, $v \leq \mathcal{I}(A\leftarrow_i\mathcal{B})$, and an interpretation $\mathcal{I}$ is a *model* of $\mathcal{P}$ if, and only if, all rules in $\mathcal{P}$ are satisfied by $\mathcal{I}$.

The ordering $\leq$ in the lattice L can be easily extended to the set of Herbrand interpretations $\mathcal{H}$, as follows: $\mathcal{I}_1 \leq \mathcal{I}_2$ iff $\mathcal{I}_1(A) \leq \mathcal{I}_2(A)$ for all ground atom A. It is important to note that $(\mathcal{H}, \leq)$ is a complete lattice.

Once detailed the syntax and semantics of our target language, in order to motivate our work, it is important to say that we observe some lacks in the declarative foundations of the multi-adjoint logic programming approach, in the sense that there is not a characterization of the declarative semantics of a multi-adjoint logic program using the notion of least fuzzy Herbrand model. We think that this is a natural notion in the context of a (fuzzy) logic programming language that must be analyzed.

The outline of this paper is as follows. In Section 2, we define the declarative semantics of a multi-adjoint logic program in a model-theory style, following the guidelines of [5] (for pure logic programs) and we also point out some relevant aspects of our non-trivial adaptation. Next, in Section 3, we relate it with its equivalent fix-point characterization described in [7]: we illustrate that our notion of least fuzzy Herbrand model might exist even for fuzzy programs not belonging to the multi-adjoint logic class (and hence, for which a fix-point semantics does not exist). Finally, in Section 4 we present our conclusions and propose some lines of future work.

2 Declarative Semantics of Multi-Adjoint Logic Programs

In pure logic programming, the declarative semantics for a program is traditionally based on the least Herbrand model. In this section, we extend this kind of declarative semantics to the richer setting of multi-adjoint logic programming. Although there exists a fix-point characterization for the declarative semantics of a multi-adjoint logic program [7], ours follows the guidelines of [5] in a purely model-theoretic way, hence its novelty in our concrete fuzzy setting. Moreover, we have only found a few number of adaptations of the least Herbrand model to other fuzzy settings ([11,8]).

Theorem 1. *Let $\mathcal{P}$ be a multi-adjoint logic program and $\mathcal{J}_{\mathcal{P}} = \inf\{\mathcal{I}_j : \mathcal{I}_j$ is a model of $\mathcal{P}\}$. $\mathcal{J}_{\mathcal{P}}$ is the least fuzzy Herbrand model of $\mathcal{P}$.*

Proof. In the following, let the symbol $\mathcal{K}$ denote the set of model interpretations of $\mathcal{P}$, thas is, $\mathcal{K} = \{\mathcal{I}_j : \mathcal{I}_j \text{ is a model of } \mathcal{P}\}$.

Clearly $\mathcal{J}_{\mathcal{P}}$ is an Herbrand interpretation by construction. Since $\langle \mathcal{H}, \leq \rangle$ is a complete lattice, there exists the infimum of the set $\mathcal{K}$ and it is a member of $\mathcal{H}$.

Moreover, $\mathcal{J}_{\mathcal{P}}$ is also a model of $\mathcal{P}$. Because $\mathcal{J}_{\mathcal{P}}$ is the infimum of $\mathcal{K}$, $\mathcal{J}_{\mathcal{P}} \leq \mathcal{I}_j$ for each model $\mathcal{I}_j$ of $\mathcal{P}$. Therefore, $\mathcal{J}_{\mathcal{P}}(A) \leq \mathcal{I}_j(A)$ for each atom A. On the other hand, since each $\mathcal{I}_j$ is a model of $\mathcal{P}$, by definition of model, each rule $\mathcal{R} : \langle A{\leftarrow_i}\mathcal{B}; v\rangle$ in $\mathcal{P}$ is satisfied by $\mathcal{I}_j$, that is, $v \leq \mathcal{I}_j(A{\leftarrow_i}\mathcal{B})$. Now, by definition of interpretation, the monotonic properties of adjoint pairs in a multi-adjoint lattice, and because $\mathcal{J}_{\mathcal{P}}(A) \leq \mathcal{I}_j(A)$:

$$v \leq \mathcal{I}_j(A{\leftarrow_i}\mathcal{B}) = \mathcal{I}_j(A)\dot{\leftarrow}_i\mathcal{I}_j(\mathcal{B}) \leq \mathcal{J}_{\mathcal{P}}(A)\dot{\leftarrow}_i\mathcal{I}_j(\mathcal{B})$$

By the adjoint property, $v \leq \mathcal{J}_{\mathcal{P}}(A)\dot{\leftarrow}_i\mathcal{I}_j(\mathcal{B})$ iff $v\dot{\&}_i\mathcal{I}_j(\mathcal{B}) \leq \mathcal{J}_{\mathcal{P}}(A)$. Also, because the operation $\dot{\&}_i$ is increasing in both arguments and $\mathcal{J}_{\mathcal{P}}(\mathcal{B}) \leq \mathcal{I}_j(\mathcal{B})$, $v\dot{\&}_i\mathcal{J}_{\mathcal{P}}(\mathcal{B}) \leq \mathcal{J}_{\mathcal{P}}(A)$. Also, applying the adjoint property once more again, $v\dot{\&}_i\mathcal{J}_{\mathcal{P}}(\mathcal{B}) \leq \mathcal{J}_{\mathcal{P}}(A)$ iff $v \leq \mathcal{J}_{\mathcal{P}}(A)\dot{\leftarrow}_i\mathcal{J}_{\mathcal{P}}(\mathcal{B}) = \mathcal{J}_{\mathcal{P}}(A{\leftarrow_i}\mathcal{B})$. Therefore, $\mathcal{J}_{\mathcal{P}}$ satisfies each rule $\mathcal{R}$ in $\mathcal{P}$, being a model of $\mathcal{P}$.

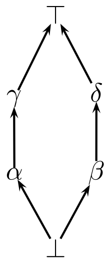
Trivially, since $\mathcal{J}_{\mathcal{P}}$ is the infimum of $\mathcal{K}$, it is the least model of $\mathcal{P}$, which concludes the proof.

Observe that, in the last proof it was essential the condition of the associated lattice of being a multi-adjoint lattice. As the following example reveals, if that condition is dropped Theorem 1 can not be proved.

Example 1. Let $(\&_{\mathtt{G}}, \leftarrow_{\mathtt{G}})$ a pair of connectives following the *Gödel*'s intuitionistic logic, that is, the truth functions for $\&_{\mathtt{G}}$ and $\leftarrow_{\mathtt{G}}$ are:

$$\dot{\&}_{\mathtt{G}}(x, y) = \inf\{x, y\} \qquad \text{and} \qquad \dot{\leftarrow}_{\mathtt{G}}(y, x) = \begin{cases} \top, \text{ if } x \leq y \\ y, \text{ otherwise} \end{cases}$$

For the lattice depicted below, these two connectives do not conform an adjoint pair since, in particular, note that although $\alpha \dot{\&} \delta \leq \beta$, it is not fulfilled that $\alpha \leq \beta \dot{\leftarrow} \delta$.



	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$	$\mathcal{I}_8$	$\mathcal{I}_9$	$\mathcal{I}_{10}$	$\mathcal{I}_{11}$	$\mathcal{I}_{12}$
p	β	δ	$\top$	α	γ	δ	$\top$	α	γ	$\top$	α	γ
q	β	β	β	β	β	δ	δ	δ	δ	$\top$	$\top$	$\top$

Given $\mathcal{P} = \{\mathcal{R}_1 : \langle p \leftarrow_{\mathtt{G}} q; \alpha\rangle, \mathcal{R}_2 : \langle q \leftarrow_{\mathtt{G}}; \beta\rangle\}$, the table of the figure summarizes the set $K = \{\mathcal{I}_1, \ldots, \mathcal{I}_{12}\}$ of all models of $\mathcal{P}$. Let us explain in detail how we have obtained such table. Note that, by definition of model, each $\mathcal{I}_j$ is a model of rule $\mathcal{R}_2$ if, and only if, $\beta \leq \mathcal{I}_j(q)$, which implies that $\mathcal{I}_j(q)$ admits the set of values $\beta, \delta, \top \in L$. Moreover, $\mathcal{I}_j$ is a model of $\mathcal{R}_1$ if, and only if, $\alpha \leq \mathcal{I}_j(p \leftarrow_{\mathtt{G}} q) = \mathcal{I}_j(p) \dot{\leftarrow}_{\mathtt{G}} \mathcal{I}_j(q)$. Then, once fixed $\mathcal{I}_j(q)$ to each one of the previous truth degrees, we add the condition $\alpha \leq \mathcal{I}_j(p) \dot{\leftarrow}_{\mathtt{G}} \mathcal{I}_j(q)$, which determines the set of valid values for $\mathcal{I}_j(p)$ and consequently, establishes the final set of models $\mathcal{I}_j \in K$. Finally, it is easy to see that K has an infimum element(the interpretation $\mathcal{J}_{\mathcal{P}}$ such that, $\mathcal{J}(p) = \bot$ and $\mathcal{J}_{\mathcal{P}}(q) = \beta$) but $\mathcal{J}_{\mathcal{P}} \notin K$, namely, $\mathcal{J}_{\mathcal{P}}$ it is not a model of $\mathcal{P}$.

We can assign declarative meaning to a fuzzy program by selecting one of its Herbrand models. Naturally, our preference must be guided by the election of the simplest model. This comment and Theorem 1 justify the following definition.

Definition 3. *The* declarative semantics *of a multi-adjoint logic program $\mathcal{P}$ is the least fuzzy Herbrand model $\mathcal{J}_{\mathcal{P}}$.*

To finish this section, note that given an interpretation $\mathcal{I}$, it is easy to prove that $\mathcal{I}$ determines a single binary relation $R_{\mathcal{I}} \subset B_{\mathcal{P}} \times L$. Under this point of view, since each model of $\mathcal{P}$ can be seen as a set of pairs (subset of $B_{\mathcal{P}} \times L$), the least fuzzy Herbrand model admits a second characterization which enjoys a nice property also reported in [5] for the weaker case of the least Herbrand model of pure logic programs. This new characterization is established in terms of the following theorem, which is immediate to prove and where we consider that $\cap \mathcal{I}_j = \{(A_i; \alpha) : (A_i, \alpha_i) \in \mathcal{I}_j, \forall j, \alpha = \inf\{\alpha_i\}\}$.

Theorem 2. *The least fuzzy Herbrand model of a program $\mathcal{P}$ is the intersection of all models of $\mathcal{P}$, i.e., $\mathcal{J}_{\mathcal{P}} = \cap \mathcal{I}_j$, where $\mathcal{I}_j$ is a model of $\mathcal{P}$ for all j.*

3 Fixpoint Characterization of $\mathcal{J}_{\mathcal{P}}$

In this section we provide a fixpoint characterization of the least fuzzy Herbrand model of a multi-adjoint logic program.

In [6,7] we find the following adaptation of the immediate consequences operator $T_{\mathcal{P}}$ of van Emden & Kowalski [9] to the multi-adjoint logic case:

Definition 4. *[7] Let $\mathcal{P}$ be a multi-adjoint logic program, $\mathcal{I}$ an interpretation and A a ground formula. Then, the $T_\mathcal{P}$ operator is a mapping in the set of interpretations such that, for any ground atom A*

$$T_\mathcal{P}(\mathcal{I})(A) = sup\{v\dot{\&}_i\mathcal{I}(\mathcal{B}\theta) \mid \langle C{\leftarrow}_i\mathcal{B}; v\rangle \in \mathcal{P}, A = C\theta\}$$

The monotonicity and continuity of $T_\mathcal{P}$ were proved in [7]. Another important property of $T_\mathcal{P}$ is its ability for detecting the models of a program.

Theorem 3. *[7] An interpretation $\mathcal{I}$ is a model of a program $\mathcal{P}$ iff $T_\mathcal{P}(\mathcal{I}) \leq \mathcal{I}$.*

So, for each model $\mathcal{I}$ of $\mathcal{P}$ we have: $_\mathcal{P}(\mathcal{I})(A) = sup\{v\dot{\&}_i\mathcal{I}(\mathcal{B}\theta) \mid \langle C{\leftarrow}_i\mathcal{B}; v\rangle \in \mathcal{P}, A = C\theta\} \leq \mathcal{I}(A)$. Moreover, apart from the previous result, the main use of $T_\mathcal{P}$ made in [7] was the design of a fix point semantics for the multi-adjoint framework. The following theorem establishes the equivalence between such construction and our notion of least fuzzy Herbrand model.

Theorem 4. *Given a multi-adjoint logic program $\mathcal{P}$, the least fuzzy Herbrand model $\mathcal{J}_\mathcal{P}$ of $\mathcal{P}$ is the least fix point of $T_\mathcal{P}$.*

Proof. By Theorem 1, $\mathcal{J}_\mathcal{P}$ is the least fuzzy Herbrand model of $\mathcal{P}$ iff $\mathcal{J}_\mathcal{P} = \inf\{\mathcal{I}_j : \mathcal{I}_j \text{ is a model of } \mathcal{P}\}$ and by Theorem 3, $\mathcal{J}_\mathcal{P} = \inf\{\mathcal{I}_j : T_\mathcal{P}(\mathcal{I}_j) \leq \mathcal{I}_j\}$. Moreover, since $T_\mathcal{P}$ is monotone in the complete lattice L (see [6]), there exists the least fix point of $T_\mathcal{P}$ which coincides with $\inf\{\mathcal{I}_j : T_\mathcal{P}(\mathcal{I}_j) \leq \mathcal{I}_j\}$ (see [5]). Hence, $\mathcal{J}_\mathcal{P}$ is the least fix point of $T_\mathcal{P}$.

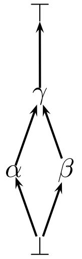
Last result states that the declarative semantics of a multi-adjoint logic program $\mathcal{P}$ can be obtained by transfinitely iterating $T_\mathcal{P}$ from the least interpretation.

Example 2. Consider the following multi-adjoint logic program $\mathcal{P}$ composed by facts (that is, rules whose bodies are implicitly assumed to be $\top$) and an associate lattice $(L, \leq)$ described by the Hasse's diagram of the figure:

$\mathcal{R}_1 : \langle p \leftarrow_\mathtt{G}\ ; \alpha\rangle$

$\mathcal{R}_2 : \langle p \leftarrow_\mathtt{G}\ ; \beta\rangle$

$\mathcal{R}_3 : \langle q \leftarrow_\mathtt{G}\ ; \beta\rangle$



	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$
p	γ	γ	γ	$\top$	$\top$	$\top$
q	β	γ	$\top$	β	γ	$\top$

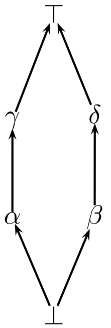
In contrast with Example 1 it is important to note now that the truth functions of connectives $(\&_\mathtt{G}, \leftarrow_\mathtt{G})$ as defined there (i.e., following the *Gödel*'s intuitionistic logic), verify the condition for conforming an adjoint pair regarding lattice $(L, \leq)$ of the figure above.

In this case there exist six different models (see $\mathcal{I}_1, \ldots, \mathcal{I}_6$ in the previous table) being $\mathcal{J}_\mathcal{P} = \mathcal{I}_1$ the least fuzzy Herbrand model. It is easy to see that $\mathcal{J}_\mathcal{P}$ is the minimal model (the only one) verifying the (least) fix point condition, since $T_\mathcal{P}(\mathcal{J}_\mathcal{P})(p) = sup\{\alpha\dot{\&}_\mathtt{G}\top, \beta\dot{\&}_\mathtt{G}\top\} = sup\{\alpha, \beta\} = \gamma = \mathcal{J}_\mathcal{P}(p)$ and $T_\mathcal{P}(\mathcal{J}_\mathcal{P})(q) = sup\{\beta\dot{\&}_\mathtt{G}\top\} = \beta = \mathcal{J}_\mathcal{P}(q)$. Observe also that Theorems 1 and 2 are fulfilled in this example, as it was expected.

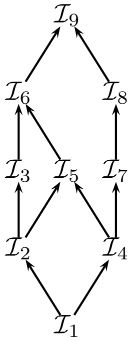
In [7], the procedural and fix point semantics of the multi-adjoint logic programming have been contrasted in order to establish the correctness of the framework. Due to the equivalence result we have just proved in Theorem 4, our notion of least fuzzy Herbrand model can directly inherit the properties asserted in [7]. Nevertheless, as we will discuss in the last section of this paper, we think that it would be possible to establish more direct and stronger relationships between the least fuzzy Herbrand model and the procedural semantics described at the beginning of this paper, without making explicit use of the fix-point semantics. Although out of the scope of this paper, in [7] the interested reader will find a discussion about the use of a kind of special rules called reductants for recovering completeness in the multi-adjoint logic setting (we also gave some proposals for the efficient construction and use of such concept in [3]).

To finish this section, we illustrate an important, surprising property of our notion of least fuzzy Herbrand model which has no precedents in other weaker settings. We are referring to its existence even in presence of those lattices not verifying the multi-adjoint properties, what indirectly implies that both the procedural and the fix-point semantics are undefined (note that both constructions make explicit use of the existing relations between the implications-conjunctions belonging to adjoint pairs).

Example 3. Consider again the lattice and the pair $(\&_{\texttt{G}}, \leftarrow_{\texttt{G}})$ of Example 1, for which we explained that the adjoint property was not fulfilled (remember that, in particular, even when $\alpha \dot{\&} \delta \leq \beta$, we have that $\alpha \leq \delta \dot{\rightarrow} \beta$ does not hold).



	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$	$\mathcal{I}_8$	$\mathcal{I}_9$
p	$\bot$	α	γ	$\bot$	α	γ	β	δ	$\top$
q	γ	γ	γ	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$



Consider now that program $\mathcal{P}$ contains the rules $\mathcal{R}_1 : \langle p \leftarrow_{\texttt{G}} q; \alpha \rangle$ and $\mathcal{R}_2 : \langle q \leftarrow_{\texttt{G}}; \gamma \rangle$, which make use of the lattice depicted in the figure. The set of models of $\mathcal{P}$ is showed in the table above, i.e., $K = \{\mathcal{I}_1, \ldots, \mathcal{I}_9\}$. Observe also that K maintains a lattice structure with an ordering inherited from $(\mathcal{H}, \leq)$, as also drawn in the figure. The graph evidences not only the existence of the least fuzzy Herbrand model of $\mathcal{P}$, represented by $\mathcal{I}_1$, but also that the properties proved in Theorems 1 (minimality) and 2 (model intersection) also hold in our case. All these facts have special importance when compared with the undefined fix-point semantics of this program (remember that this last notion requires the verification of the adjoint property, which is not fulfilled in this case).

4 Conclusions and Future Work

Although the semantics of a program was described in [7] in terms of the least fix point of an appropriately defined immediate consequences operator, a pending task in the field of multi-adjoint logic programming was its description following a purely model-theoretic way. This has been the main motivation of this work, where we have introduced the notion of least fuzzy Herbrand model of a program and proved some nice results, such as its "model intersection" property, which have a correlate in the pure logic programming setting. However, the inheritance has not always been achieved in a trivial and evident way. In doing this, we have also found examples of fuzzy programs (not verifying the adjoint property) where it is possible to establish the existence of a least fuzzy Herbrand model (when no fix-point semantics is defined) and the preservation of its "minimality" property.

On the other hand, due to the equivalence of our definition of $\mathcal{J}_\mathcal{P}$ and the fix point semantics of [7], we inherit from there the correctness properties of multi-adjoint-logic programming (established by contrasting fuzzy computed answers and correct answers for a given program and goal). Nevertheless, in the future we think that our present approach could help to improve such results in the following senses: i) It is possible to redefine the notion of "correct answers" in terms of the new notion of $\mathcal{J}_\mathcal{P}$ instead of the fix point semantics of [7]. ii) Regarding the quasi-completeness results of [7], we observe that they are referred to the use of special rules called "reductants". However, we are working now in the extension of Definition 1 in order to provide a more accurate procedural semantics not requiring the use of reductants for granting completeness.

References

1. Guadarrama, S., Muñoz, S., Vaucheret, C.: Fuzzy Prolog: A new approach using soft constraints propagation. Fuzzy Sets and Systems 144(1), 127–150 (2004)
2. Ishizuka, M., Kanai, N.: Prolog-ELF Incorporating Fuzzy Logic. In: Proc. of IJCAI 1985, pp. 701–703. Morgan Kaufmann, San Francisco (1985)
3. Julián, P., Moreno, G., Penabad, J.: An Improved Reductant Calculus using Fuzzy Partial Evaluation Techniques. Fuzzy Sets and Systems 160(2), 162–181 (2009)
4. Krajči, S., Lencses, R., Vojtáš, P.: A comparison of fuzzy and annotated logic programming. Fuzzy Sets and Systems 144, 173–192 (2004)
5. Lloyd, J.W.: Foundations of Logic Programming. Springer, Berlin (1987)
6. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: Multi-adjoint logic programming with continuous semantics. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, pp. 351–364. Springer, Heidelberg (2001)
7. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: Similarity-based Unification: a multi-adjoint approach. Fuzzy Sets and Systems 146, 43–62 (2004)
8. Sessa, M.I.: Approximate reasoning by similarity-based SLD resolution. Fuzzy Sets and Systems 275, 389–426 (2002)
9. van Emden, M.H., Kowalski, R.A.: The semantics of predicate logic as a programming language. Journal of the ACM 23(4), 733–742 (1976)
10. Vojtáš, P.: Fuzzy Logic Programming. Fuzzy Sets and Systems 124, 361–370 (2001)
11. Vojtáš, P., Paulík, L.: Soundness and completeness of non-classical extended SLD-resolution. In: Herre, H., Dyckhoff, R., Schroeder-Heister, P. (eds.) ELP 1996. LNCS, vol. 1050, pp. 289–301. Springer, Heidelberg (1996)

A Complete Logic for Fuzzy Functional Dependencies over Domains with Similarity Relations*

P. Cordero, M. Enciso, A. Mora, and I.P. de Guzmán

E.T.S.Ingeniería Informática. Universidad de Málaga. Spain
{pcordero,enciso}@uma.es, {amora,guzman}@ctima.uma.es

Abstract. An axiomatic system for fuzzy functional dependencies is introduced. The main novelty of the system is that it is not based on the transitivity rule like all the others, but it is built around a simplification rule which allows the removal of redundancy. The axiomatic system presented here is shown to be sound and complete.

1 Introduction

Lately, L. Zadeh [27] is using the term Fuzzy Logic in a wider sense than it has been used in the past. He suggests that the notion of fuzzy logic is more than a logical system and it has four facets: the logical facet -FLl- which is fuzzy logic in its narrow sense, the fuzzy-set-theoretical facet -FLs- which arises from the fact that the theory of fuzzy sets is the basis of fuzzy logic in its wide sense, the epistemic facet -FLe- which concerns with knowledge representation, semantics of natural language and information analysis and finally the relational facet -FLr-focussed on fuzzy relations and fuzzy dependency. In this last facet, L. Zadeh refers to some works devoted to the notion of Fuzzy Attribute Implication [4]. This notion corresponds with the idea of Fuzzy Functional Dependency (FFD), but they are defined in two different frameworks: lattice theory and the relational model respectively. In this paper we work with FFDs in the second area.

Functional Dependency (FD) was first introduced in the environment of the Relational Model [8] to specify a strong relation between two sets of attributes of a table [1,5] and it has been successfully used in other areas in Computer Science. So, a Functional Dependency between attributes A and B of a relation ($A \rightarrow B$) denotes that any two rows which agree on A, also agree on B. FDs may be considered as the pioneer of other kinds of dependencies defined over the Classical Relational Model and its extensions (multivalued dependencies, join dependencies, temporal dependencies, dependencies in semi-structured data, fuzzy dependencies, etc). Functional Dependencies were introduced to identify update anomalies in database system and they play an outstanding role in the development of the Theory of Normalization [24]. The design of a normalization process is based on a selected framework which allows us to reasoning about

---

* Partially supported by Spanish project TIN07-65819.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 261–269, 2009.
© Springer-Verlag Berlin Heidelberg 2009

functional dependencies. One of the most popular frameworks is propositional logic and several classical logics for functional dependencies have been introduced in the literature [1,12,24]. These logics are not only equivalent, but they are very similar with very little and slight differences among them.

The relational model has been successfully generalized to consider imprecise knowledge. In the original model the imprecise information was transformed into NULL values, but in the Fuzzy Relational Model, data become fuzzy. Three approaches have been proposed in the literature [22] to store and manage this dimension of the data: the use of fuzzy membership values [13,3], the introduction of possibility distribution [18, 19] or similarity relations [6]. In this work we consider this last approach and introduce a similarity relation between the values of the attributes.

In this approach the Functional Dependency is generalized into a Functional Dependency over Domains with Similarity Relations and some definition of Fuzzy Functional Dependency (FFD) have been introduced [26,23,25]. The FD is extended to consider imprecise information and a FFD between attributes A and B is read as: for any two tuples which have similar values on A, they must have similar values on B. But this is not the only way to generalize the notion of FD towards impreciseness. In [22] the authors propose to generalize the constraint itself. Thus, they do not to force that the similarity of the B-values must be satisfied by all the tuples in the relation. So, a more general reading of a FDD between the attributes A and B is: for any two tuples which have similar values on A, they usually have similar values on B. This new dependency is denoted as $A \xrightarrow{\theta} B$, where θ represents a threshold for the fuzziness of the Functional Dependency. In this work we will consider this more generalized notion of Fuzzy Functional Dependency. We will also show that the extension of this FFDs to consider multi-valued attributes (i.e. relations which are not in first normal form [15]) is not an extension but a particular case of the FFD over relations in first normal form.

All the classical FD logics and their extension to FFD lack of an executable perspective and they were designed to be a formal framework to show the semantics of the functional dependency (either classical or fuzzy). As a matter of fact, they have not been used to reason about functional dependencies and it is not a trivial matter to design an automatic method directly based on their axiomatic system. The main obstacle is that all of them are strongly based on the transitivity axiom, which limits the efficiency of the automatic deduction methods for these logics. In [9] we introduce a new logic for classical FD where the transitivity rule does not appear as a primitive rule of its axiomatic system. The core of the new axiomatic system is a new *Simplification Rule* with a strong executable orientation. In this work we introduce for the first time an axiomatic system for Fuzzy Functional Dependencies where the core is a Fuzzy Simplification Rule instead of the usual Fuzzy Transitivity Rule. We formally present the logic, its axiomatic system and we prove its soundness and completeness. As a motivation of its usefulness, we show how the simplication rule for FFD may be used to remove redundancy from a set of FFDs.

This work is organized as follows: section 2 introduces the notation of the basic concepts for the classical functional dependencies. In section 3 we select a notion of Fuzzy Functional Dependency and we formally define the FFDs used in this work (the one introduced in [22]). A new simplification rule for FFDs and its complete axiomatic system is presented in subsection 4.1. We prove its correctness and completeness in subsection 4.2 and motivate its usefulness in subsection 4.3. Conclusions are presented in the last section of the paper.

2 Functional Dependencies

We present here the basic concepts about functional dependencies as they appear in database literature. Let $\{D_a \mid a \in \mathcal{A}\}$ be a family of sets indexed in a finite non-empty set of indexes $\mathcal{A}$. We call *attributes* to the indexes and *domain of the attribute* a to the set D_a. We work over the product of these domains, $\mathbb{D} = \prod\limits_{a\in\mathcal{A}} D_a$. The elements in this product $t = (t_a)_{a\in\mathcal{A}} \in \mathbb{D}$ will be named tuples. A relation is a set of tuples $R \subseteq \mathbb{D}$, usually represented as a table.

We introduce here the notation widely accepted in the database community. Given $X, Y \subseteq \mathcal{A}$, XY denotes $X \cup Y$. Given $X \subseteq \mathcal{A}$, D_X denotes $\prod_{a\in X} D_a$. Let $t \in R$ be a tuple, then $t_{/X}$ denotes the *projection* of t to D_X; that is, if $t = (t_a)_{a\in\mathcal{A}}$ then $t_{/X} = (t_a)_{a\in X}$.

Definition 1. *Any affirmation of the type* $X \rightarrow Y$*, where* $X, Y \subseteq \mathcal{A}$*, is called a* **functional dependency**. *We say that a relation* $R \in \mathbb{D}$ *satisfies* $X \rightarrow Y$ *if, for all* $t_1, t_2 \in R$ *we have that:* $t_{1/X} = t_{2/X}$ *implies that* $t_{1/Y} = t_{2/Y}$.

Remark 1. The term *functional* comes from the fact that: the relation R satisfies the FD $X \rightarrow Y$ if R restricted to XY is a (partial) function from D_X to D_Y.

3 Fuzzy Functional Dependencies

The most usual way to fuzzify the concept of functional dependency is by replacing the equality in the definition by similarity relations. We will consider that each domain D_a is endowed with a similarity relation $\rho_a : D_a \times D_a \to [0,1]$, that is, a reflexive, symmetric and max-min-transitive fuzzy relation[1]. We can extend these relations to D_X for all $X \subseteq \mathcal{A}$ as follow:

$$\rho_X(t_1, t_2) = \min\{\rho_a(t_1, t_2) \mid a \in X\}$$[2]

Remark 2. The definitions of fuzzy functional dependency in the literature [26, 23, 25, 20, 10] are very similar, having slight differences among them. They fuzzify the equality between the attributes value in the following way: *A relation* $R \subseteq \mathbb{D}$ *satisfies the FFD* $X \rightarrow Y$ *if* $\rho_Y(t_1, t_2) \geq \rho_X(t_1, t_2)$ *holds, for all* $t_1, t_2 \in R$.

---

[1] Note that the relations are defined in finite sets.

[2] To simplify the notation, when no confusion arise, we write $\rho_X(t_1, t_2)$ instead of $\rho_X(t_{1/X}, t_{2/X})$.

Although the definition introduces the fuzzy relations of similarity and generalizes the classical definition, we can say that the functional dependency remains crisp. The inclusion of a degree of fuzziness in the dependency itself is done in [22]. In this work we adopt the following definition of fuzzy functional dependency.

Definition 2. *A* **fuzzy functional dependency** *is an expression* $X \xrightarrow{\theta} Y$ *where* $X, Y \subseteq \mathcal{A}$ *and* $\theta \in [0,1]$. *A relation* $R \subseteq \mathbb{D}$ *is said to satisfy* $X \xrightarrow{\theta} Y$ *if, for all* $t_1, t_2 \in R$, *the inequation* $\rho_Y(t_1,t_2) \geq \min\{\theta, \rho_X(t_1,t_2)\}$ *holds.*

We remark that, if $\theta = 1$, the previous definition of FFD matches up with the definition of FFD proposed in [20].

Sozat and Yazici [22,23] work with multivalued attributes, considering that the values of the attributes are sets. In the relational model, this model corresponds with relations which are not in First Normal Form [15]. Given $a \in \mathcal{A}$, the elements of D_a are sets, $D_a = 2^{B_a}$, and the similarity relation ρ_a is defined over the set B_a. An extension of the similarity relations to relations on D_a is required. The new relation, introduced by them, is named conformance and it is defined as follows:

Definition 3. *Let* $a \in \mathcal{A}$ *be an attribute and let* $U, V \in D_a$ *be two elements over a domain* D_a, **the conformance** *is defined as*

$$\mathcal{C}_a(U,V) = \min\Big\{\min_{\lambda \in U}\Big\{\max_{\mu \in V}\{\rho_a(\lambda,\mu)\}\Big\}, \min_{\mu \in V}\Big\{\max_{\lambda \in U}\{\rho_a(\lambda,\mu)\}\Big\}\Big\}$$

and its extension to a set of attributes, $X \subseteq \mathcal{A}$, *is* $\mathcal{C}_X(t_1,t_2) = \min_{a \in X}\{\mathcal{C}_a(t_1,t_2)\}$ *for all* $t_1, t_2 \in D_X$.

They study several properties of the conformance relation and finally they also give the definition of fuzzy functional dependency. Nevertheless, relations which are not in First Normal Form obscure the nature of FD and it is not necessary to be considered because any fuzzy relation can be extended to the power-set. Moreover, in [7], we show that we may consider only similarity relations over non power-set domains:

Definition 4. *Let* $\mathcal{U}$ *be a set and* ρ *a similarity relation,* $\rho : \mathcal{U} \times \mathcal{U} \to [0,1]$. *Its* **power extension** $\widehat{\rho} : 2^{\mathcal{U}} \times 2^{\mathcal{U}} \to [0,1]$ *is defined as follows:*

$$\widehat{\rho}(A,B) = \min\{\inf_{a \in A}\{\sup_{b \in B} \rho(a,b)\}, \inf_{b \in B}\{\sup_{a \in A} \rho(a,b)\}\}, \ for\ all\ A, B \subseteq U.$$

Theorem 1. *Let* ρ *be a fuzzy relation and* $\widehat{\rho}$ *its power-extension.* ρ *is a similarity relation if and only if* $\widehat{\rho}$ *is a similarity relation.*

We emphasize that the definition of conformance given in [22] is a particular case of our definition 4. Thus, Theorem 1 (proved in [7]) allows us to use fuzzy relations in first normal form without generality loss.

4 The Simplification Logic for FFDs

We are interested in an axiomatic system that allows us to syntactically derive FFDs. There exists in the literature some complete axiomatic system defined over FFDs with similarity relations [26,23,25]. However, there are not many axiomatic systems in the literature to reasoning with fuzzy functional dependencies where the dependency is fuzzy. One of them is given by Sozat and Yazici [22]. It is a fuzzy extension of Armstrong Axiom's and it has the problem inherent in transitivity.

4.1 FSLlogic

We introduce **FSL**, a new logic more adequate for the applications, named *Fuzzy Simplification Logic* for fuzzy functional dependencies. Its language is the following:

Definition 5. *Given a numerable set of attribute symbols* $\mathcal{A}$*, we define* **the language L** $= \{X \xrightarrow{\theta} Y \mid X, Y \in 2^{\mathcal{A}},\ X \neq \varnothing \text{ and } \theta \in [0,1]\}$.

We are interested in the implication between the syntactic and the semantic level. Given a relation $R \subseteq \mathbb{D}$, a fuzzy functional dependency $X \xrightarrow{\theta} Y$ and a set of FFDs Γ, $R \models X \xrightarrow{\theta} Y$ denotes that R satisfies $X \xrightarrow{\theta} Y$, $R \models \Gamma$ denotes that R satisfies every fuzzy functional dependency in Γ and $\Gamma \models X \xrightarrow{\theta} Y$ denotes that, for all $R \subseteq \mathbb{D}$, $R \models \Gamma$ implies $R \models X \xrightarrow{\theta} Y$. In this point we present the axiomatic system:

Definition 6. *The* **axiomatic system** $\mathcal{S}_F$ *on* **L** *has one axiom scheme:*[3]

Ax*:* $\vdash_{\mathcal{S}_F} X \xrightarrow{1} Y$ *, for all* $Y \subseteq X$ **Reflexive Axioms**

The inferences rules are the following:

InR*:* $X \xrightarrow{\theta_1} Y \ \vdash_{\mathcal{S}_F} X \xrightarrow{\theta_2} Y$*, if* $\theta_1 \geq \theta_2$ **Inclusion Rule**

DeR*:* $X \xrightarrow{\theta} Y \vdash_{\mathcal{S}_F} X \xrightarrow{\theta} Y'$*, if* $Y' \subseteq Y$ **Decomposition Rule**

CoR*:* $X \xrightarrow{\theta_1} Y,\ U \xrightarrow{\theta_2} V \ \vdash_{\mathcal{S}_F} XU \xrightarrow{min(\theta_1,\theta_2)} YV$ **Composition Rule**

SiR*:* $X \xrightarrow{\theta_1} Y,\ U \xrightarrow{\theta_2} V \ \vdash_{\mathcal{S}_F} U\text{-}Y \xrightarrow{min(\theta_1,\theta_2)} V\text{-}Y$,
if $X \subseteq U$ *and* $X \cap Y = \varnothing$ **Simplification Rule**

The *deduction* ($\vdash_{\mathcal{S}_F}$) and *equivalence* ($\equiv_{\mathcal{S}_F}$) concepts are introduced as usual.

4.2 Soundness and Completeness of FSLlogic

In [22] Sozat and Yazici prove that the following Axiomatic System, $\mathcal{S}_{SY}$, is a sound and complete formal system to manage FFDs.

---

[3] In the literature, the set of attributes Y must be non-empty. In **FSL**, we consider the empty attribute, denoted $\top$. Note that $X \xrightarrow{1} \top$ is an axiom scheme.

The Sozat and Yazici Axiomatic System The axioms in $\mathcal{S}_{SY}$ are:

Ax: $\vdash_{S_{SY}} X \xrightarrow{1} Y$, for all $Y \subseteq X$ **Reflexive Axioms**

and the inference rules considered are:

InR: $X\xrightarrow{\theta_1}Y \vdash_{S_{SY}} X\xrightarrow{\theta_2}Y$, if $\theta_1 \geq \theta_2$ **Inclusive Rule**

AuR: $X\xrightarrow{\theta}Y \vdash_{S_{SY}} XZ\xrightarrow{\theta}YZ$ **Augmentation Rule**

TrR: $X\xrightarrow{\theta_1}Y,\ Y\xrightarrow{\theta_2}Z \vdash_{S_{SY}} X\xrightarrow{min(\theta_1,\theta_2)}Z,$ **Transitivity Rule**

Now, we prove that the axiomatic system of the new **FSL** logic is equivalent to this axiomatic system.

Theorem 2. *$\mathcal{S}_F$ and $\mathcal{S}_{SY}$ are equivalent.*

Proof. First, we prove that **DeR**, **CoR** and **SiR** are derived rules in $\mathcal{S}_{SY}$. The proof for decomposition rule (**DeR**) appears in [22].
Composition rule (**CoR**): Let $X\xrightarrow{\theta_1}Y, U\xrightarrow{\theta_2}V \in \mathbf{L}$,

1. $X\xrightarrow{\theta_1}Y$ Hypothesis
2. $U\xrightarrow{\theta_2}V$ Hypothesis
3. $XU\xrightarrow{\theta_1}YU$ 1, **AuR**
4. $YU\xrightarrow{\theta_2}YV$ 2, **AuR**
5. $XU\xrightarrow{min(\theta_1,\theta_2)}YV$ 4, 5, **TrR**

Simplification rule (**SiR**): Let $X\xrightarrow{\theta_1}Y, U\xrightarrow{\theta_2}V \in \mathbf{L}$ such that $X \subseteq U$ and $X \cap Y = \varnothing$. Then $X \subseteq U\text{-}Y$ and, therefore, $X(U\text{-}Y) = U\text{-}Y$ and $Y(U\text{-}Y) = YU$.

1. $X\xrightarrow{\theta_1}Y$ Hypothesis
2. $U\xrightarrow{\theta_2}V$ Hypothesis
3. $(U\text{-}Y)\xrightarrow{\theta_1}YU$ 1, **AuR**
4. $(U\text{-}Y)\xrightarrow{\theta_1}U$ 3, **DeR**
5. $(U\text{-}Y)\xrightarrow{min(\theta_1,\theta_2)}V$ 2, 4, **TrR**
6. $(U\text{-}Y)\xrightarrow{min(\theta_1,\theta_2)}(V\text{-}Y)$ 5, **DeR**

Conversely, we prove that **AuR** and **TrR** are derived rules in $\mathcal{S}_F$. The first one is straightforward and the proof of the second one is:

1. $X\xrightarrow{\theta_1}Y$ Hypothesis
2. $Y\xrightarrow{\theta_2}Z$ Hypothesis
3. $X\xrightarrow{\theta_1}Y\text{-}X$ 1, **DeR**
4. $Y\xrightarrow{\theta_2}Z\text{-}Y$ 2, **DeR**
5. $X\xrightarrow{1}\top$ **Ax**
6. $XY\xrightarrow{\theta_2}Z\text{-}Y$ 4, 5, **CoR**
7. $XY\text{-}(Y\text{-}X)\xrightarrow{min(\theta_1,\theta_2)}(Z\text{-}Y)\text{-}(Y\text{-}X)$ 3, 6, **SiR**
   $= X\xrightarrow{min(\theta_1,\theta_2)}Z\text{-}Y$
8. $X\xrightarrow{min(\theta_1,\theta_2)}ZY$ 1, 7, **CoR**
9. $X\xrightarrow{min(\theta_1,\theta_2)}Z$ 8, **DeR**

4.3 Redundancy Elimination via Simplification

In database systems we look for designs with no redundancy and functional dependencies were defined to capture some semantics of the data strongly connected with the occurrence of redundancy in a database. In the same sense, redundancy is not desirable in the integrity constraint of a database.

The logic that we have introduced is more adequate for the applications. In the following, we illustrate this assertion by showing its good behavior for removing redundancy. The primitive rules allow us to directly eliminate redundancy without using other more costly tools. It is possible because our inference rules are really equivalences if redundancy exists, as the following theorem shows. The systematically application of the following equivalences removes redundancy.

Theorem 3. *Let* $X\xrightarrow{\theta_1}Y, U\xrightarrow{\theta_2}V \in \mathbf{L}$.

Decomposition equivalence: If $X \cap Y \neq \varnothing$, *then* $\{X\xrightarrow{\theta_1}Y\} \equiv_{\mathcal{S}_F} \{X\xrightarrow{\theta_1}Y\text{-}X\}$.

Simplification equivalence: If $X \cap Y = \varnothing$, $X \subseteq U$ *and* $\theta_1 \geq \theta_2$, *then*

$$\{X\xrightarrow{\theta_1}Y, U\xrightarrow{\theta_2}V\} \equiv_{\mathcal{S}_F} \{X\xrightarrow{\theta_1}Y, U\text{-}Y\xrightarrow{\theta_2}V\text{-}Y\}$$

Proof. The decomposition equivalence is straightforward. In the second equivalence, the left-right implication is due to **SiR** and the proof of the converse implication is the following:

1. $U\xrightarrow{1}\top$	**Ax**	4. $U\text{-}Y\xrightarrow{\theta_2}V\text{-}Y$	Hypothesis
2. $X\xrightarrow{\theta_1}Y$	Hypothesis	5. $U\xrightarrow{\theta_2}VY$	3, 4, **CoR**
3. $U\xrightarrow{\theta_1}Y$	1, 2, **CoR**	6. $U\xrightarrow{\theta_2}V$	5, **DeR**

5 Conclusions and Future Works

In [2] twenty nine leading expert on database system present the guidelines of the new development of the database theory and point out the need to face on with some open problems. The management of imprecise data is cited as a very interesting problem not completely fulfilled. In this work we deal with this problem and present some interesting results related with logic and fuzzy relational database systems.

We have dealt with two of the facets cited by L. Zadeh [27]: the logical facet -FLl- and the relational facet -FLr-. We have selected the most generalized version of functional dependency, considering domain of data with similarity relations and including a threshold for the fuzziness of the dependency itself. This is the most general notion because the inclusion of attributes with multi-valued attributes (non first normal form) is not an extension as we showed in Theorem 1. We have introduced a new inference rule for this general notion of fuzzy functional dependency and we have presented a new logic. The new axiomatic system has been designed looking for good executable characteristics and we motivate it by showing how the simplification rule removes redundancy from a set of FFDs. The soundness and completeness results for this axiomatic systems have been proven.

As future works in the use of **FSL** logic, we are developing an algorithm to automatically remove the redundancy of FFDs. We will also study an extension of the closure algorithm and we will face on with the implication problem using the rules of the **FSL** logic. And finally, a generalization of this work using t-norms defined in a lattice, instead unit interval $[0, 1]$, is nowadays in course.

References

1. Armstrong, W.W.: Dependency structures of data base relationships. In: Proc. IFIP Congress, pp. 580–583. North-Holland, Amsterdam (1974)
2. Abiteboul, S., Agrawal, R., Bernstein, P.A., et al.: The Lowell database research self-assessment. Commun. ACM 48(5), 580–583 (2005)
3. Baldwin, J.F.: Knowledge engineering using a fuzzy relational inference language. In: Proc. IFAC Conf. on Fuzzy Information, Knowledge Representation, and Decision Processes, Marseille, France, pp. 15–23 (1983)
4. Belohlávek, R., Vychodil, V.: Basic Algorithm for Attribute Implications and Functional Dependencies in Graded Setting. Int. J. Found. Comput. Sci. 19(2), 297–317 (2008)
5. Bernstein, P.A., Swenson, J.R.: A unified approach to functional dependencies and relations. In: Proc. ACM SIGMOD Conf., San Jose, CA, pp. 237–245 (1974)
6. Buckles, B.P., Petry, F.E.: Uncertainty models in information and database systems. J. Inform. Sci. 11, 77–87 (1985)
7. Cabrera, I.P., Cordero, P., Gutiérrez, G., Martínez, J., Ojeda-Aciego, M.: Fuzzy congruence relations on nd-groupoids, Int. J. on Computer Mathematics (to appear, 2009)
8. Codd, E.F.: The relational model for database management: Version 2. Addison Wesley, Reading (1990)
9. Cordero, P., Enciso, M., de Guzmán, I.P., Mora, A.: Slfd logic: Elimination of data redundancy in knowledge representation. In: Garijo, F.J., Riquelme, J.-C., Toro, M. (eds.) IBERAMIA 2002. LNCS (LNAI), vol. 2527, pp. 141–150. Springer, Heidelberg (2002)
10. Cordero, P., Mora, A., Enciso, M., Aguilera, G., de Guzmán, I.P.: Reasoning about Fuzzy Functional Dependencies. In: ESTYLF 2008, pp. 121–126 (2008)
11. Cubero, J.C., Vila, M.A.: A new definition of fuzzy functional dependency in fuzzy relational databases. Internat. J. Intell. Systems 9(5), 441–448 (1994)
12. Fagin, R.: Functional dependencies in a relational database and propositional logic. IBM J. of Research and Development 21(6), 534–544 (1977)
13. Giardina, C., Sack, I., Sinha, D.: Fuzzy Field Relational Database. Report 8332, Elect. Engng. and Computer Science Dept., Stevens Institute of Technology, Hoboken, NJ (1983)
14. Guan, J.W., Bell, D.A.: Rough computational methods for information systems. Artificial Intelligence 1051(2), 77–103 (1998)
15. Jaeschke, G., Schek, H.-J.: Remarks on the algebra of non first normal form relations. In: Proceedings of the ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, pp. 124–138. ACM Press, New York (1982)
16. Mora, A., Enciso, M., Cordero, P., de Guzmán, I.P.: The functional dependence implication problem: optimality and minimality. In: Conejo, R., Urretavizcaya, M., Pérez-de-la-Cruz, J.-L. (eds.) CAEPIA/TTIA 2003. LNCS, vol. 3040, pp. 136–146. Springer, Heidelberg (2004)
17. Mora, A., Aguilera, G., Enciso, M., Cordero, P., de Guzmán, I.P.: A new closure algorithm based in logic: SLFD-Closure versus classical closures. Inteligencia Artificial, Revista Iberoamericana de IA 31(10), 31–40 (2006)
18. Prade, H.: The connection between Lipskios approach to incomplete information data bases and Zadeh's Possibility Theory. In: Proc. Int. Conf. Systems Meth., pp. 402–408 (1982)

19. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. Information Sciences 34, 115–143 (1984)
20. Raju, H.V.S.V.N., Mazumdar, A.K.: Fuzzy dependencies and losless join descomposition of fuzzy relational database systems. ACM Transaction of Database Systems 13(2), 129–166 (1988)
21. Savnik, I., Flach, P.A.: Bottom-up induction of functional dependencies from relations. In: Proc. of AAAI 1993 Workshop: Knowledge Discovery in Databases, pp. 174–185 (1993)
22. Sozat, M.I., Yazici, A.: A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations. Fuzzy Sets and Systems 117(2), 161–181 (2001)
23. Tyagi, B.K., Sharfuddin, A., Dutta, R.N., Tayal, D.K.: A complete axiomatization of fuzzy functional dependencies using fuzzy function. Fuzzy Set and Systems 151(2), 363–379 (2005)
24. Ullman, J.D.: Database and knowledge-base systems. Computer Science Press (1988)
25. Bělohlávek, R., Vychodil, V.: Data Tables with Similarity Relations: Functional Dependencies, Complete Rules and Non-redundant Bases. In: Li Lee, M., Tan, K.-L., Wuwongse, V. (eds.) DASFAA 2006. LNCS, vol. 3882, pp. 644–658. Springer, Heidelberg (2006)
26. Yahia, S.B., Ounalli, H., Jaoua, A.: An Extension of Classical Functional Dependency: Dynamic Fuzzy Functional Dependency. Inf. Sci. 119(3-4), 219–234 (1999)
27. Zadeh, L.: Is there a need for fuzzy logic? Information Science 178(13), 2751–2779 (2008)

RFuzzy: An Expressive Simple Fuzzy Compiler

Susana Munoz-Hernandez, Victor Pablos Ceruelo, and Hannes Strass

Universidad Politécnica de Madrid*
{susana,vpablos}@fi.upm.es, hannes.strass@alumnos.upm.es
http://babel.ls.fi.upm.es/

Abstract. Fuzzy reasoning is a very productive research field that during the last years has provided a number of theoretical approaches and practical implementation prototypes. Nevertheless, the classical implementations, like Fril, are not adapted to the latest formal approaches, like multi-adjoint logic semantics.

Some promising implementations, like Fuzzy Prolog, are so general that the regular user/programmer does not feel comfortable because either the representation of fuzzy concepts is complex or the results of the fuzzy queries are difficult to interpret.

In this paper we present a modern framework, *RFuzzy*, that is modeling multi-adjoint logic in a practical way. It provides some extensions as default values (to represent missing information), partial default values (for a subset of data) and typed variables. *RFuzzy* represents the truth value of predicates using facts, rules and also can define fuzzy predicates as continuous functions. Queries are answered with direct results (instead of providing complex constraints), so it is easy to use for any person that wants to represent a problem using fuzzy reasoning in a simple way (just using the classical fuzzy representation with real numbers). The most promising characteristic of *RFuzzy* is that the user can obtain constructive answers to queries that restrict the truth value.

Keywords: Fuzzy reasoning, Implementation tool, Fuzzy Logic, Multi-adjoint logic, Logic Programming Implementation, Fuzzy Logic Application.

1 Introduction

One of the reasoning models that is more useful to represent real situations is fuzzy reasoning. Indeed, world information is not represented in a crisp way. Its representation is imperfect, fuzzy, etc., so that the management of uncertainty is very important in knowledge representation. There are multiple frameworks for incorporating uncertainty (in the sense of fuzziness) in logic programming.

---

* This work is partially supported by the project DESAFIOS - TIN 2006-15660-C02-02 from the Spanish Ministry of Education and Science, by the Spanish Ministry of Science and Innovation Research Staff Training Program - BES-2008-008320 and by the project PROMESAS - S-0505/TIC/0407 from the Madrid Regional Government.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 270–277, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Despite of the multitude of theoretical approaches to this issue, few of them resulted in current usable tools. Since Logic Programming is traditionally used in Knowledge Representation and Reasoning, we argue (as in [15]) that it is perfectly well-suited to implement a fuzzy reasoning tool as ours.

1.1 Fuzzy Approaches in Logic Programming

Introducing Fuzzy Logic into Logic Programming has provided the development of several fuzzy systems over Prolog. These systems replace its inference mechanism, SLD-resolution, with a fuzzy variant that is able to handle partial truth. Most of these systems implement the fuzzy resolution introduced by Lee in [4], as the Prolog-Elf system, the FRIL Prolog system and the F-Prolog language. However, there is no common method for fuzzifying Prolog, as noted in [11].

Some of these Fuzzy Prolog systems only consider fuzziness on predicates whereas other systems consider fuzzy facts or fuzzy rules. There is no agreement about which fuzzy logic should be used. Most of them use min-max logic (for modelling the conjunction and disjunction operations) but other systems just use Łukasiewicz logic.

There is also an extension of constraint logic programming [2], which can model logics based on semiring structures. This framework can model min-max fuzzy logic, which is the only logic with semiring structure. Another theoretical model for fuzzy logic programming without negation has been proposed by Vojtáš in [14], which deals with many-valued implications.

1.2 Fuzzy Prolog

One of the most promising fuzzy tools for Prolog was the "Fuzzy Prolog" system [13,3]. The most important advantages against the other approaches are:

1. A truth value is represented as a finite union of sub-intervals on $[0, 1]$. An interval is a particular case of union of one element, and a unique truth value (a real number) is a particular case of having an interval with only one element.
2. A truth value is propagated through the rules by means of an *aggregation operator*. The definition of this *aggregation operator* is general and it subsumes conjunctive operators (triangular norms like min, prod, etc.), disjunctive operators (triangular co-norms, like max, sum, etc.), average operators (averages as arithmetic average, quasi-linear average, etc) and hybrid operators (combinations of the above operators).
3. Crisp and fuzzy reasoning are consistently combined [10].

Fuzzy Prolog adds fuzziness to a Prolog compiler using CLP($\mathcal{R}$) instead of implementing a new fuzzy resolution method, as other former fuzzy Prologs do. It represents intervals as constraints over real numbers and *aggregation operators* as operations with these constraints, so it uses Prolog built-in inference mechanism to handle the concept of partial truth.

1.3 Motivation and *RFuzzy* Approach

Over the last few years several papers have been published by Medina et al. ([6,7,5]) about multi-adjoint programming, which describe a theoretical model, but no means of serious implementations apart from promising prototypes [1] and recently the FLOPER tool [9,8].

FLOPER implementation is inspired in Fuzzy Prolog [3] and adds the modellization of multi-adjoint logic. On one side Fuzzy Prolog is more expressive in the sense that can represent continuous fuzzy functions and its truth value is more general (union of intervals of real numbers), on the other side Fuzzy Prolog syntax is so flexible and general that can be complex for non-expert programmers just interested in modelling simple fuzzy problems.

This is the reason why we propose here the *RFuzzy* approach that is simpler than Fuzzy Prolog for the user because the truth values are simple real numbers instead of the general structures of Fuzzy Prolog. *RFuzzy* also models multi-adjoint logic and moreover provides some interesting improvements with respect to FLOPER: default values, partial default values (just for a subset of data), types for variables, and a useful sugar-syntax (for representing facts, rules and functions). Additionally *RFuzzy* inherits Fuzzy Prolog characteristics that are more expressive than other tools (uses Prolog-like syntax, has flexibility in the queries syntax, combines crisp and fuzzy predicates, uses general aggregation operators and provides constructive answers querying data and querying truth values).

Besides, *RFuzzy* implements multi-adjoint logic with a simple representation of the concept of credibility of the rules of multi-adjoint logic[1].

2 *RFuzzy* Expressiveness

RFuzzy enhances regular Prolog with truth values and with credibility values. In this section we enumerate and describe some of the most interesting characteristics of *RFuzzy* expressiveness through its syntax.

2.1 Types Definition

Prolog does not have types. It assigns values to the variables taking terms from the Herbrand Universe that can be created from the set of constants and constructors defined in a program. Nevertheless if we use types, then we can constraint the domain of values of the variables and this help us to return finite constructive answers (semantically correct). In *RFuzzy* types are defined according to (1) syntax.

$$\textbf{:- set\_prop}\ \textit{pred/ar} => \textit{type\_pred\_}1/1\ [,\ \textit{type\_pred\_n}/1\ ]^*\ . \qquad (1)$$

---

[1] A complete formalization of the semantics of *RFuzzy* with a description of a least model semantics, a least fixpoint semantics, an operational semantics and the proof of their equivalence can be downloaded at `http://babel.ls.fi.upm.es/software/rfuzzy/`

where *set_prop* is a reserved word, *pred* is the name of the typed predicate, *ar* is its arity and *type_pred_1*, *type_pred_n* ($n \in 2, 3, \ldots, ar$) are predicates used to define types for each argument of *pred*. They must have arity 1. The definition is constraining the values of the n_th argument of *pred* to the values of the type *type_pred_n*. This definition of types ensures that the values assigned to the arguments of *pred* are correctly typed.

The example below shows that the arguments of predicates *has_lower_price/2* and *expensive_car/1* have to be of type *car/1*. The domain of type *car* is enumerated.

$$: -set\_prop\ has\_lower\_price/2\ => \ car/1,\ car/1.$$

$$: -set\_prop\ expensive\_car/1\ => \ car/1.$$

$$car(vw\_caddy). \qquad car(alfa\_romeo\_gt).$$

$$car(aston\_martin\_bulldog). \qquad car(lamborghini\_urraco).$$

2.2 Simple Truth Value Assignment

It is possible to assign a truth value to an individual using fuzzy facts. Their syntax, that we can see in (2), is different than regular Prolog facts syntax.

$$pred(args)\ \textbf{value}\ truth\_val. \tag{2}$$

Arguments, *args*, should be ground and the truth value, *truth_val*, must be a real number between 0 and 1. The example below defines that the car *alfa_romeo_gt* is an *expensive_car* with a truth value 0.6.

$$expensive\_car(alfa\_romeo\_gt)\ value\ 0.6.$$

2.3 Continuous Function to Represent Truth Values

Facts definition (see subsection 2.2) is worth for a finite (and relative small) number of individuals. Nevertheless, it is very common to represent fuzzy truth using continuous functions. Fig. 1 shows an example in which the continuous function assigns the truth value of being *teenager* to each age.

Functions used to define the truth value of some group of individuals are usually continuous and linear over intervals. To define those functions there is no necessity to write down the value assigned to each element in their domains. We have to take into account that the domain can be infinite.

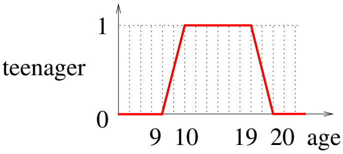


Fig. 1. Teenager truth value continuous representation

RFuzzy provides the syntax for defining functions by stretches. This syntax is shown in (3). External brackets represent the Prolog list symbols and internal brackets represent cardinality in the formula notation. Predicate *pred* has arity 1, *val1, ..., valN* should be ground terms representing numbers of the domain (they are possible values of the argument of *pred*) and *truth_val1, ..., truth_valN* should be the truth values associated to these numbers. The truth value of the rest of the elements is obtained by interpolation.

$$pred \; \mathbf{:\#} \;\; ([(val1, truth\_val1),\;(val2, truth\_val2)\;[,\;(valn, truth\_valn)\;]^{*}])\;. \tag{3}$$

The *RFuzzy* syntax for the predicate *teenager/1* (represented in Fig.1) is:

$$teenager : \#([(9,0),(10,1),(19,1),(20,0)]).$$

2.4 Rule Definition with Truth Values and Credibility

A tool which only allows the user to define truth values through functions and facts lacks on allowing him to combine those truth values for representing more complex situations. A rule is the tool to combine the truth values of facts, functions, and other rules.

Rules allow the user to combine truth values in the correct way (by means of aggregation operators, like *minimum*, *maximum*, *product*, etc.). The aggregation operator combines the truth values of the subgoals of the body of the rule to obtain the truth value of the head of the rule.

Appart from this, rules are assigned a credibility value to obtain the final truth value for the head of the clause. Credibility is used to express how much we trust a rule. It is used another opperator to aggregate the truth value obtained (from the aggregation of the subgoals of the body) with the rule's credibility.

RFuzzy offers a simple syntax for representing these rules, defined in (5). There are two aggregation operators, *op2* for combining the truth values of the subgoals of the rule body and *op1* for combining the previous result with the rule's credibility. The user can choose for any of them an aggregation operator from the list of the available ones[2] or define his/her own aggregation operator.

$$\begin{aligned} &pred(arg1\;[,\;argn]^{*}\;)\;[\;\mathbf{cred}\;\mathbf{(}op1, value1\mathbf{)}\;]\;\mathbf{:\sim}\;op2 \\ &\qquad\qquad\qquad\qquad pred1(args\_pred\_1)\;[,\;predm(args\_pred\_m)]\;. \end{aligned} \tag{4}$$

The following example uses the operator *prod* for aggregating truth values of the subgoals of the body and *min* to aggregate the result with the credibility of the rule (which is 0.8). "**cred (**$op1, value1$**)**" can only appear 0 or 1 times.

$$good\_player(J)\;cred(min, 0.8)\;\mathbf{:\sim}\;prod\;swift(J),\;tall(J),\;has\_experience(J).$$

---

[2] Aggregation operators available are: *min* for minimum, *max* for maximum, *prod* for the product, *luka* for the Łukasiewicz operator, *dprod* for the inverse product, *dluka* for the inverse Łukasiewicz operator and *complement*.

2.5 General and Conditioned Default Truth Values

Unfortunately, information provided by the user is not complete in general. So there are many cases in which we have no information about the truth value for a fuzzy predicate of an individual or a set of them. Nevertheless, it is interesting not to stop a complex query evaluation just because we have no information about one or more subgoals if we can use a reasonable approximation. A solution to this problem is using default truth values for these cases. The *RFuzzy* extension to define a default truth value for a predicate when applied to individuals for which the user has not defined an explicit truth value is named *general default truth value*. The syntax for defining a general default truth value is shown in (5).

Conditioned default truth value is used when the default truth value only applies to a subset of the domain. This subset is defined by a membership predicate which is true only when an individual belongs to the subset. The membership predicate (*membership_predicate/ar*) and the predicate to which it is applied (*pred/ar*) need to have the same arity (*ar*). The syntax is shown in (6).

$$\textbf{:- default(}pred/ar,\ truth\_value\textbf{)}\ \textbf{.} \tag{5}$$

$$\textbf{:- default(}pred/ar,\ truth\_value\textbf{)}\ \textbf{=>}\ membership\_predicate/ar\textbf{.} \tag{6}$$

pred/ar is in both cases the predicate to which we are defining default values. As expected, when defining the three cases (explicit, conditioned and default truth value) only one will be given back when doing a query. The precedence when looking for the truth value goes from the most concrete to the least one.

The code from the example below joint with the code from examples in subsections 2.1 and 2.2 assigns to the predicate *expensive_car* a truth value of 0.5 when the car is *vw_caddy* (default truth value), 0.9 when it is *lamborghini_urraco* or *aston_martin_bulldog* (conditioned default truth value) and 0.6 when it is *alfa_romeo_gt* (explicit truth value).

$$: -default(expensive\_car/1, 0.9) => expensive\_make/1.$$
$$: -default(expensive\_car/1, 0.5).$$
$$expensive\_make(lamborghini\_urraco).$$
$$expensive\_make(aston\_martin\_bulldog).$$

2.6 Constructive Answers

A very interesting characteristic for a fuzzy tool is being able to provide constructive answers for queries. The regular (easy) questions ask for the truth value of an element. For example, how expensive is an *Volkswagen Caddy*? (See left hand side example below)

$? - expensive\_car(vw\_caddy, V).$	$? - expensive\_car(X, V), V > 0.8.$
$V = 0.5?;$	$V = 0.9, X = aston\_martin\_bulldog?;$
no	$V = 0.9, X = lamborghini\_urraco?;$
	no

But the really interesting queries are the ones that ask for values that satisfy constraints over the truth value. For example, which cars are very expensive? (See right hand side example above). *RFuzzy* provides this constructive functionality.

3 Implementation Details

RFuzzy has to deal with two kinds of queries, (1) queries in which the user asks for the truth value of an individual, and (2) queries in which the user asks for an individual with a concrete or a restricted truth value. For this reason *RFuzzy* is implemented as a Ciao Prolog [12] package because Ciao Prolog offers the possibility of dealing with a higher order compilation through the implementation of Ciao packages.

The compilation process of a *RFuzzy* program has two pre-compilation steps: (1) the *RFuzzy* program is translated into CLP($\mathcal{R}$) constraints by means of the *RFuzzy* package and (2) the program with constraints is translated into ISO Prolog by using the CLP($\mathcal{R}$) package. Fig. 2 shows the whole process.

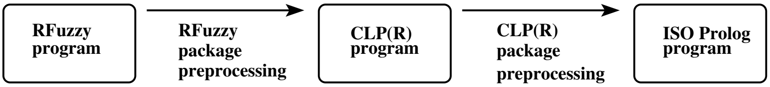


Fig. 2. *RFuzzy* architecture

4 Conclusions

RFuzzy is not a work in progress. It is an available implementation[3] that offers to the users/programmers a new framework to represent fuzzy problems over real numbers. Main *RFuzzy* advantages over *Fuzzy Prolog* are a simpler syntax and the elimination of answers with constraints. Moreover *RFuzzy* is one of the first tools modelling multi-adjoint logic, as explained in subsection 1.3. All the advanced characteristics of *RFuzzy* are missing in other tools as FLOPER[9,8].

Extensions added to *Prolog* by *RFuzzy* are: types (subsection 2.1), default truth values conditioned or general (subsection 2.5), assignment of truth values to individuals by means of facts (subsection 2.2), functions (subsection 2.3) or rules with credibility (subsection 2.4).

One of the most important consequences of these extensions is the constructivity of the answers with the possibility of constraining the truth value in the queries as we describe in section 2.6.

There are countless applications and research lines which can benefit from the advantages of using the fuzzy representations offered by *RFuzzy*. Some examples are: Search Engines, Knowledge Extraction (from databases, ontologies, etc.), Semantic Web, Business Rules, Coding Rules, etc.

---

[3] The *RFuzzy* module with installation instructions and examples can be downloaded from `http://babel.ls.fi.upm.es/software/rfuzzy/`

References

1. Abietar, J.M., Morcillo, P.J., Moreno, G.: Designing a software tool for fuzzy logic programming. In: Simos, T.E., Maroulis, G. (eds.) Proc. of the Int. Conf. of Computational Methods in Sciences and Engineering. ICCMSE 2007. Computation in Mordern Science and Engineering, vol. 2, pp. 1117–1120. American Institute of Physics (2007) (Distributed by Springer)
2. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint Logic Programming: syntax and semantics. In: ACM TOPLAS, vol. 23, pp. 1–29 (2001)
3. Guadarrama, S., Munoz-Hernandez, S., Vaucheret, C.: Fuzzy Prolog: A new approach using soft constraints propagation. Fuzzy Sets and Systems 144(1), 127–150 (2004)
4. Lee, R.C.T.: Fuzzy Logic and the resolution principle. Journal of the Association for Computing Machinery 19(1), 119–129 (1972)
5. Medina, J., Ojeda-Aciego, M., Votjas, P.: A completeness theorem for multi-adjoint Logic Programming. In: International Fuzzy Systems Conference, pp. 1031–1034. IEEE, Los Alamitos (2001)
6. Medina, J., Ojeda-Aciego, M., Votjas, P.: Multi-adjoint Logic Programming with continuous semantics. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS, vol. 2173, pp. 351–364. Springer, Heidelberg (2001)
7. Medina, J., Ojeda-Aciego, M., Votjas, P.: A procedural semantics for multi-adjoint Logic Programming. In: Brazdil, P.B., Jorge, A.M. (eds.) EPIA 2001. LNCS, vol. 2258, pp. 290–297. Springer, Heidelberg (2001)
8. Morcillo, P.J., Moreno, G.: Floper, a fuzzy logic programming environment for research. In: Proceedings of the Spanish Conference on Programming and Computer Languages, PROLE 2008, Gijón, Spain (2008)
9. Moreno, G.: Building a fuzzy transformation system. In: SOFtware SEMinar 2006: Theory and Practice of Computer Science, pp. 409–418 (2006)
10. Munoz-Hernandez, S., Vaucheret, C., Guadarrama, S.: Combining crisp and fuzzy Logic in a prolog compiler. In: Moreno-Navarro, J.J., Mariño, J. (eds.) Joint Conf. on Declarative Programming: APPIA-GULP-PRODE 2002, Madrid, Spain, pp. 23–38 (September 2002)
11. Shen, Z., Ding, L., Mukaidono, M.: Fuzzy resolution principle. In: Proc. of 18th International Symposium on Multiple-valued Logic, vol. 5 (1989)
12. The CLIP Lab. The Ciao Prolog Development System WWW Site, `http://www.clip.dia.fi.upm.es/Software/Ciao/`
13. Vaucheret, C., Guadarrama, S., Munoz-Hernandez, S.: Fuzzy prolog: A simple general implementation using clp(r). In: Baaz, M., Voronkov, A. (eds.) LPAR 2002. LNCS (LNAI), vol. 2514, pp. 450–463. Springer, Heidelberg (2002)
14. Vojtas, P.: Fuzzy logic programming. Fuzzy Sets and Systems 124(1), 361–370 (2001)
15. Ehud, Y., Shapiro: Logic programs with uncertainties: A tool for implementing rule-based systems. In: International Joint Conference on Artificial Intelligence, pp. 529–532 (1983)

Overcoming Non-commutativity in Multi-adjoint Concept Lattices

Jesús Medina*

Dept. Matemáticas, Universidad de Cádiz, Spain
jesus.medina@uca.es

Abstract. Formal concept analysis has become an important and appealing research topic. In this paper, we present the t-concept lattice as a set of triples associated to graded tabular information interpreted in a non-commutative fuzzy logic, in order to "soften" the non-commutativity character. Moreover, we show that the common information to both (sided) concept lattices can be seen as a sublattice of the Cartesian product of both concept lattices.

1 Introduction

Data analysis is one of the more important stages in artificial intelligence. In this topic, Wille introduced in 1982 the formal concept analysis [17]. Regarding applications, we can find papers ranging from ontology merging [7,15], to applications to the Semantic Web by using the notion of concept similarity [8], and from processing of medical records in the clinical domain [10] to the development of recommender systems [6].

On the fuzzy extensions of formal concept analysis a number of different approaches have been presented. To the best of our knowledge, the first one was given in [5], although they did not advance much beyond the basic definitions, probably due to the fact that they did not use residuated implications. Later, in [16,2] the authors independently used complete residuated lattices as structures for the truth degrees; for this approach, a representation theorem was proved directly in a fuzzy framework in [4], setting the basis of most of the subsequent direct proofs.

In recent years there has been an increased interest in studying formal concept analysis on the perspective of using non-commutative conjunctors. This is not a mere mathematical generalization, but a real need since, for instance, when one learns a conjunction from examples it is not unusual that the resulting conjunction does not satisfy commutativity. Different authors have argued in favour of considering non-commutative conjunctors. Actually, there exist quite reasonable examples of non-commutative and even non-associative conjunctors defined on a regular partition of the unit interval, see [13]. Multi-adjoint concept lattices were introduced [13] as a new general approach to formal concept analysis, with the

---

* Partially supported by the Spanish Science Ministry under grant TIN 2006-15455-C03-01 and by Junta de Andalucía under grant P06-FQM-02049.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 278–285, 2009.
© Springer-Verlag Berlin Heidelberg 2009

idea of providing a general framework in which the different approaches stated above could be conveniently accommodated, the authors worked in a general non-commutative environment.

Based on the new kinds of concept lattices, developed from the use of non-commutative logic and similarity [9], and following the general techniques of formal concept analysis, given a non-commutative conjunctor, it is possible to provide generalizations of the mappings for the intension and the extension in two different ways, generating a pair of concept lattices. In this paper, we show that the common information to both sided concept lattices can be seen as a sublattice of the Cartesian product of both concept lattices. In some sense, this common information may be thought of as "neutral" information with regard to the non-commutativity character of the conjunctor and, in some sense, allows to draw conclusions which somehow "soften" the non-commutativity of the conjunctor. The resulting theory allows for obtaining a simpler proof of a general representation theorem for t-concepts, which can be easily instantiated to obtain the representation theorem in [9].

2 Multi-adjoint Concept Lattices

In order to make this paper self-contained, we provide in this section the more important definitions and results from [13]. The first definition introduces the basic building blocks of the multi-adjoint concept lattices, the *adjoint triples*, which are generalisations of the notion of adjoint pair under the hypothesis of having a non-commutative conjunctor.

Definition 1. *Let* $(P_1, \leq_1)$, $(P_2, \leq_2)$, $(P_3, \leq_3)$ *be posets and* $\& \colon P_1 \times P_2 \to P_3$, $\swarrow \colon P_3 \times P_2 \to P_1$, $\nwarrow \colon P_3 \times P_1 \to P_2$ *be mappings, then* $(\&, \swarrow, \nwarrow)$ *is an* adjoint triple *with respect to* P_1, P_2, P_3 *if:*

1. $\&$ *is order-preserving in both arguments.*
2. $\swarrow$ *and* $\nwarrow$ *are order-preserving in the consequent and order-reversing in the antecedent.*
3. $x \leq_1 z \swarrow y$ *iff* $x \,\&\, y \leq_3 z$ *iff* $y \leq_2 z \nwarrow x$, *where* $x \in P_1$, $y \in P_2$ *and* $z \in P_3$.

Note that in the domain and codomain of the considered conjunctor we have three (in principle) different sorts, thus providing a more flexible language to a potential user. Furthermore, notice that no boundary condition is required, in difference to the usual definition of multi-adjoint lattice [14] or implication triple [1]. Nevertheless, some boundary conditions follow from the definition, specifically, from the adjoint property (condition (3) above) [13].

In order to provide more flexibility into our language, we will allow the existence of several adjoint triples for a given triplet of posets.

Definition 2. *A* multi-adjoint frame $\mathcal{L}$ *is a tuple*

$$(L_1, L_2, P, \preceq_1, \preceq_2, \leq, \&_1, \swarrow^1, \nwarrow_1, \dots, \&_n, \swarrow^n, \nwarrow_n)$$

where $(L_1, \preceq_1)$ *and* $(L_2, \preceq_2)$ *are complete lattices,* $(P, \leq)$ *is a poset and, for all* $i = 1, \dots, n$, $(\&_i, \swarrow^i, \nwarrow_i)$ *is an adjoint triple with respect to* L_1, L_2, P.

For short, a multi-adjoint frame will be denoted as $(L_1, L_2, P, \&_1, \dots, \&_n)$.

Definition 3. *Given a multi-adjoint frame* $(L_1, L_2, P, \&_1, \dots, \&_n)$*, a* context *is a tuple* (A, B, R, σ) *such that* A *and* B *are non-empty sets (usually interpreted as attributes and objects, respectively),* R *is a* P*-fuzzy relation* $R\colon A \times B \to P$ *and* $\sigma\colon B \to \{1, \dots, n\}$ *is a mapping which associates any element in* B *with some particular adjoint triple in the frame.*[1]

The function σ above, which assigns an adjoint triple to each object (or attribute), is important in that it allows for defining subgroups of objects or attributes in terms of different degrees of preference, see [13].

The following mappings ${}^{\uparrow_\sigma}\colon L_2^B \longrightarrow L_1^A$ and ${}^{\downarrow^\sigma}\colon L_1^A \longrightarrow L_2^B$ can be seen as generalisations of those given in [3,11]:

$$g^{\uparrow_\sigma}(a) = \inf\{R(a,b) \swarrow^{\sigma(b)} g(b) \mid b \in B\} \tag{1}$$

$$f^{\downarrow^\sigma}(b) = \inf\{R(a,b) \nwarrow_{\sigma(b)} f(a) \mid a \in A\} \tag{2}$$

These two arrows, $({}^{\uparrow_\sigma}, {}^{\downarrow^\sigma})$, generate a Galois connection [13].

As usual in the different frameworks of formal concept analysis, a *multi-adjoint concept* is a pair $\langle g, f\rangle$ satisfying that $g \in L_2^B$, $f \in L_1^A$ and that $g^{\uparrow_\sigma} = f$ and $f^{\downarrow^\sigma} = g$; with $({}^{\uparrow_\sigma}, {}^{\downarrow^\sigma})$ being the Galois connection defined above.

Definition 4. *The* multi-adjoint concept lattice *associated to a multi-adjoint frame* $(L_1, L_2, P, \&_1, \dots, \&_n)$ *and a context* (A, B, R, σ) *is the set*

$$\mathcal{M} = \{\langle g, f\rangle \mid g \in L_2^B, f \in L_1^A \text{ and } g^{\uparrow_\sigma} = f, f^{\downarrow^\sigma} = g\}$$

in which the ordering is defined by $\langle g_1, f_1\rangle \preceq \langle g_2, f_2\rangle$ *if and only if* $g_1 \preceq_2 g_2$ *(equivalently* $f_2 \preceq_1 f_1$*).*

The ordering just defined above actually provides $\mathcal{M}$ with the structure of a complete lattice [13].

From now on, we will fix a multi-adjoint frame $(L_1, L_2, P, \&_1, \dots, \&_n)$ and context (A, B, R, σ). Moreover, to improve readability, we will write $({}^\uparrow, {}^\downarrow)$ instead of $({}^{\uparrow_\sigma}, {}^{\downarrow^\sigma})$ and $\swarrow^b$, $\nwarrow_b$ instead of $\swarrow^{\sigma(b)}$, $\nwarrow_{\sigma(b)}$.

Definition 5. *Given a complete lattice* L*, a subset* $K \subseteq L$ *is* infimum-dense *(resp.* supremum-dense*) if and only if for all* $x \in L$ *there exists* $K' \subseteq K$ *such that* $x = \inf(K')$ *(resp.* $x = \sup(K')$*).*

A multi-adjoint concept lattice is said to be represented by a complete lattice provided there is a pair of functions, α and β, satisfying the conditions stated in the definition below:

---

[1] A similar theory could be developed by considering a mapping $\tau\colon A \to \{1, \dots, n\}$ which associates any element in A with some particular adjoint triple in the frame.

Definition 6. *A multi-adjoint concept lattice*[2] $(\mathcal{M}, \preceq)$ *is* represented *by a complete lattice* $(V, \sqsubseteq)$ *if there exists a pair of mappings* $\alpha\colon A \times L_1 \to V$ *and* $\beta\colon B \times L_2 \to V$ *such that:*

1a) $\alpha[A \times L_1]$ *is infimum-dense;*
1b) $\beta[B \times L_2]$ *is supremum-dense; and*
2) For all $a \in A$, $b \in B$, $x \in L_1$, $y \in L_2$*:*

$$\beta(b, y) \sqsubseteq \alpha(a, x) \quad \textit{if and only if} \quad x \mathbin{\&_b} y \leq R(a, b)$$

From the definition of representability the following properties follow:

Proposition 1. *Given a complete lattice* $(V, \sqsubseteq)$ *which represents a multi-adjoint concept lattice* $(\mathcal{M}, \preceq)$*, and mappings* $f \in L_1^A$ *and* $g \in L_2^B$*, we have:*

1. β *is order-preserving in the second argument.*
2. α *is order-reversing in the second argument.*
3. $g^{\uparrow}(a) = \sup\{x \in L_1 \mid v_g \sqsubseteq \alpha(a, x)\}$*, where* $v_g = \sup\{\beta(b, g(b)) \mid b \in B\}$.
4. $f^{\downarrow}(b) = \sup\{y \in L_2 \mid \beta(b, y) \sqsubseteq v_f\}$*, where* $v_f = \inf\{\alpha(a, f(a)) \mid a \in A\}$.
5. *If* $g_v(b) = \sup\{y \in L_2 \mid \beta(b, y) \sqsubseteq v\}$*, then* $\sup\{\beta(b, g_v(b)) \mid b \in B\} = v$.
6. *If* $f_v(a) = \sup\{x \in L_1 \mid v \sqsubseteq \alpha(a, x)\}$*, then* $\sup\{\alpha(a, f_v(a)) \mid a \in A\} = v$.

The fundamental theorem for multi-adjoint concept lattices presented in [13] is the following.

Theorem 1. *A complete lattice* $(V, \sqsubseteq)$ *represents a multi-adjoint concept lattice* $(\mathcal{M}, \preceq)$ *if and only if* $(V, \sqsubseteq)$ *is isomorphic to* $(\mathcal{M}, \preceq)$.

Some new interesting properties about the mappings α and β can be extract from Definition 6. So, let us assume a complete lattice $(V, \sqsubseteq)$ which represents a multi-adjoint concept lattice $(\mathcal{M}, \preceq)$ and the mappings $\alpha\colon A \times L_1 \to V$, $\beta\colon B \times L_2 \to V$.

We will restate below the isomorphism constructed in fundamental theorem, based on both the α and β functions, since these expressions will be used later.

Proposition 2 ([13]). *If a complete lattice* $(V, \sqsubseteq)$ *represents a multi-adjoint concept lattice* $(\mathcal{M}, \preceq)$*, then there exists an isomorphism* $\varphi\colon \mathcal{M} \to V$ *and two mappings* $\beta\colon B \times L_2 \to V$, $\alpha\colon A \times L_1 \to V$*, such that:*

$$\varphi(\langle g, f\rangle) = \sup\{\beta(b, g(b)) \mid b \in B\} = \inf\{\alpha(a, f(a)) \mid a \in A\}$$

for all concept $\langle g, f\rangle \in \mathcal{M}$.

The following result shows continuity-related properties of α and β.

Proposition 3. *The applications* $\beta\colon B \times L_2 \to V$, $\alpha\colon A \times L_1 \to V$ *satisfy that:*

1. $\beta(b, \sup Y) = \sup\{\beta(b, y) \mid y \in Y\}$*, for all* $Y \subseteq L_2$ *and* $b \in B$.

2. $\alpha(a, \sup X) = \inf\{\alpha(a, x) \mid x \in X\}$*, for all* $X \subseteq L_1$ *and* $a \in A$.

---

[2] Recall that we are considering a multi-adjoint concept lattice on a fixed frame $(L_1, L_2, P, \&_1, \ldots, \&_n)$ and context (A, B, R, σ).

Finally, the following property shows that any subset of $A \times L_1$ or of $B \times L_2$ is related to a concept via α and φ, or β and φ, respectively.

Proposition 4. *Given a multi-adjoint concept lattice* $(\mathcal{M}, \preceq)$ *represented by a complete lattice* $(V, \sqsubseteq)$ *and the mappings* $\alpha\colon A \times L_1 \to V$, $\beta\colon B \times L_2 \to V$, *we have that for each* $K \subseteq A \times L_1$, *there exists a unique concept* $\langle g, f\rangle \in \mathcal{M}$ *such that*

$$\inf\{\alpha(a, x) \mid (a, x) \in K\} = \varphi(\langle g, f\rangle)$$

Analogously, for each $K' \subseteq B \times L_2$, *there exists a unique concept* $\langle g, f\rangle \in \mathcal{M}$ *such that* $\sup\{\beta(b, y) \mid (b, y) \in K'\} = \varphi(\langle g, f\rangle)$.

3 Multi-adjoint t-Concept Lattice

In this section a new construction based on the previous notion of multi-adjoint concept lattice is presented. The t-concepts are introduced as a generalisation of the approach given in [9] for non-commutative conjunctors, which provides greater flexibility and, hence, allows for specifying and solving a greater number of problems in more complex knowledge-based systems.

The basic structure we will work with is that of multi-adjoint frame, where the complete lattices $\langle L_1, \preceq_1\rangle, \langle L_2, \preceq_2\rangle$ coincide, and we will denote $\langle L, \preceq\rangle$. This way, given a context (A, B, R, σ), besides the Galois connection $({}^{\uparrow}, {}^{\downarrow})$ defined for the multi-adjoint concept lattice, it is possible to define an alternative version as follows:

$$g^{\uparrow_{op}}(a) = \inf\{R(a, b) \nwarrow_b g(b) \mid b \in B\}$$
$$f^{\downarrow^{op}}(b) = \inf\{R(a, b) \swarrow^b f(a) \mid a \in A\}$$

We can easily check that the definition above is actually a Galois connection, since it coincides with the Galois connection defined by equations (1), (2), on the multi-adjoint frame $(L, L, P, \&_1^{op}, \dots, \&_n^{op})$ and context (A, B, R, σ), being $\&_i^{op}\colon L \times L \to P$ and $x \,\&_i^{op}\, y = y \,\&_i\, x$ for all $i \in \{1, \dots, n\}$. Since the implications are permuted, if the initial adjoint triples are $(\&_i, \nwarrow_i, \swarrow^i)$, then the adjoint triples considered are $(\&_i^{op}, \swarrow^i, \nwarrow_i)$. Now, we have two Galois connections $({}^{\uparrow}, {}^{\downarrow})$, $({}^{\uparrow_{op}}, {}^{\downarrow_{op}})$, on which two different multi-adjoint concept lattices $(\mathcal{M}, \preceq)$, $(\mathcal{M}^{op}, \preceq)$ can be defined.[3]

Both lattices are different if at least one conjunctor $\&_i$ is non-commutative, but are certainly related. This suggests to consider the following subsets of $\mathcal{M} \times \mathcal{M}^{op}$:

$$\mathcal{N}_1 = \{(\langle g, f_1\rangle, \langle g, f_2\rangle) \mid \langle g, f_1\rangle \in \mathcal{M}, \langle g, f_2\rangle \in \mathcal{M}^{op}\}$$
$$\mathcal{N}_2 = \{(\langle g_1, f\rangle, \langle g_2, f\rangle) \mid \langle g_1, f\rangle \in \mathcal{M}, \langle g_2, f\rangle \in \mathcal{M}^{op}\}$$

---

[3] Note that the ordering relation is the same for both lattices, although its domain might differ from one to another.

which, together with the orderings

$$\begin{aligned}
(\langle g, f_1\rangle, \langle g, f_2\rangle) \preceq (\langle g', f_1'\rangle, \langle g', f_2'\rangle) \qquad &\text{if and only if} \qquad g \preceq g' \\
(\langle g_1, f\rangle, \langle g_2, f\rangle) \preceq (\langle g_1', f'\rangle, \langle g_2', f'\rangle) \qquad &\text{if and only if} \qquad f' \preceq f
\end{aligned}$$

are sublattices of $\mathcal{M} \times \mathcal{M}^{op}$ and, thus, are complete lattices.

Now we will show that Georgescu and Popescu's framework can be reproduced by means of the multi-adjoint framework. In [12] it has been proved that the concept lattices defined in [9] can be constructed in terms of generalized concept lattices [11]. On the other hand, in [13] we proved that generalized concept lattices can be embedded into the multi-adjoint framework. As a result, we can obtain that every non-commutative fuzzy concept lattice $\mathcal{L}$, in the sense of Georgescu and Popescu, can be embedded into a specific product, $\mathcal{M} \times \mathcal{M}^{op}$, of multi-adjoint concept lattices. Specifically, the theorem below shows that, there exists a particular choice of multi-adjoint frame and context such that $\mathcal{L}$ is isomorphic to the sublattice $\mathcal{N}_1$ of $\mathcal{M} \times \mathcal{M}^{op}$.

Theorem 2. *Given a complete generalized residuated lattice* $(L, \preceq, \&, \swarrow, \nwarrow)$ *and a residuated context* (A, B, R)*, then there exists a multi-adjoint concept lattice,* $\mathcal{M}$*, such that the sublattice* $\mathcal{N}_1$ *is isomorphic to the non-commutative fuzzy concept lattice* $\mathcal{L}$*.*

Proof. Follows by from [12, theorem 6] and [13, theorem 14]. □

The theorem above justifies considering the general construction of $\mathcal{N}_1$ as a generalized concept lattice. However, it is important to note that the theorem above shows a strict embedding of the framework by Georgescu-Popescu into our framework. Their construction explicitly assumes that $(L, \&, \top)$ should be a commutative monoid, but $\mathcal{N}_1$ can be defined as well by directly considering an adjoint triple which, obviously needs not be either commutative nor associative.

The particular form of the elements of $\mathcal{N}_1$ suggests to abuse a little bit the notation, denote them as $\langle g, f_1, f_2\rangle$, and use the term *t-concept* to refer to them (t- for triple). Note that we will concentrate hereafter on the lattice $\mathcal{N}_1$, but similar results can be obtained for $\mathcal{N}_2$.

4 Representation Theorem of the Lattice of t-Concepts $\mathcal{N}_1$

We start this section by introducing some preliminary definitions and results needed for the statement and proof of the representation theorem.

Definition 7. *Given a set* A*, a poset* P *with bottom element* $\bot$*, and elements* $a \in A$*,* $x \in P$ *, the characteristic mapping* $@_a^x \colon A \to P$*, read "at point* a *the value is* x*", is defined as:*

$$@_a^x(a') = \begin{cases} x, & \textit{if } a' = a \\ \bot, & \textit{otherwise} \end{cases}$$

Definition 8. *Given two complete lattices* $(V_1, \sqsubseteq_1)$, $(V_2, \sqsubseteq_2)$ *which represent the multi-adjoint concept lattices* $(\mathcal{M}, \preceq)$, $(\mathcal{M}^{op}, \preceq)$, *respectively, the sublattice* $\mathcal{V}_{12}$ *of* $V_1 \times V_2$ *is defined as:*

$$\mathcal{V}_{12} = \left\{ \left(\inf_{(a,x)\in K_1} \alpha_1(a,x), \inf_{(a,x)\in K_2} \alpha_2(a,x) \right) \mid (K_1, K_2) \in \mathcal{K} \right\}$$

where $\mathcal{K} = \{(K_1, K_2) \mid K_1, K_2 \subseteq A \times L$ *and* $\inf_{(a,x)\in K_1} @_a^{x\downarrow} = \inf_{(a,x)\in K_2} @_a^{x\downarrow^{op}}\}$ *and* α_1, α_2 *are the maps given in Definition 6 associated to* $(\mathcal{M}, \preceq)$, $(\mathcal{M}^{op}, \preceq)$ *respectively.*

The following result proves that the sublattice above is isomorphic to the complete sublattice of t-concepts $\mathcal{N}_1$ of $\mathcal{M} \times \mathcal{M}^{op}$:

Proposition 5. *Let* $(V_1, \sqsubseteq_1)$, $(V_2, \sqsubseteq_2)$ *be the complete lattices which represent two multi-adjoint concept lattices* $(\mathcal{M}, \preceq_1)$, $(\mathcal{M}^{op}, \preceq_2)$ *respectively, then* $\mathcal{N}_1$ *is isomorphic to the sublattice* $\mathcal{V}_{12}$ *of* $V_1 \times V_2$.

We can now introduce the representation theorem to the multi-adjoint t-concept lattice $\mathcal{N}_1$ as a generalization of that by Georgescu and Popescu.

Theorem 3. *A lattice* $(V, \sqsubseteq)$ *is isomorphic to a complete lattice of t-concepts* $(\mathcal{N}_1, \preceq)$ *if and only if there exist two complete lattices* $(V_1, \sqsubseteq_1)$ *and* $(V_2, \sqsubseteq_2)$ *such that they represent the multi-adjoint concept lattices* $(\mathcal{M}, \preceq)$, $(\mathcal{M}^{op}, \preceq)$, *and there exists an isomorphism* ν *from* V *to the sublattice* $\mathcal{V}_{12}$ *of* $V_1 \times V_2$.

As a consequence of the previous results, we can obtain the representation theorem of the framework presented in [9].

Corollary 1. *Let* $I\colon B \times A \to P$ *be a relation. A lattice* $(V, \sqsubseteq)$ *is isomorphic to* $\mathcal{L}$ *if and only if there exist two complete lattices* $(V_1, \sqsubseteq_1)$ *and* $(V_2, \sqsubseteq_2)$ *and five applications:*

$$\begin{aligned} &\alpha_1\colon A \times L \to V_1; \quad \beta_1\colon B \times L \to V_1; \quad \nu\colon V \to V_1 \times V_2 \\ &\alpha_2\colon A \times L \to V_2; \quad \beta_2\colon B \times L \to V_2; \end{aligned}$$

such that:

1. $\alpha_1[A \times L]$ *is infimum dense in* V_1 *and* $\alpha_2[A \times L]$ *is infimum dense in* V_2.
2. $\beta_1[B \times L]$ *is supremum-dense in* V_1 *and* $\beta_2[B \times L]$ *is supremum-dense in* V_2.
3. *For each* $a \in A$, $b \in B$, $x, y \in L$:

$$\begin{aligned} \beta_1(b,y) \sqsubseteq_1 \alpha_1(a,x) \quad &\textit{iff} \quad x \,\&\, y \preceq I(b,a) \\ \beta_2(b,y) \sqsubseteq_2 \alpha_2(a,x) \quad &\textit{iff} \quad x \,\&^{op}\, y \preceq I(b,a) \end{aligned}$$

4. ν *is a join-preserving monomorphism from* V *onto* $V_1 \times V_2$ *such that, for any* $v \in V$, *there exist* $K_1, K_2 \subseteq A \times L$ *satisfying that* $\inf_{(a,x)\in K_1} @_a^{x\downarrow} = \inf_{(a,x)\in K_2} @_a^{x\downarrow^{op}}$ *and such that* $\nu(v)$ *is equal to the pair:*

$$\big(\inf_{(a,x)\in K_1} (\sup_{b\in B} \beta_1(b, I(b,a) \nwarrow x)), \inf_{(a,x)\in K_2} (\sup_{b\in B} \beta_2(b, I(b,a) \nwarrow x)) \big)$$

5 Conclusions

We showed that the common information to the two sided concept lattices generated from the two possible residual implications associated to a non-commutative conjunctor, can be seen as a sublattice of the Cartesian product of both concept lattices. Such common information can be thought of as "neutral" information with regard to the non-commutativity character of the conjunctor and, in some sense, allows to draw conclusions which somehow "soften" the non-commutativity of the conjunctor.

The resulting theory allows for obtaining a simpler proof of a general representation theorem for t-concepts, which can be easily instantiated to obtain the representation theorem in [9].

References

1. Abdel-Hamid, A., Morsi, N.: Associatively tied implicacions. Fuzzy Sets and Systems 136(3), 291–311 (2003)
2. Belohlávek, R.: Lattice generated by binary fuzzy relations (extended abstract). In: 4th Intl. Conf. on Fuzzy Sets Theory and Applications, p. 11 (1998)
3. Belohlávek, R.: Concept lattices and order in fuzzy logic. Annals of Pure and Applied Logic 128, 277–298 (2004)
4. Belohlávek, R.: Lattices of fixed points of fuzzy Galois connections. Mathematical Logic Quartely 47(1), 111–116 (2004)
5. Burusco, A., Fuentes-González, R.: The study of L-fuzzy concept lattice. Mathware & Soft Computing 3, 209–218 (1994)
6. du Boucher-Ryana, P., Bridge, D.: Collaborative recommending using formal concept analysis. Knowledge-Based Systems 19(5), 309–315 (2006)
7. Formica, A.: Ontology-based concept similarity in formal concept analysis. Information Sciences 176(18), 2624–2641 (2006)
8. Formica, A.: Concept similarity in formal concept analysis: An information content approach. Knowledge-Based Systems 21(1), 80–87 (2008)
9. Georgescu, G., Popescu, A.: Concept lattices and similarity in non-commutative fuzzy logic. Fundamenta Informaticae 53(1), 23–54 (2002)
10. Jiang, G., Ogasawara, K., Endoh, A., Sakurai, T.: Context-based ontology building support in clinical domains using formal concept analysis. International Journal of Medical Informatics 71(1), 71–81 (2003)
11. Krajci, S.: A generalized concept lattice. Logic Journal of IGPL 13(5), 543–550 (2005)
12. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: Relating generalized concept lattices with concept lattices for non-commutative conjunctors. Applied Mathematics Letters 21(12), 1296–1300 (2008)
13. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: Formal concept analysis via multi-adjoint concept lattices. Fuzzy Sets and Systems 160(2), 130–144 (2009)
14. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: Similarity-based unification: a multi-adjoint approach. Fuzzy Sets and Systems 146, 43–62 (2004)
15. Phan-Luong, V.: A framework for integrating information sources under lattice structure. Information Fusion 9, 278–292 (2008)
16. Pollandt, S.: Fuzzy Begriffe. Springer, Berlin (1997)
17. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel (1982)

Evolutionary Fuzzy Scheduler for Grid Computing

R.P. Prado, S. García Galán, A.J. Yuste, J.E. Muñoz Expósito,
A.J. Sánchez Santiago, and S. Bruque

Telecommunication Engineering Department. University of Jaén
Alfonso X el Sabio, 28 Linares, Jaén. Spain
{rperez,sgalan,ajyuste,jemunoz,ajsantia,sbruque}@ujaen.es

Abstract. In the last few years, the Grid community has been growing very rapidly and many new components have been proposed. In this sense, the scheduler represents a very relevant element that influences decisively on the grid system performance. The scheduling task of a set of heterogeneous, dynamically changing resources is a complex problem. Several scheduling systems have already been implemented; however, they still provide only "ad hoc" solutions to manage scheduling resources in a grid system. This paper presents a fuzzy scheduler obtained by means of evolving a previous fuzzy scheduler using Pittsburgh approach. This new evolutionary fuzzy scheduler improves the performance of the classical scheduling system.

Keywords: Genetic Fuzzy Systems, Grid Computing, Automatic Learning.

1 Introduction

A grid is a large scale, heterogeneous collection of autonomous systems, geographically distributed and interconnected with high speed networks [1]. Computing resources are geographically distributed under different ownerships each having their own access policy, cost and other various constraints; basically, a grid system is a complex set of shared heterogeneous resources made available by different providers.

Grid computing allows us to use these resources to solve large scale computational problems. Some of these applications belong to different fields such as medical image analysis [2] or bio-informatics [3]. A grid system offers several advantages [4] such as:

- Ability to build dynamic applications that use distributed resources in order to improve a fitness function.
- Utilization of resources that are located in a particular domain to increase throughput or reduce execution costs.
- Execution of spanning multiple administrative domains to obtain specific processing capabilities.
- Integration of multiple teams involved in managing different parts of the experiment workflow-thus promoting inter-organizational collaborations.

Despite of its advantages, Grid Computing also entails some new challenges, ranging from adaptation of parallel programs previously developed for homogeneous resources clusters to handling dynamic and heterogeneous grid resources with

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 286–293, 2009.
© Springer-Verlag Berlin Heidelberg 2009

minimal intrusion into the code or resource scheduling. In a grid environment, the scheduler is the manager of the workflow, acting as an intermediary between the user and the distributed resources, thus concealing the complexity of the grid system [5]. A grid system scheduler provides the functionality for resources discovery and publishing and also it undertakes scheduling, submission and monitoring jobs. The job scheduling problem is known to be NP-complete [6]. A scheduler is designed to satisfy one or more of the following common objectives: maximizing system throughput, maximizing resource utilizations, maximizing economic gains or minimizing the turn-around time for a given application.

In this work we improve the overall response time for the entire workflow. In order to improve the scheduler performance, we use an evolutionary fuzzy system [7] whose rules have been obtained using a genetic learning process based on Pittsburgh approach [7][8]. According to this approach, an entire fuzzy rule base is encoded as a chromosome, shaping one of the individuals of the candidate population. The experimental results obtained with our proposed scheduler are compared with the results of other classical schedulers (Round Robin-Random, Random-Random) and other results obtained with a scheduler based on a fuzzy system which has not evolving system.

The paper is organized as follows. In section 2 we analyze recent works in grid resource scheduling. Section 3 presents the structure of the proposed evolutionary fuzzy scheduler. Section 4 describes the approach used to obtain the knowledge base of the evolutionary fuzzy scheduler. In section 5 we show the simulation results, and finally we present some conclusions and future work in section 6.

2 Background

A grid system is a complex and heterogeneous entity that uses several sites and organizations. The grid structure or Virtual Organization (VO), Fig. 1, is made up by a set of Physical Organizations (PO). A VO comprises a set of independent organizations that share resources to achieve a common goal, but that are not limited to an alliance among firms [9]. Interaction among members of the virtual organization is mainly done through computer networks.

A scheduler must use available resources in the most efficient way in order to reduce the execution time of a given set of tasks. In Fig. 1, two kinds of schedulers appear: a metascheduler, placed in a Grid Resource Center that decides the PO to which the task should be sent to, and a local scheduler that decides the resource that should execute the task inside the PO.

In addition, when concerning heterogeneous environments such as grid, it must be underlined the fact that each resource computational capacity may differs and also that they are exposed to changes at any time, so a self-adapting, flexible and computationally inexpensive scheduler is relevant in order to use all the available resources in a more efficient way. Therefore, scheduling techniques can be broadly divided into two categories: static [10] and dynamic [11]. A static algorithm assigns all properties at design time and assumes that parameters of all the tasks are known a priori and builds a scheduler based on these assumptions, i.e. round-robin, random dispatching or easy backfilling [12], that requires a runtime estimation for each job provided by the user.

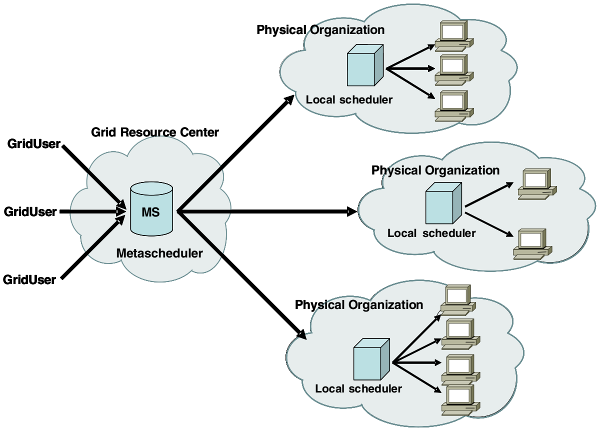


Fig. 1. Structure of a generic grid system

A dynamic algorithm assigns properties at runtime, based on execution parameters of tasks. A dynamic job scheduling scheme is more appropriate for a grid environment because it dispatches jobs according to the current workload status of each node. Fuzzy scheduling models are dynamic and have recently attracted a great interest among the scheduling research community as the number of papers published in journals and conference proceedings is continuously increasing.

Some works use fuzzy decision rules in task dispatching [13]. Other works use fuzzy constraints like CPU's utilization memory utilization [14], or due dates, resources, and processing time [15]. In [16], a scheduler based on fuzzy association rule mining was built to optimize the overall schedule process. Fuzzy sets could be used in scheduling problems in different ways, i.e. to develop algorithms in many fields like security [17] which allow supporting certainty in a grid, controlling the number of repeated tasks. In this paper we apply fuzzy logic to build a metascheduler which uses dynamic inputs for the fuzzy system: number of free resources in a PO (NFR), number of idle MIPS in a PO, free memory (FM) in a PO and size of executed tasks (STE) per PO.

3 Proposed Fuzzy System Structure

There may be many inputs for a fuzzy logic scheduler (i.e. memory utilization, CPU utilization, network state, costs, tasks to be executed, etc.), we use four inputs that can be considered the most relevant ones. These inputs may be categorized into machine utilization and load balancing.

Machines utilization contains three inputs:

- Number of free resources in a PO (NFR): this input indicates the number of resources in the PO having a free processor to execute a task. If the number of available resources is low, the network is able to choose a less congested PO in order to have better conditions to carry out our tasks.

- Number of idle MIPS: it indicates the number of free MIPS in each PO; the more idle MIPS, the more capacity this PO has to execute our tasks.
- Free memory (FM): it is the total free memory of all the resources in a PO.

Load balancing contains one input:

- Size of executed tasks (STE): this input indicates the sum of the lengths of the tasks that have been executed in this PO.

Through these four inputs we are able to control the system, given that these inputs are those that can influence overall simulation time. With this initial information, the fuzzy system will decide which PO should execute a given task. Then the local scheduler of the selected PO will send the task to the most suitable free resource.

These four fuzzy variables are defined by a triangle shaped fuzzy set since they have been extensively used in real time applications due to their simple formulas and computational efficiency. The output fuzzy variable of the system is defined as a PO selection factor.

The membership functions for input and output fuzzy variables are represented in Fig. 2. The structure of the proposed fuzzy scheduler is shown in Fig. 3.

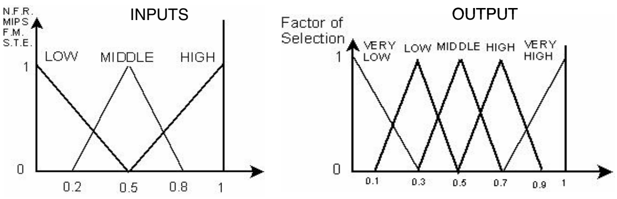


Fig. 2. Membership functions for input and output variables

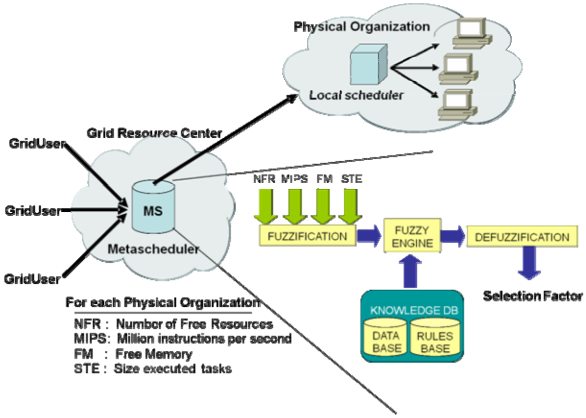


Fig. 3. Proposed fuzzy scheduler structure

4 Evolutionary Fuzzy Scheduler Obtained

The fuzzy system performance is strongly related to the knowledge base quality, and consequently to the process of knowledge acquisition. In some cases, this process does not take place because there are no available experts in the field of a certain problem. It is also possible that the acquired knowledge only represents an incomplete or partially correct fuzzy system description. When the obtained knowledge is not considered good enough to be used, some kind of learning is needed. Therefore, the automatic definition of Fuzzy Rules-Based-Systems (FRBSs) can be considered in many cases as an optimization or search process. Genetic Algorithms (GA) are known to be capable of finding near optimal solutions in complex search spaces. In this work, the new rules added to the FRBS knowledge base have been obtained using evolutionary computation based on GAs. The genetic learning process used in this work to evolve a FRBS is the Pittsburgh algorithm [7][8]. In Pittsburgh approach, each chromosome represents an entire base of rules and evolution is accomplished by means of genetic operators applied at the level of fuzzy rule sets. The fitness function evaluates the accuracy of the entire rule base encoded in the chromosome. The genetic learning process used in order to obtain the fuzzy scheduler is illustrated in figure 4.

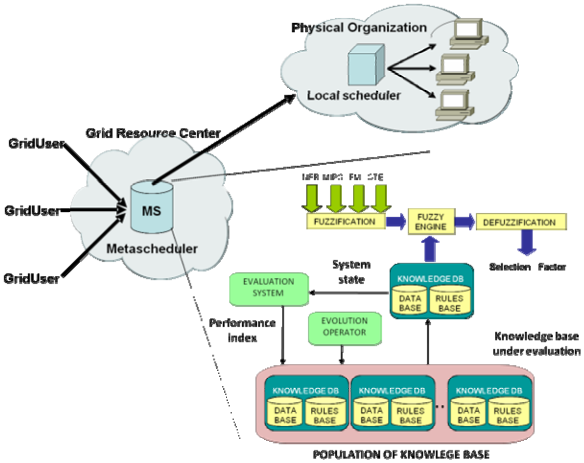


Fig. 4. General structure of Pittsburgh approach

Next, we present the basic mechanism of GAs (coding, initial population creation, fitness function, genetic operators and stopping condition) used in the present work.

Coding of the fuzzy rule base: each chromosome encodes an entire fuzzy rule base. The rules are coded by integer numbers that represent the index of the fuzzy sets that appear in the antecedent and consequent part of the rule. Each rule contains an integer number to indicate the only one connector per rule between the antecedents (“0” for de OR connector and “1” for the AND connector).

Initial population: The initial population is randomly generated and it is necessary to previously define the number of fuzzy rules to be coded in the chromosomes.

Fitness function: The fitness function used is the overall response time for the workflow.

Genetic operators: The genetic operators used in this work are two-point crossover, stochastic universal sampling selection and mutation. Mutation involves replacing an existing rule by another randomly generated one. All these genetic operators work with the elitist strategy.

Stopping condition: In computer simulations, the maximum number of generations has been used as the stopping criterion.

It is important to emphasize that the rules obtained by the Pittsburgh approach do not modify the knowledge base fuzzy sets. Therefore, the interpretability of the obtained rules is not affected.

5 Experimental Results

Due to the difficulties associated to tests in real settings, the scheduler has been verified through the use of simulations. Therefore, it was necessary to develop our scheduler using a grid simulator. We use GridSim [18] which is a simulation toolkit for resource modeling and application scheduling in parallel and distributed computing systems.

We made three comparisons of our proposal. On the one hand, we compared our system with two different schedulers (Random-Random, Round_Robin-Random) and on the other hand, with our own non evolutionary fuzzy scheduler whose rules were chosen according with our previous experience. This fuzzy scheduler has the same structure as the new proposed evolutionary fuzzy scheduler but with different rules. The Random-Random scheduler randomly chooses a PO where to execute a task and then, inside this PO, randomly chooses a resource. The Round_Robin-Random scheduler cyclically chooses the PO and then, inside this PO, randomly chooses a resource where to execute the task.

We executed an optimization problem with 200 independent tasks. Time between arrivals has been modeled using an exponential function. For the grid system we used the following parameters: the number of PO indicates how many PO are involved in each simulation, in this case we have five PO. For each PO we have a number of resources that ranges between 12 and 36. The number of processors for each resource ranges between 1 and 3. The MIPS variable for each processor ranges between 10000 and 16500.

We have applied Pittsburgh approach to obtain our proposed evolutionary fuzzy scheduler. The parameter specifications used in this work are the following: initial population: 20 knowledge bases, crossover probability: 0.8, mutation probability: 0.1, stopping condition: 30 generations.

In table 1, each scheduler execution time is shown. In this table we observe how the fuzzy scheduler improves the used classical scheduler performance. In addition, our proposed evolutionary fuzzy scheduler improves the performance over the rest of schedulers. Concretely, 19.66% better than performance obtained with fuzzy scheduler, 270.80% better than performance obtained with Round_Robin-Random scheduler and 332.87% better than performance obtained with Random-Random scheduler.

The best solution of each generation is showed in Fig 5. Each point represents the mean values of 30 simulations. As expected, execution time decreases with generation number.

Table 1. Execution time for different scheduling approach

Scheduling Approach	Time (sec)
Random-Random	279969.20
Round Robin-Random	239829.44
Fuzzy Scheduler	77391.64
Evolutionary Fuzzy Scheduler	64678.10

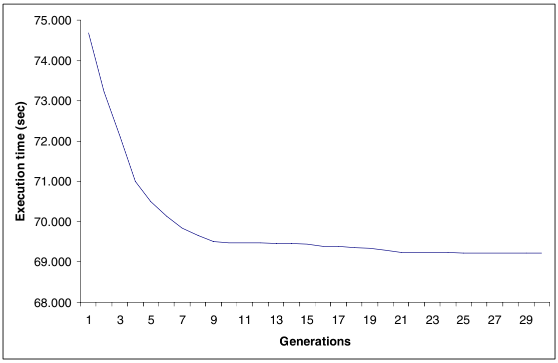


Fig. 5. Execution time of the best solution for each generation (mean of 30 simulations)

6 Conclusions and Future Work

We have proposed an evolutionary fuzzy scheduler obtained by means of a genetic process, concretely, Pittsburgh approach. This new scheduler improves the behavior of traditional scheduling algorithms (Random-Random, Round Robin-Random) and of another non evolved fuzzy scheduler with the same structure. To be precise, we improve 19.66% compared to the best of the other schedulers. In future work we will develop other schemes to evolve the fuzzy scheduler knowledge base. As an example of these schemes we can cite Michigan approach and Iterative Rule Learning approach [7].

Acknowledgments. This work has been financially supported by the Andalusian Government (Research Project P06-SEJ-01694).

References

1. Foster, I., Kesselman, C.: The Grid: Blueprint for a new Computing infrastructure. Morgan Kaufmann Publishers, San Francisco (1999)
2. Germain, C., Breton, V., Clarysse, P., Gaudeau, Y., Glatard, T., Jeannot, E., Legré, Y., Loomis, C., Magnin, I., Montagnat, J., Moureaux, J.-M., Osorio, A., Pennec, X., Texier, R.: Grid-enabling medical image analysis. Journal of Clinical Monitoring and Computing 19(4-5), 339–349 (2005)
3. Stevens, R.D., Robinson, A.J., Goble, C.A.: BmyGrid: Personalized Bioinformatics on the Information Grid. Bioinformatics 19(1), i302–i304 (2003)

4. Spooner, D.P., Cao, J., Jarvis, S.A., He, L., Nudd, G.R.: Performance-aware Workflow Management for Grid Computing. The Computer Journal (2004)
5. Cao, J., Spooner, D.P., Jarvis, S.A., Nudd, G.R.: Grid load balancing using intelligent agents. Future Genetation Comput. Syst. 21(1), 135–149 (2005)
6. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)
7. Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L.: Genetic fuzzy systems. Evolutionary tuning and learning of fuzzy knowledge bases. Advances in fuzzy systems. Applications and theory 19 (2001)
8. Smith, S.F.: A learning system based on genetic adaptive algorithms. Ph.D. thesis, University of Pittsburgh (1980)
9. Norman, T.J., Preece, A., Chalmers, S., Jennings, N.R., Luck, M., Dang, V.D., Nguyen, T.D., Deora, V., Shao, J., Gray, W.A., Fiddian, N.J.: Agent-based formation of virtual organizations. Knowledge-Based Systems 17(2-4), 103–111 (2004)
10. Mandal, A., Kennedy, K., Koelbel, C., Marin, G., Mellor- Crummey, J., Liu, B., Johnsson, L.: Scheduling strategies for mapping application workflows onto the grid. In: Proceedings of HPDC 2005, pp. 125–134 (2005)
11. Sánchez Santiago, A.J., Yuste, A.J., Muñoz Expósito, J.E., García Galán, S., Maqueira Marin, J.M., Bruque Cámara, S.: A dynamic-balanced scheduler for Genetic Algorithms for Grid Computing. Wseas Transactions On Computers, 11–20 (2009)
12. Feitelson, D.G., Weil, A.M.: Utilization and Predictability in Scheduling the IBM SP2 with Backfilling. In: Proceedings of the 12th International Parallel Processing Symposium and the 9th Symposium on Parallel and Distributed Processing, pp. 542–547. IEEE Computer Society Press, Los Alamitos (1998)
13. Litoiu, M., Tadei, R.: Fuzzy Scheduling with Applications on Real Time Systems. Fuzzy Sets and Systems 121, 523–535 (2001)
14. Zhou, J., Yu, J., Chou, C., Yang, L., Luo, Z.: A Dynamic Resource Broker and Fuzzy Logic Based Scheduling Algorithm in Grid Environment. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 604–613. Springer, Heidelberg (2007)
15. Slowinski, R., Hapke, M.: Scheduling Under Fuzziness. Physica-Verlag, Heidelberg (2000)
16. Huang, J., Jin, H., Xie, X., Zhang, Q.: An Approach to Grid Scheduling Optimization Based on Fuzzy Association Rule Mining. In: Proceedings of the First International Conference on e-Science and Grid Computing, e-Science 2005, pp. 189–195 (2005)
17. Jiang, C., Wang, C., Liu, X., Zhao, Y.: A Fuzzy Logic Approach for Secure and Fault Tolerant Grid Job Scheduling. In: Xiao, B., Yang, L.T., Ma, J., Muller-Schloer, C., Hua, Y. (eds.) ATC 2007. LNCS, vol. 4610, pp. 549–558. Springer, Heidelberg (2007)
18. Sulistio, A., Cibej, U., Robic Bb, S., Buyya, R.: A toolkit for Modelling and Simulating Data Grids: An Extension to GridSim. In: Concurrency and Computation: Practice and Experience (CCPE), pp. 1591–1609 (2007)

Improving the Performance of Fuzzy Rule Based Classification Systems for Highly Imbalanced Data-Sets Using an Evolutionary Adaptive Inference System*

Alberto Fernández[1,**], María José del Jesus[2], and Francisco Herrera[1]

[1] Dept. of Computer Science and A.I., University of Granada
{alberto,herrera}@decsai.ugr.es
[2] Dept. of Computer Science, University of Jaén, Spain
mjjesus@ujaen.es

Abstract. In this contribution, we study the influence of an Evolutionary Adaptive Inference System with parametric conjunction operators for Fuzzy Rule Based Classification Systems. Specifically, we work in the context of highly imbalanced data-sets, which is a common scenario in real applications, since the number of examples that represents one of the classes of the data-set (usually the concept of interest) is usually much lower than that of the other classes.

Our experimental study shows empirically that the use of the parametric conjunction operators enables simple Fuzzy Rule Based Classification Systems to enhance their performance for data-sets with a high imbalance ratio.

Keywords: Fuzzy Rule-Based Classification Systems, Inference System, Parametric Conjunction Operators, Genetic Fuzzy Systems, Imbalanced Data-Sets.

1 Introduction

This work is focused on the use of Fuzzy Rule Based Classification Systems (FRBCSs), for classification with imbalanced data-sets [1]. This problem occurs when one or more classes are represented by few examples (known as positive classes), whereas the others are described by many instances (negative classes).

Standard classification algorithms are usually biased towards the majority classes trying to maximize the overall accuracy and their performance is poor on highly imbalanced data-sets. In our previous study on the topic [2], we studied different configurations for FRBCS in order to determine the most suitable model in this framework. Furthermore, we showed the necessity to apply a resampling procedure. Specifically, we found a very good behaviour in the case of the "Synthetic Minority Over-sampling Technique" (SMOTE) [3].

---

* Supported by the Spanish Ministry of Science and Technology under Project TIN2008-06681-C06-01.

** Corresponding author. Tel.:+34-958-240598; Fax: +34-958-243317

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 294–301, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Our aim now is to improve the fuzzy system behaviour by means of the application of parametrized expressions for the conjunction connectors in the Inference System (IS), while maintaining the original interpretability associated to fuzzy systems [4]. This approach is usually called Adaptive Inference System (AIS) and it has shown very good results in fuzzy modelling [5,6]. The idea is to perform a tuning of parameters with a Rule Base (RB) previously established, using Genetic Algorithms (GAs) as a tool to evolve the connector parameters resulting on an Evolutionary Adaptive Inference System (EAIS). Thus, we will increase the performance of the fuzzy models to make them competitive with C4.5 [7], a decision tree algorithm that presents a good behaviour in imbalanced data-sets [8,9].

We will develop an experimental study with twenty-two data-sets from UCI repository with a high imbalance ratio (IR). Multi-class data sets are modified to obtain two-class non-balanced problems, defining the joint of one or more classes as positive and the joint of one or more classes as negative. To evaluate our results we have applied the Area Under the Curve (AUC) metric [10] carrying out some non-parametric tests [11,12] with the aim to show the significance in the performance improvements obtained with the EAIS model.

In order to do that, the contribution is organized as follows. Section 2 presents an introduction on the class imbalance problem, including the description of the problem, proposed solutions, and proper measures for evaluating classification performance in the presence of the imbalance data-set problem. In Section 3 we describe the fuzzy rule learning methodology used in this study, the Chi et al. rule generation method [13]. Next, Section 4 introduces the EAIS with the parametric conjunction operators. In Section 5 we include our experimental analysis in highly imbalanced data-sets. Finally, in Section 6 some concluding remarks are pointed out.

2 Imbalanced Data-Sets in Classification

Learning from imbalanced data is an important topic that has recently appeared in the Machine Learning community [1]. This problem is very representative since it appears in a variety of real-world applications including, but not limited to, medical applications, finance, telecommunications, biology and so on.

In this framework, the class distribution is not uniform, resulting on a high number of examples belonging to one or more classes and a few from the rest. The minority classes are usually associated to the most interesting concepts from the point of view of learning and, due to that fact, the cost derived from a misclassification of one of the examples of these classes is higher than that of the majority classes. In this work we will focus on binary problems where there are just one positive and negative class.

Standard classifier algorithms have a bias towards the majority class, since the rules that predicts the higher number of examples are positively weighted during the learning process in favour of the accuracy metric. Consequently, the instances that belongs to the minority class are misclassified more often than those belonging to the majority class. Nevertheless, the most important issues of this type of problems are the apparition of small disjuncts and the overlapping between the examples of the positive and the negative classes.

In our previous work on this topic [2], we analyzed the cooperation of some preprocessing methods with FRBCSs, showing a good behaviour for the oversampling methods, specially in the case of the SMOTE methodology [3]. According to this, we will employ in this contribution the SMOTE algorithm in order to deal with the problem of imbalanced data-sets.

We will use the IR [8] as a threshold to categorize the different imbalanced scenarios, which is defined as the ratio of the number of instances of the majority class and the minority class. We consider that a data-set presents a high degree of imbalance when its IR is higher than 9 (less than 10% of positive instances).

Regarding the empirical measure, instead of using accuracy, a more correct metric is considered. This is due to the fact that accuracy can lead to erroneous conclusions, since it doesn't take into account the proportion of examples for each class. Because of this, in this work we use the AUC metric [10], which can be defined as

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{1}$$

where TP_{rate} is the percentage of positive cases correctly classified as belonging to the positive class and FP_{rate} is the percentage of negative cases misclassified as belonging to the positive class.

3 Fuzzy Rule Based Classification Systems and Linguistic Rule Generation Method

Any classification problem consists of m training patterns $x_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ from M classes where x_{pi} is the ith attribute value ($i = 1, 2, \ldots, n$) of the p-th training pattern.

In this work we use fuzzy rules of the following form for our FRBCSs:

$$\text{Rule } R_j : \text{ If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then Class} = C_j \text{ with } RW_j \tag{2}$$

where R_j is the label of the jth rule, $x = (x_1, \ldots, x_n)$ is an n-dimensional pattern vector, A_{ji} is an antecedent fuzzy set, C_j is a class label, and RW_j is the rule weight. We use triangular membership functions as antecedent fuzzy sets.

Fuzzy learning methods are the basis to build a FRBCS. The algorithm used in this work is the method proposed in [13], that we have called the Chi et al.'s rule generation. To generate the fuzzy RB this FRBCSs design method determines the relationship between the variables of the problem and establishes an association between the space of the features and the space of the classes by means of the following steps:

1. *Establishment of the linguistic partitions.* Once the domain of variation of each feature A_i is determined, the fuzzy partitions are computed.
2. *Generation of a fuzzy rule for each example* $x_p = (x_{p1}, \ldots, x_{pn}, C_p)$. To do this is necessary:

2.1 To compute the matching degree $\mu(x_p)$ of the example to the different fuzzy regions using a conjunction operator.
2.2 To assign the example x_p to the fuzzy region with the greatest membership degree.
2.3 To generate a rule for the example, whose antecedent is determined by the selected fuzzy region and whose consequent is the label of class of the example.
2.4 To compute the rule weight.

Rules with the same antecedent can be generated during the learning. If they have the same class in the consequent we remove one of the duplicated rules, but if it is different, only the rule with the highest weight is kept in the RB.

4 Evolutionary Adaptive Inference System

The conjunction operator is employed in the IS to compute the activation strength of the if-part for all rules in the RB with an input pattern. This operator is suitable to be parameterized in order to adapt the IS showing a considerable improvement in the accuracy of linguistic Fuzzy Systems [5,6].

We will use Dubois t-norm, because it is more efficiently computed and since it has obtained a better behaviour than other parametric t-norms [5]. The expression for the computation of this parametric t-norm is shown below:

$$T_{Dubois}(x, y, \alpha) = \frac{x \cdot y}{\max(x, y, \alpha)} \quad (0 \leq \alpha \leq 1) \tag{3}$$

We must note that Dubois t-norm achieves like a minimum when $\alpha = 0$ and like algebraic product $\alpha = 1$. When $0 < \alpha < 1$, it continues performing like minimum excepting when every match with antecedents are below α, that takes values between minimum and product, being similar to a concentration effect.

In this work we will use a model that considers individual parameters α_i for every rule of the KB, having a local tuning mechanism of the behavior of the IS in order to obtain a good accuracy level, because of the high degree of freedom of this model. We will consider the use of a specific GA to design the proposed learning method, the CHC algorithm. The CHC algorithm is a GA that presents a good trade-off between diversity and convergence, being a good choice in problems with complex search spaces.

This genetic model makes use of a mechanism of "Selection of Populations". M parents and their corresponding offspring are put together to select the best M individuals to take part of the next population (with M being the population size). Furthermore, no mutation is applied during the recombination phase, Instead, when the population converges or the search stops making progress, the population is re-initialized.

The components needed to design this process are explained below. They are: coding scheme, initial gene pool, chromosome evaluation, crossover operator (together with an incest prevention) and restarting approach.

1. *Coding Scheme:* Each chromosome will be composed by R genes, being R the number of rules in the RB, having a value between 0 and 1.
2. *Chromosome Evaluation:* The fitness function must be in accordance with the framework of imbalanced data-sets. Thus, we will use, as presented in Section 2, the AUC measure.
3. *Initial Gene Pool:* We will initialize one chromosome with all its genes at 0 to model the minimum t-norm and another chromosome with all genes at 1 to model the product t-norm. The remaining individuals of the population will be generated at random in the interval [0,1].
4. *Crossover Operator:* We consider the Parent Centric BLX (PCBLX) operator with incest prevention mechanism. In this manner, two parents are crossed if their hamming distance divided by 2 is above a predetermined threshold, L. Since we consider a real coding scheme, we have to transform each gene considering a Gray Code (binary code) with a fixed number of bits per gene ($BITSGENE$). In this way, the threshold value is initialized as:

   $$L = (\#Genes \cdot BITSGENE)/4.0$$

   L is decremented by $BITSGENE$ when there are no new individuals in the next generation.
5. *Restarting approach:* When the threshold value L is lower than zero, all the chromosomes are regenerated at random within the interval [0.0, 1.0]. Furthermore, the best global solution found is included in the population to increase the convergence of the algorithm.

5 Experimental Study

The objective of this work is to analyse the behaviour achieved by FRBCSs using the EAIS model. With this aim, we will perform an empirical study using a large collection of highly imbalanced data-sets to support our analysis. We will also include the C4.5 decision tree [7], since this method has a good behaviour in the framework of imbalanced data-sets [8,9].

We have selected twenty-two data-sets with different IR from UCI repository. The data is summarized in Table 1, showing the number of examples (#Ex.), number of attributes (#Atts.), class name of each class (minority and majority), class attribute distribution and IR.

In the remaining of this section, we will first present the experimental framework and all the parameters employed in this study. Then, we will illustrate how the EAIS model enhances the behaviour of the base FRBCS model also contrasting its performance with C4.5.

5.1 Experimental Set-Up

To develop the different experiments we consider a *5-folder cross-validation model*, i.e., 5 random partitions of data with a 20%, and the combination of

Table 1. Summary Description for Imbalanced Data-Sets

Data-set	**#Ex.**	**#Atts.**	**Class (min., maj.)**	**%Class(min.; maj.)**	**IR**
Yeast2vs4	514	8	(cyt; me2)	(9.92, 90.08)	9.08
Yeast05679vs4	528	8	(me2; mit,me3,exc,vac,erl)	(9.66, 90.34)	9.35
Vowel0	988	13	(hid; remainder)	(9.01, 90.99)	10.10
Glass016vs2	192	9	(ve-win-float-proc; build-win-float-proc, build-win-non_float-proc,headlamps)	(8.89, 91.11)	10.29
Glass2	214	9	(Ve-win-float-proc; remainder)	(8.78, 91.22)	10.39
Ecoli4	336	7	(om; remainder)	(6.74, 93.26)	13.84
Yeast1vs7	459	8	(nuc; vac)	(6.72, 93.28)	13.87
Shuttle0vs4	1829	9	(Rad Flow; Bypass)	(6.72, 93.28)	13.87
Glass4	214	9	(containers; remainder)	(6.07, 93.93)	15.47
Page-blocks13vs2	472	10	(graphic; horiz.line,picture)	(5.93, 94.07)	15.85
Abalone9vs18	731	8	(18; 9)	(5.65, 94.25)	16.68
Glass016vs5	184	9	(tableware; build-win-float-proc, build-win-non_float-proc,headlamps)	(4.89, 95.11)	19.44
Shuttle2vs4	129	9	(Fpv Open; Bypass)	(4.65, 95.35)	20.5
Yeast1458vs7	693	8	(vac; nuc,me2,me3,pox)	(4.33, 95.67)	22.10
Glass5	214	9	(tableware; remainder)	(4.20, 95.80)	22.81
Yeast2vs8	482	8	(pox; cyt)	(4.15, 95.85)	23.10
Yeast4	1484	8	(me2; remainder)	(3.43, 96.57)	28.41
Yeast1289vs7	947	8	(vac; nuc,cyt,pox,erl)	(3.17, 96.83)	30.56
Yeast5	1484	8	(me1; remainder)	(2.96, 97.04)	32.78
Ecoli0137vs26	281	7	(pp,imL; cp,im,imU,imS)	(2.49, 97.51)	39.15
Yeast6	1484	8	(exc; remainder)	(2.49, 97.51)	39.15
Abalone19	4174	8	(19; remainder)	(0.77, 99.23)	128.87

4 of them (80%) as training and the remaining one as test. For each data-set we consider the average results of the five partitions.

We must emphasize that, in order to reduce the effect of imbalance, we have employed the SMOTE preprocessing method [3] for all our experiments, considering only the 1-nearest neighbour to generate the synthetic samples, and balancing both classes to the 50% distribution.

We will carry out a rigorous comparison of the performance obtained by the algorithms used in this work. We consider the use of non-parametric tests due to the fact that the initial conditions that guarantee the reliability of the parametric tests are not satisfied in a single data-set analysis [11,12]. We will use Wilcoxon's Signed-Ranks Test [14], which computes the ranking of both algorithms (noted as R^+ and R^-) with respect to the absolute differences between their performance in each data-set (average ranks are assigned in case of ties). The level of confidence (α) will be set at 0.05 (95% of confidence).

We will employ the following configuration for the FRBCS: 3 fuzzy labels, product T-norm as conjunction operator (for the base FRBCS), together with the Penalized Certainty Factor approach [15] for the rule weight and FRM of the winning rule. Finally, the values that have been considered for the parameters of the CHC algorithm are the following ones: 50 individuals for the population size, 5,000 $\cdot$ *number of variables* as number of evaluations and 30 bits per gene for the Gray codification (incest prevention mechanism).

5.2 Analysis of the Behaviour of the EAIS

The results for the Chi et al.'s algorithm, the Chi method with EAIS and C4.5 are shown in Table 2. We observe that the EAIS improves the performance for the FRBCS in many data-sets and achieves the best global result among the algorithms used in the analysis.

Table 2. Detailed results table for the Chi FRBCS, basic approach and with EAIS (parametric conjunction operator), and C4.5 for highly imbalanced data-sets

Data-set	Chi		Chi+EIAS		C4.5	
	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
Yeast2vs4	89.68 ± 1.30	87.36 ± 5.16	92.99 ± 1.91	86.31 ± 5.29	98.14 ± 0.88	85.88 ± 8.78
Yeast05679vs4	82.65 ± 1.38	79.17 ± 5.66	86.38 ± 1.39	80.75 ± 5.92	95.26 ± 0.94	76.02 ± 9.36
Vowel0	98.57 ± 0.18	98.39 ± 0.60	99.86 ± 0.18	98.28 ± 1.57	99.67 ± 0.48	94.94 ± 4.95
Glass016vs2	62.71 ± 2.15	54.17 ± 6.82	73.52 ± 6.50	63.69 ± 15.38	97.16 ± 1.86	60.62 ± 12.66
Glass2	66.54 ± 2.18	55.30 ± 14.48	77.39 ± 2.97	59.71 ± 17.82	95.71 ± 1.51	54.24 ± 14.01
Ecoli4	94.06 ± 1.49	91.51 ± 7.21	98.62 ± 0.42	92.15 ± 7.45	97.69 ± 1.96	83.10 ± 9.90
Yeast1vs7	82.00 ± 2.34	80.63 ± 6.61	86.82 ± 1.71	73.93 ± 12.07	93.51 ± 2.20	70.03 ± 1.46
Shuttle0vs4	100.00 ± 0.00	99.12 ± 1.14	100.00 ± 0.00	99.12 ± 1.14	99.99 ± 0.02	99.97 ± 0.07
Glass4	95.27 ± 0.91	85.70 ± 12.92	97.82 ± 0.73	86.94 ± 12.10	98.44 ± 2.29	85.08 ± 9.35
Page-Blocks13vs2	93.68 ± 2.41	92.05 ± 4.73	97.38 ± 0.58	93.94 ± 4.42	99.75 ± 0.21	99.55 ± 0.47
Abalone9-18	70.23 ± 2.25	64.70 ± 10.73	78.08 ± 4.68	62.52 ± 12.67	95.31 ± 4.44	62.15 ± 4.96
Glass016vs5	90.57 ± 4.12	79.71 ± 23.29	98.57 ± 0.56	86.29 ± 22.71	99.21 ± 0.47	81.29 ± 24.44
Shuttle2vs4	95.00 ± 4.71	90.78 ± 7.80	98.98 ± 2.28	96.38 ± 6.98	99.90 ± 0.23	99.17 ± 1.86
Yeast1458vs7	71.25 ± 3.52	64.65 ± 5.92	80.50 ± 1.05	58.79 ± 4.48	91.58 ± 2.78	53.67 ± 2.09
Glass5	94.33 ± 1.23	83.17 ± 11.12	98.23 ± 0.59	86.10 ± 12.45	99.76 ± 0.40	88.29 ± 13.31
Yeast2vs8	78.61 ± 2.61	77.28 ± 10.36	81.71 ± 1.41	77.18 ± 10.23	91.25 ± 1.84	80.66 ± 11.22
Yeast4	83.58 ± 0.93	83.15 ± 2.96	89.63 ± 1.31	83.92 ± 4.12	91.01 ± 2.64	70.04 ± 5.65
Yeast1289vs7	74.70 ± 1.79	77.12 ± 6.50	84.65 ± 2.45	74.51 ± 3.91	94.65 ± 1.13	68.32 ± 6.16
Yeast5	94.68 ± 1.28	93.58 ± 5.11	98.16 ± 0.14	93.30 ± 4.61	97.77 ± 1.45	92.33 ± 4.72
Ecoli0137vs26	93.96 ± 5.63	81.90 ± 20.49	98.27 ± 1.02	82.08 ± 20.58	96.78 ± 3.28	81.36 ± 21.68
Yeast6	88.48 ± 2.38	88.09 ± 9.82	92.73 ± 1.79	87.94 ± 8.03	92.42 ± 3.54	82.80 ± 12.77
Abalone19	71.44 ± 1.82	63.94 ± 9.32	76.76 ± 3.26	58.23 ± 8.12	85.44 ± 2.49	52.02 ± 4.41
Mean	85.09 ± 2.12	80.52 ± 8.58	90.32 ± 1.68	**81.00 ± 9.18**	95.93 ± 1.68	78.25 ± 8.38

In order to support this conclusion, we carry out a Wilcoxon test (shown in Table 3) comparing the Chi et al.'s method versus the EAIS. We observe that the EAIS achieves a higher ranking, which implies the robustness of this approach. When comparing with C4.5 (also in Table 3) we must emphasize that the use of the EAIS outperforms C4.5, whereas the simple FRBCS is not enough to obtain significant differences.

Table 3. Wilcoxon's test to compare the basic Chi method (R^+) against the Chi approach with EAIS (parametric conjunction operator) (R^-)

Comparison	R^+	R^-	Hypothesis ($\alpha = 0.05$)	p-value
Chi+AIS vs. Chi	143.5	109.5	Not Rejected	0.566
Chi vs. C4.5	176.0	77.0	Not Rejected	0.108
Chi+AIS vs. C4.5	205.0	48.0	Rejected for Chi+AIS	0.011

In brief, the EAIS is a good design option in order to improve the accuracy in linguistic FRBCSs for imbalanced data-sets with a high IR. Furthermore, we must point out that the effect of the adaptive t-norm playing the role of conjunction operator does not modify the shape of the inferred fuzzy set, maintaining the original interpretability of the fuzzy labels.

6 Conclusions

The aim of this contribution was to study the goodness of the EAIS model to improve the performance of simple FRBCS in the framework of imbalanced data-sets with a high IR. The idea is to maintain the initial interpretability of the fuzzy model by changing the computation of the conjunction operator.

Our experimental analysis have shown that the EAIS approach obtains very good results in highly imbalanced data-sets in contrast with the base FRBCS. Furthermore, we observe that the Chi et al. method by itself do not outperforms the results of the C4.5 decision tree, but this algorithm is highly improved with the application of the EAIS model, obtaining significant differences with respect to C4.5 in this case.

References

1. Chawla, N.V., Japkowicz, N., Kolcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations 6(1), 1–6 (2004)
2. Fernández, A., García, S., del Jesus, M.J., Herrera, F.: A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. Fuzzy Sets and Systems 159(18), 2378–2398 (2008)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligent Research 16, 321–357 (2002)
4. Crockett, K.A., Bandar, Z., Mclean, D., O'Shea, J.: On constructing a fuzzy inference framework using crisp decision trees. Fuzzy Sets and Systems 157, 2809–2832 (2006)
5. Alcalá-Fdez, J., Herrera, F., Márquez, F.A., Peregrín, A.: Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems. International Journal of Intelligent Systems 22(9), 1035–1064 (2007)
6. Márquez, F.A., Peregrín, A., Herrera, F.: Cooperative evolutionary learning of fuzzy rules and parametric aggregation connectors for mamdani linguistic fuzzy systems. IEEE Transactions on Fuzzy Systems 15(6), 1162–1178 (2007)
7. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
8. Orriols-Puig, A., Bernadó-Mansilla, E.: Evolutionary rule-based systems for imbalanced datasets. Soft Computing 13(3), 213–225 (2009)
9. Su, C.T., Hsiao, Y.H.: An evaluation of the robustness of MTS for imbalanced data. IEEE Transactions on Knowledge Data Engineering 19(10), 1321–1332 (2007)
10. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering 17(3), 299–310 (2005)
11. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
12. García, S., Fernández, A., Luengo, J., Herrera, F.: A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. Soft Computing 13(10), 959–977 (2009)
13. Chi, Z., Yan, H., Pham, T.: Fuzzy algorithms with applications to image processing and pattern recognition. World Scientific, Singapore (1996)
14. Sheskin, D.: Handbook of parametric and nonparametric statistical procedures. Chapman & Hall/CRC, Boca Raton (2006)
15. Ishibuchi, H., Yamamoto, T.: Rule weight specification in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 13, 428–435 (2005)

A t-Norm Based Approach to Edge Detection

C. Lopez-Molina[1], H. Bustince[1], J. Fernández[1], E. Barrenechea[1], P. Couto[2],
and B. De Baets[3]

[1] Departamento de Automática y Computación, Universidad Pública de Navarra,
Campus Arrosadia s/n, P.O. Box 31006, Pamplona, Spain
bustince@unavarra.es

[2] Department of Engineering, CITAB, University of Trás-os-Montes e Alto Douro
P.O. Box 1013, 5001-801 Vila Real, Portugal
pcouto@utad.pt

[3] Dept. of Applied Mathematics, Biometrics and Process Control,
Universiteit Gent Coupure links 653, 9000 Gent, Belgium
bernard.debaets@ugent.be

Abstract. In this paper we study a modification of the original method by Genyun Sun *et al.* [1] for egde detection based on the Law of Universal Gravitation. We analyze the effect of the substitution of the product by other t-norms in the calculation of the gravitatory forces. We construct a fuzzy set where memberships to the edges are extracted from the magnitude of the resulting force on each pixel. To finish, we experimentally proof that the features of each t-norm determine the kind of edges to be detected.

Keywords: Image processing, t-norm, Edge detection, The law of universal gravity.

1 Introduction

In the early years of image processing, many methods were introduced to detect edges in an image. Most of them were based on a convolution of the image with a given mask (e.g. Sobel [2] or Prewitt [3]), and a few of them included some kind of both post- and pre-processing (as Canny [4]).

In the last years, completely new approaches were explored. These include edge detectors based on filtering techniques [5,6], neural networks [7,8], statistical measurements [9], Interval-valued fuzzy sets treatments [10] or fuzzy pattern matching [11,12]. Some of those methods ([13,14,15]) were adapted not only to single greyscale images but also to color or multispectral ones. Even if their performance might be better than the classic ones, they are not very commonly applied. Most common implementations are still the mask-based ones.

Following this trend, we propose a simple, customizable method for edge detection. Our method is based on the innovatory original usage of the gravitational forces by Genyun Sun *et al.* (see [1]). We generalize this algorithm by considering a t-norm other than the product. The proposed algorithm is very simple, but keeps a high degree of flexibility with little configuration issues.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 302–309, 2009.
© Springer-Verlag Berlin Heidelberg 2009

We have structured the reminder of this paper as follows. Section 2 includes some basic concepts. Then, section 3 describes the gravitational method introduced by Genyun Sun *et al.* [1]. Our proposal is depicted in section 4. In the section 5 we present the algorithm. We introduce some results on both synthetic environments and natural images (section 6). To finish, some short conclusions are included in section 7.

2 Preliminaries

In this paper we make an extensive use of the triangular norms and their properties. A triangular norm (t-norm for short) is a binary aggregation function $T : [0,1]^2 \to [0,1]$, which is commutative, associative, symmetric and has neutral element 1. See [16,17] for further details.

Some examples of tnorms, all of them used in the course of this paper, are

$$\begin{array}{ll} \text{product t-norm} & T_{\mathbf{P}}(a,b) = a \cdot b \\ \text{minimum t-norm} & T_{\mathbf{M}}(a,b) = \min(a,b) \\ \text{Łukasiewicz t-norm} & T_{\mathbf{L}}(a,b) = \max(a+b-1,0) \\ \text{nilpotent minimum} & T_{\mathbf{nM}}(a,b) = \begin{cases} \min(a,b) & \textit{if } a+b>1 \\ 0 & \text{otherwise} \end{cases} \end{array}$$

The first three t-norms are the most classical examples [17,18], whereas the last one has been selected as an example of discontinuous t-norm.

3 Gravity Approach to Edge Detection

In this paper we will denote with (x,y) the coordinates of each pixel on the image Q, of height M and width N; with $q(x,y)$ we will represent the intensity or grey level of the pixel (x,y) so that $0 \leq q(x,y) \leq L-1$ for each $(x,y) \in Q$, where $L-1$ is the highest grey level of our grayscale. Under this representation an image is considered a relation on the universal set $M \times N$.

Newton's Law of Universal Gravitation (LUG)[19] states that any mass attracs every other mass with an intensity directly proportional to the product of both masses and inversely proportional to the square of the distance that separates them. Mathematically, the gravitatory force $\overrightarrow{f}_{1,2}$ exerted by a mass m_1 over a mass m_2 can be written as

$$\overrightarrow{f}_{1,2} = G \cdot \frac{m_1 \cdot m_2}{|\overrightarrow{r}|^2} \cdot \frac{\overrightarrow{r}}{|\overrightarrow{r}|}$$

where $\overrightarrow{r}$ is the vector connecting m_1 with m_2 and G the gravitational constant. In [1] this law was adapted to image processing by identifying each pixel with a body of mass equal to its intensity. In that approach the image is considered to be a grid of side 1. Given the paralelism we will consider the terms *intensity of a pixel* $(q(x,y))$ and *mass of a pixel* $(m(x,y))$ as equivalent.

In short, the methodology proposes is to associate each pixel with the resulting force of the gravitational forces exerted on it by the surrounding. In order to

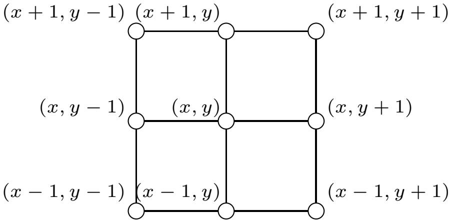


Fig. 1. Pixel environment convention

simplify the problem, the authors reduce the environment to a 3 × 3 or 5 × 5 window centered on each pixel.

In [1] forces exected on a pixel do not depend exclusively on the difference of intensities in the surrounding pixels but also on the intensity of the pixel itself. Therefore, the zones with higher intensities are more likely to include edges. This might lead to bad performance by missing some edges in low intensity zones, or by empowering little variations in the high intensity regions.

4 Construction of a Fuzzy Set from Gravitational Forces

Our idea is to create a fuzzy set representing the edges, so that every position in the image is assigned a membership proportional to the estimated resulting force. To do it, we have to be sure that all the resulting forces are going to have magnitudes in the range $[0, 1]$, so that they can be used as valid memberships.

Considering zero-mass bodies as valid ones we could by means of normalization, with no loss of generality, consider all the masses to be in the interval $[0, 1]$. Note that the resulting forces will not be the same, but only proportional. Even if this would be problematic in a real world environment, it can be applied to digital images, where the maximum intensity of a pixel (representing its mass) is well known. As we want to avoid the completely black pixels (representing 0 masses), the pixel intensities will be normalized to the $(0, 1]$ interval. This can be done as shown in section 5.

Thus, we can tune up the LUG if considering normalized masses

$$\overrightarrow{f}(x,y) = G \cdot \frac{m_1 \cdot m_2}{|\overrightarrow{r}|^2} \frac{\overrightarrow{r}}{|\overrightarrow{r}|} = G \cdot \frac{T_{\mathbf{P}}(m_1, m_2)}{|\overrightarrow{r}|^2} \frac{\overrightarrow{r}}{|\overrightarrow{r}|}$$

The previous expression gave us the idea to adapt the formula to any t-norm T.

$$\overrightarrow{f}(x,y) = G \cdot \frac{T(m_1, m_2)}{|\overrightarrow{r}|^2} \frac{\overrightarrow{r}}{|\overrightarrow{r}|} \tag{1}$$

Using the formula (1), it is possible to apply the adapted LUG edge detection. Reducing the environment to a 3 × 3 pixel entourage, the naming of each of the pixels, given a concrete pixel (x, y) is as shown in image 1. The total force exerted on a given pixel (x, y) is

$$\overrightarrow{F_T}(x,y) = \sum_{\substack{-1 \leq i \leq 1 \\ -1 \leq j \leq 1 \\ (i,j) \neq (0,0)}} \Theta \cdot a_r(i,j) \cdot \frac{T(q(x,y), q(x+i,y+j))}{|\overrightarrow{r}|^2} \cdot \frac{\overrightarrow{r}}{|\overrightarrow{r}|} \tag{2}$$

where Θ is a normalizing constant that substitutes the gravitational constant G, $\overrightarrow{r}$ the vector joining each pair of pixels and $a_r(i,j)$ is a positive factor in $(0,1]$ used to reduce the impact of the diagonal neighbours [1]. For simplicity, we choose for the pixels out of the diagonals, $a_r(i,j) = 1$, while the other pixels are assigned a fixed $a_r(i,j) \leq 1$. The effect of this parameter is similar to the effect obtained by Sobel [2] in contrast to Prewitt [3]. In the experiments, $a_r(i,j) = 1$ for pixels in any position (i,j).

Let be the referencial set $(M-2) \times (N-2)$, that is, the set of positions of the image, onces the first and last rows and columns have been removed. In these conditions a fuzzy set is constructed so that

$$\begin{aligned} \mu_A : (M-2) \times (N-2) \rightarrow & \quad [0,1] \quad \text{given by} \\ \mu_A(x,y) = & \left| \overrightarrow{F_T}(x,y) \right| \end{aligned}$$

That is, $A = \{((x,y), \mu_A(x,y)) \, | (x,y) \in (M-2) \times (N-2)\}$

Remark. Last fuzzy set definition excludes the pixels in the limits of the image. This is a typical problem in image processing, and has little practical impact in any application, Therefore, they will be ignored, associating them any fixed membership.

Regarding Θ, remember that the magnitude of the resulting forces has to be in $[0,1]$. Θ is a normalizing constant ensuring this fact. In particular, it can take many different values. Another solution would have been to divide each of the magnitudes of a gradient in an image by the highest one. However, this would make each of the fuzzy sets non-comparable among them. That is, a high membership of a pixel to a set could be related to a edge situation or to the fact that the normalization process made it increase.

5 Algorithmic Representation

Once shown how the methodology is, we could present a simple algorithm for the basic membership and direction extraction, such as seen in the figure 2.

6 Experimental Results

Initially synthetic edges in 32 × 32 pixels images have been tested. The results on vertical ones can be seen in the first rows of the figure 3. In each row it is shown the original image (first on the left side), then the edge fuzzy sets created.

There are some remarks to do after this experiment. First, it can be seen how the $T_{\mathbf{M}}$ t-norm performs slightly better than the one based on $T_{\mathbf{P}}$, both having the best results. However, the most noteworthy fact is the unability of the $T_{\mathbf{L}}$ and $T_{\mathbf{nM}}$ based detectors to discern some of the edges.

```
Algorithm Edge detection
Inputs
    image ← image in the range [0..1]
    $T$ ← $t-$norm to be used
    $\{b', a_r\}$ ← normalization and diagonal pixel factors
    $\Theta$ ← normalizing constant
Outputs
    $A$ ← edges fuzzy set
Begin
    image ← $\frac{image + b'}{1 + b'}$
    for each non-boundary position $(x, y) \in Q$ do
        $\mu_A(x,y) = \left|\sum_{\substack{-1 \leq i \leq 1 \\ -1 \leq j \leq 1 \\ (i,j) \neq (0,0)}} \Theta \cdot a_r(i,j) \cdot \frac{T(q(x,y), q(x+i, y+j))}{|\overrightarrow{r}|^2} \cdot \frac{\overrightarrow{r}}{|\overrightarrow{r}|}\right|$
    end
FAlgoritmo
```

Fig. 2. Basic edge detection algorithm using figure 1 nomenclature

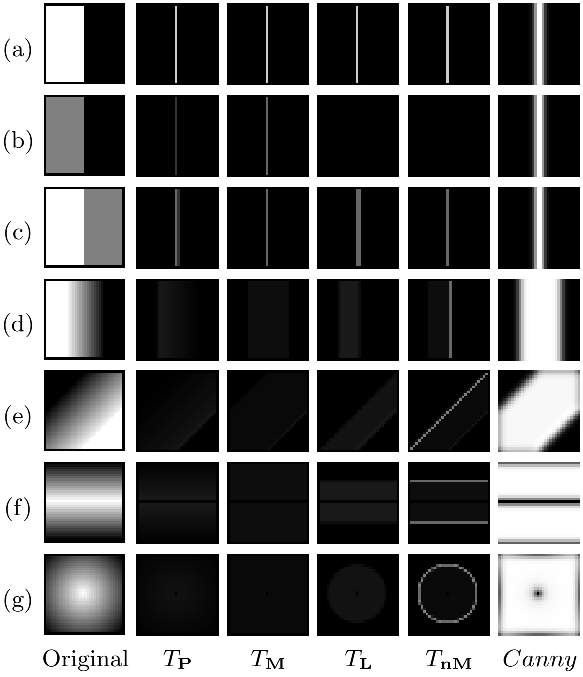
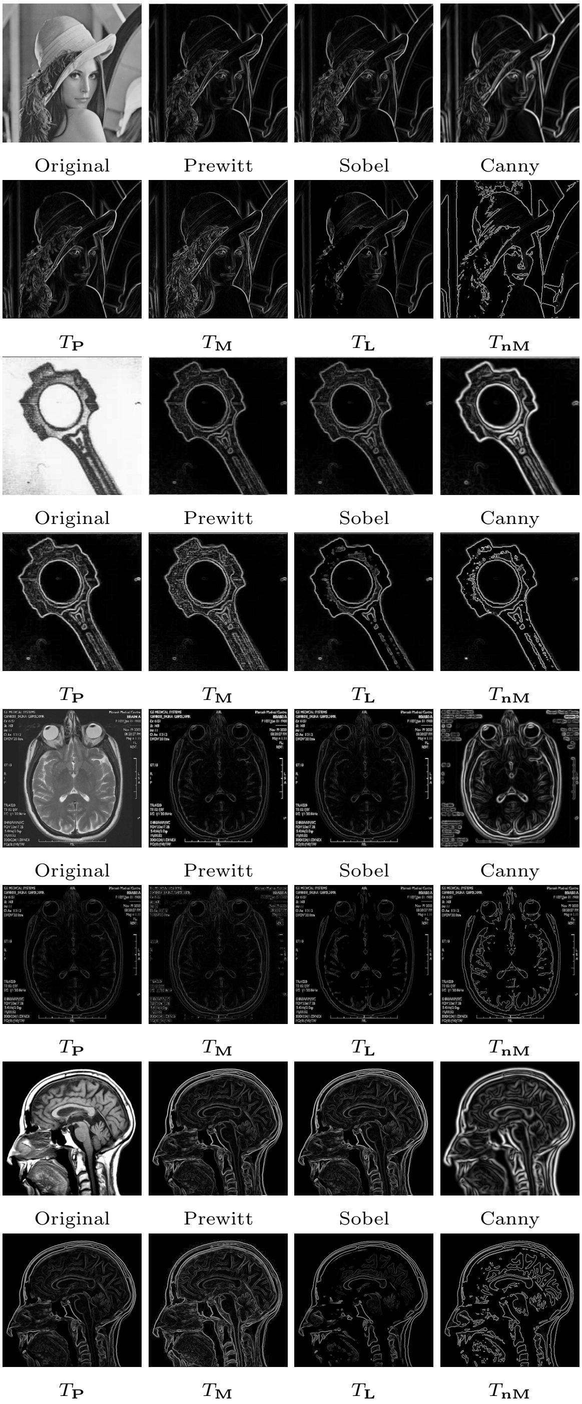


Fig. 3. Results with vertical edges synthetic images


Original Prewitt Sobel Canny

$T_{\mathbf{P}}$ $T_{\mathbf{M}}$ $T_{\mathbf{L}}$ $T_{\mathbf{nM}}$

Original Prewitt Sobel Canny

$T_{\mathbf{P}}$ $T_{\mathbf{M}}$ $T_{\mathbf{L}}$ $T_{\mathbf{nM}}$

e

Original Prewitt Sobel Canny

$T_{\mathbf{P}}$ $T_{\mathbf{M}}$ $T_{\mathbf{L}}$ $T_{\mathbf{nM}}$

Original Prewitt Sobel Canny

$T_{\mathbf{P}}$ $T_{\mathbf{M}}$ $T_{\mathbf{L}}$ $T_{\mathbf{nM}}$

Fig. 4. Results for a Lena, Span, two MRI images

The most relevant facts were discovered when using gradient edges. Those edges are usually left aside, as they are not very common in natural images. Thus, many detectors are just not prepared for their extraction. In the figure 3 (d-e) it can be perceived how discontinuous t-norm $T_{\mathbf{nM}}$ is a great basis for the detection of these edges. The key to be understood in this context is that, in the discontinuity point of these t-norms, forces produced on a single pixel can have great variations with little intensity changes. Thus, significant forces can come up in points simply ignorable when analyzed with different continuous t-norms. This result, along with most of the others, was predictable after analyzing the features of the operators (see [17,18,16]).

Many proofs have been done with large sets of over 100 natural pictures, all of them taken from Carnegie Mellon University's web page (http://www.cs.cmu.edu/-cil/v-images.html). However, only a little portion of them could be introduced into the final version of the paper. The tests shown in here (figure 4) make use of some common 256 × 256 pixel images, plus some other medical images of special interest in our research. The results are compared with the greyscale image generated in some of the most common edge detection methods. Both Sobel [2] and Prewitt [3] methods are applied with a classical 3×3 mask, while Canny [4] is obtained from the Matlab 7.6.0 implementation, with a default 7×7 mask. No postprocessing (e.g. thresholding or non-maximum supression) have been applied on any of the results.

Remark. All the results are normalized, to turn the comparison easier. Therefore, Canny results for roof edges show extremely wide edges.

The results we obtained are the same as in the synthetical experiments. Both $T_{\mathbf{M}}$ and $T_{\mathbf{P}}$ based approaches show regular results, with the $T_{\mathbf{M}}$ detector assigning higher memberships, specially in darker zones. Łukasiewicz t-norm based detector avoids intensity changes in the darker zones, but becomes sensitive in the brighter ones. This allows the texture of the span to be avoided in figure 4. To finish, $T_{\mathbf{nM}}$ detector makes use of the discontinuity to show sharp, thin edges. These results are specially desirable in the case of the medical images.

However, the key question here is not how good do the different detectors perform, but which differences can be observed on behalf of the used tnorm. Moreover, we made a link between well-known functions (as the t-norms are) and a hard problem like edge detection. While keeping this connection we can work in the analytical field to improve the results to be obtained later on.

7 Conclusions and Future Lines

This paper has introduced a development in the method proposed in [1], leading to a customizable edge detector. The results in the natural images has been proven to be competitive with the common edge detection methods, with different results obtained from different t-norms. The results obtained in the natural images (section 6) expose an special accuracy of some methods in some concrete situations, like the $T_{\mathbf{nM}}$ based operator in medical images. Furthermore, the results can be predicted from the features of the t-norm, what is a great help in case we know the kind of images we are dealing with.

Acknowledgements. This research was partially supported by the grant TIN 2007-65981. The authors would like to acknowledge the referees for their useful comments and suggestions. Some of their comments have been reproduced in the text.

References

1. Sun, G., Liu, Q., Ji, C., Li, X.: A novel approach for edge detection based on the theory of universal gravity. Pattern Recognition 40(10), 2766–2775 (2007)
2. Sobel, I., Feldman, G.: A 3x3 isotropic gradient operator for image processing. Presented at a talk at the Stanford Artificial Project
3. Prewitt, J.: Object enhancement and extraction. Picture processing and Psychopictorics, 75–149 (1970)
4. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)
5. Ducottet, C., Fournel, T., Barat, C.: Scale-adaptive detection and local characterization of edges based on wavelet transform. Signal Processing 84(11), 2115–2137 (2004); Special Section Signal Processing in Communications
6. Zhang, L., Bao, P.: Edge detection by scale multiplication in wavelet domain. Pattern Recognition Letters 23(14), 1771–1784 (2002)
7. Lu, S., Wang, Z., Shen, J.: Neuro-fuzzy synergism to the intelligent system for edge detection and enhancement. Pattern Recognition 36(10), 2395–2409 (2003)
8. Toivanen, P.J., Ansamäki, J., Parkkinen, J.P.S., Mielikäinen, J.: Edge detection in multispectral images using the self-organizing map. Pattern Recognition Letters 24(16), 2987–2994 (2003)
9. Konishi, S., Yuille, A., Coughlan, J.: A statistical approach to multi-scale edge detection. Image and Vision Computing 21(1), 37–48 (2003)
10. Bustince, H., Barrenechea, E., Pagola, M., Fernandez, J.: Interval-valued fuzzy sets constructed from matrices: Application to edge detection. Fuzzy Sets and Systems (2008), doi:10.1016/j.fss.2008.08.005
11. Kim, D.S., Lee, W.H., Kweon, I.S.: Automatic edge detection using 3 x 3 ideal binary pixel patterns and fuzzy-based edge thresholding. Pattern Recognition Letters 25(1), 101–106 (2004)
12. Liang, L.R., Looney, C.G.: Competitive fuzzy edge detection. Applied Soft Computing 3(2), 123–137 (2003)
13. Bezdek, J., Chandrasekhar, R., Attikouzel, Y.: A geometric approach to edge detection. IEEE Transactions on Fuzzy Systems 6(1), 52–75 (1998)
14. Kang, C.C., Wang, W.J.: A novel edge detection method based on the maximizing objective function. Pattern Recognition 40(2), 609–618 (2007)
15. Tang, H., Wu, E.X., Ma, Q.Y., Gallagher, D., Perera, G.M., Zhuang, T.: MRI brain image segmentation by multi-resolution edge detection and region selection. Computerized Medical Imaging and Graphics 24(6), 349–357 (2000)
16. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms. Kluwer Academic Publishers, Dordrecht (2000)
17. Beliakov, G., Pradera, A., Calvo, T.: Aggregation Functions: A Guide for Practitioners. Studies in Fuzziness and Soft Computing, vol. 221. Springer, Heidelberg (2007)
18. Alsina, C., Frank, M.J., Schweizer, B.: Associative functions: Triangular Norms and Copulas. World Scientific, Singapore (2006)
19. Newton, I.: Philosophiae Naturalis Principia Mathematica. University of California (1999); original 1687, translation guided by I. Bernard Cohen

Applying Evolutionary Computation Methods to Formal Testing and Model Checking[⋆]

Pablo Rabanal, Ismael Rodríguez, and Fernando Rubio

Dept. Sistemas Informáticos y Computación
Universidad Complutense de Madrid, 28040 Madrid, Spain
prabanal@fdi.ucm.es, {isrodrig,fernando}@sip.ucm.es

Abstract. Formal Methods (FM) provide mathematically founded algorithms for analyzing the correctness of systems. Though FM have been successfully applied to many industrial problems, these methods typically find the practical problem that the number of states to be systematically analyzed grows exponentially with the size of the system to be analyzed. Thus, exhaustive techniques to find system faults are typically substituted by heuristic strategies allowing to focalize the search for potential faults in some suspicious or critical configurations. In particular, Evolutionary Computation (EC) methods provide efficient generic strategies to search for good solutions in big solution spaces, which fit into the kind of problems appearing in FM. This paper summarizes several works where EC techniques have been applied to FM problems.

1 Introduction

Nature-inspired methods (see e.g. [19,18,10]) are useful because they face from an alternative point of view some computational problems that are known to be especially difficult. This is the case of NP-hard problems, which are (strongly) supposed to require exponential time in the worst case to be solved. Since it is not feasible in practice to optimally solve these problems, optimal methods are substituted by *heuristic* algorithms that provide suboptimal solutions. Since FM face many NP-hard problems, EC can help FM to cope with them. In particular, many recent works use Genetic Algorithms or Ant Colony Optimization to deal with formal problems appearing in testing and model checking techniques.

Genetic algorithms (GAs) are search methods inspired by the mechanics of genetics and natural selection [15,6]. A GA provides an algorithmic framework for exploiting heuristics that simulates natural-evolution processes like selection and mutation. It evolves candidate solutions to problems that have large solution spaces and are not tractable to exhaustive search or traditional techniques.

Ant Colony Optimization (ACO) [11,25] simulates the behavior of real ants. ACO is a probabilistic technique that can be applied to generate solutions for combinatorial optimizations problems. It consists in copying the method used

---

[⋆] Research partially supported by projects TIN2006-15578-C02-01, PAC06-0008-6995, and CCG08-UCM/TIC-4124.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 310–317, 2009.
© Springer-Verlag Berlin Heidelberg 2009

by ants to find good paths from the colony to food sources. Ants release some amount of pheromone as they move, and this pheromone attracts other ants. When an ant finds a path to the food source, it takes the food back to the colony and repeats the path afterwards, which reinforces the pheromone trail.

In the next section we briefly introduce the FM areas that will be considered in the rest of the paper. Then in Section 3 we describe the main works developed up to now applying EC methods to these areas.

2 Formal Testing and Model Checking

The research in FM has resulted in many formal languages and in verification techniques (supported by prototype tools) to verify high-level properties in formal system descriptions. Although these methods are based on important mathematical theories, there are not many systems developed nowadays for which correctness is completely formally verified using these methods. On the other hand, the current practice of checking correctness of computing systems is based on a more informal and pragmatic approach. *Testing* is usually the predominant technique, where an implementation has to pass a number of tests which have been obtained in an ad-hoc or heuristic manner. The combination of testing and formal methods is a very promising area.

A tremendous progress has been done in developing tools and techniques for verifying requirements and design. One of the most successful approaches is *model checking* [23,5]. It consists on given an abstract simplified model of a program, test automatically whether this model meets a given specification. When combined with strict use of a formal modeling language, you can automate the verification process fairly well. The idea is the next: A model-checking tool accepts system requirements or designs (called *models*) and a property (called *specification*) that the system is expected to satisfy. If the given model satisfies given specifications then the tool outputs yes and otherwise generates a counterexample. By ensuring that the model satisfies enough properties of the system, we will increase our confidence in the correctness of the model.

For control-oriented systems, finite state machines (FSM) are widely accepted as a good, clean, and abstract notation for defining requirements and design. But a *pure* FSM is not adequate for modeling complex real-life industrial systems. We need extended finite state machines (EFSM) for modeling. Most model checking tools have their own rigorous formal language for defining models, but most of them are variants of EFSM.

3 Applications of EC to Testing and Model Checking

A number of published works (see e.g. [4,24]) have begun to examine the effective use of Artificial Intelligence (AI) for Software Engineering (SE) related activities which are inherently knowledge intensive and human-centred. On one hand, it has been identified that one of the SE areas with a more prolific use of AI techniques is software testing. The focus of these techniques involves the applications of GAs [21]. Recently, ACO is starting to be applied in software testing [9,20]. On the other hand, most model checkers found in the literature use exact deterministic algorithms to check the properties. These algorithms usually require huge amounts of computational resources if the checked model is large. The idea of exploiting heuristics for model checking has received little attention, altough this tendency is starting to change (see e.g. [1,14,2]).

Next we comment some relevant papers that apply evolutionary computation techniques to solve software testing and model checking problems.

The work [3] explores strategies that combine automated test suite generation techniques with high volume or long sequence testing. Long sequence testing repeats test cases many times, simulating extended execution intervals. These testing techniques have been found useful for uncovering errors resulting from component coordination problems, as well as system resource consumption or corruption. Coupling automated test suite generation with long sequence testing could make this approach more scalable and effective in the field.

Software testing techniques often involve developing test suites that achieve some level of *coverage* with regard to the target code. The goal is to ensure that all, or at least an important subset of the execution pathways, are exercised. Less attention has been paid to *long sequence testing* (LST) where systems are run for extended periods with test cases repeatedly executed, perhaps under varying loads for stress testing. At first glance, programmers might expect a test case to execute the same way at any point in time. Of course, a test case that has repeatedly executed successfully can suddenly fail after prolonged periods of execution. System resources can be consumed (memory leaks for example) or become corrupted, and internal clocks, counters, as well as other state-based components can become incorrect or simply overflow.

Authors describe software system failures in complex, distributed systems. These failures all occurred after the systems were deployed. One example of an extended execution error was exhibited by the Patriot missile system. This system was intended to intercept and destroy Scud missiles, but a bug in the software processing the internal clocks caused them to fail. The clocks were originally supposed to be reset frequently, but extended deployment for more than a hundred hours caused the Patriot missiles to miss their Scud targets. Authors mention another interesting examples of system failures, such as the London Ambulance Service or in the German Railway Switching.

Traditional measures of code coverage do not adequately capture the idea that test cases may begin to fail only after repeated attempts due to the changing state of complex systems. Genetic algorithms offer one such technique with several desirable characteristics for long sequence testing demonstrated by previous research. For example, automated techniques can generate large numbers of test cases. Genetic algorithms in particular provide a powerful search technique that is effective in very large search spaces as represented by system environment attributes and input parameters in the testing arena.

In the applications, the genetic algorithm is used to generate good test cases. However the notion of goodness or fitness depends on results from previous testing cycles. That is, a relative rather than absolute fitness function is used.

For the generation of test cases, a relative fitness function changes along with the population, allowing subsequent generations to take advantage of any insights collected from past results stored in the *fossil record*. This fossil record contains all previously generated test cases and information about the type of error generated, if any, and when that test case was run.

In [20] the authors propose to use *UML diagrams* and ACO for test data generation. Using the developed ACO algorithm, a group of ants can effectively explore the graph and generate optimal test data for achieving the test coverage requirement. In order to apply ACO to test case generation, it is necessary: (1) Transforming the testing problem into a graph; (2) A heuristic test for measuring the *quality* of paths through the graph; (3) A mechanism for creating possible solutions efficiently and a suitable criterion to stop solution generation; (4) An appropriate method for updating the pheromone; and (5) A transition rule for determining the probability of an ant traversing from one node to the next.

The authors consider the problem of simultaneously dispatching a group of ants to cooperatively search a directed graph G. An ant k at a vertex v of the graph is associated with a tuple (S_k, D_k, T_k, P) where S_k keeps a vertex track of the ant's route history; D_k indicates those vertices which are always connected to the current vertex v; T_k represents the direct connections of the current vertex v with the neighbouring vertices; P represents the pheromone levels at the neighbouring vertices which are feasible to an ant at the current vertex. S_k, D_k, and T_k are private for each ant, while P is shared by all ants.

The advantages of the proposed algorithm are: (1) This approach directly uses the standard UML created in software design processes; (2) The test sequence that is generated automatically is always feasible, non-redundant and achieves the all state coverage criterion. There are two problems associated with state-based software testing: (1) Some of the generated test cases are infeasible; (2) Inevitably many redundant test cases have to be generated in order to achieve the required testing coverage.

The purpose in [9] is investigating methods for deriving a suitable set of test paths for a software system. The design and the possible uses of the software system are modeled using a *Markov Usage Model* (MUM) [28] which is a versatile model description for the use of a given software product in the application field that reflects the operational distribution of the system. Furthermore, the MUM is enriched by estimates of failure probabilities, losses in case of failure and testing costs. It is common that users do not use all functions described in the specification of a product with the same frequency. This fact gives the considered problem the character of a *decision problem under uncertainty*. The authors consider the tradeoff between coverage and testing costs and try to find an optimal compromise between both using ACO. The main problem is that for determining the solution, the uncertain knowledge the developer has on possible failures and losses in case of failure needs to be quantified, and the MUM must be enriched using this information. The aim of this work is to automatically derive a suitable set of test paths through the MUM graph using ACO.

For putting it up, it is necessary to identify: (1) States of the software; (2) Possible transitions between these states; and (3) Probabilities for the possible transitions. It is convenient to use a directed graph for the representation of the MUM. The nodes are the states, the edges are the transitions, and the transition probabilities are indicated as labels assigned to the arcs.

In their computational experiments, they applied three different ACO variants. In the *standard* variant, the probability of an arc of being taken is chosen proportional to the trail level ($\tau(e)$) and an attractiveness value ($\eta(e)$). In the *variant A*, the probability that a particular edge e is chosen is proportional to $\tau(e) \cdot \eta(e)$, and inverse proportional to how often e has been traversed in all previous steps. Thus, edges that have been used frequently get a lower probability of being considered in the future, which favors diversification of routes. In *variant B*, they apply the same selection mechanism as in the standard variant, but the trail level update mechanism is changed: Only an edge that lies on the best route found so far with a minimum number of traversals in this best route gets a trail level increment. This variant also favors diversification.

The experimental results indicate that the proposed approach outperforms a greedy-type solution heuristic. Similar problems have already been investigated in the literature, mainly in the context of protocol conformance testing or user interface testing. While these approaches attempt for satisfying pre-defined coverage criteria (usually, arc coverage) they do not force complete coverage.

In [1] the authors propose the use of a new kind of ACO model (ACOhg) to refute safety properties in concurrent systems. The results state that ACOhg finds optimal or near optimal error trails in *buggy* concurrent systems with reduced resources, outperforming state-of-the-art model checking algorithms. This makes it suitable for checking safety properties in large concurrent systems, in which traditional techniques fail to find errors because of the model size.

In model checking, all possible program states are analyzed in order to prove (or refute) that the implementation satisfies a given property. The given property is typically specified using a temporal logic like Linear Temporal Logic (LTL) or Computation Tree Logic (CTL). In particular, authors use SPIN [17] (one of the best known explicit model checkers) which takes a software model codified in Promela and a property specified in LTL as inputs and transforms the model and the negation of the LTL formula into a *Büchi automata* [12] (an extension of a finite state automaton to infinite inputs) in order to perform the synchronous product of them. The resulting automaton is explored to search for a cycle of states containing an accepting state reachable from the initial state. If the cycle is found, then there exists at least one execution of the system not fulfilling the LTL property (for more details see [17]). If such kind of cycle does not exist then the system fulfills the property and the verification ends with success.

In this paper the authors propose using a new ACO model for finding counterexamples of LTL formulae in concurrent systems. The results show that, by using heuristic search, the length of the counterexamples can be shortened and the amount of memory required to obtain an error trail is reduced, allowing the exploration of larger models. Compared against the state-of-the art exhaustive methods, this novel application is able to outperform them in efficacy and efficiency. Besides, ACOhg algorithms can be used with other techniques for reducing the amount of memory required in the search such as partial order reduction, symmetry reduction, or state compression.

In [14] a framework for exploring *very large state spaces* of concurrent reactive systems is presented. It exploits application-independent heuristics using GAs to guide a state-space search towards error states. The framework is implemented in conjunction with VeriSoft [13] (a tool for exploring the state spaces of software applications composed of several concurrent processes). Their experiments show that, for finding errors in very large state spaces, a genetic search using simple heuristics can significantly outperform random and systematic searches.

The candidate solutions in this framework are finite sequences of transitions in the state space starting from the initial state. Each candidate solution is encoded by a chromosome. To evaluate the fitness of a chromosome, the GA executes the path encoded by the chromosome. This is done combining the GA with a model checker. Given the representation of a system, the model checker determines the state space to explore. The execution of a path starts in the initial state. If there are more than one possible transitions from the current state, the model checker informs the GA about the number of possibilities. The GA decodes a part of the chromosome it is processing and informs the model checker of which transition to take. The model checker checks if the state following that transition is an error state. If so, the current path is saved and the user is notified. Otherwise, the model checker repeats this process from the new state. Since a chromosome can only encode a finite number of transitions, the state space is explored up to a fixed depth. Whenever the model checker has explored a path up to this maximum depth, it prompts the GA to evaluate the fitness of the current chromosome.

In [2], the authors compare a GA against classical exact techniques. They propose a new operator for this problem, called *memory operator*, which allows the GA to explore even larger search spaces. They show that a variation on GAs combined with Java Pathfinder [16], a well known model checker, can be used to search for errors in complex applications. It is shown that they can obtain good results faster and with fewer resources than using exact search methods. While they focus on searching for deadlock, any kind of safety property violation could be searched for with their algorithm, changing only the fitness function.

To find safety property violations, model checking works by searching through all the states of a model and validating each of them against the specified properties. If all the states are valid, it is proved that those properties are valid in the model. If any of those properties fail, the model checker can give an execution path that leads to that error. In this case, an individual represents a path to a state that causes a safety property violation in the model. The path can be described as the sequence of transitions from an initial state to a final state. As the number of transitions needed to reach a goal state is unknown beforehand, authors use variable-length chromosomes to codify transition sequences.

The *memory operator* (MO) allows the path to extend indefinitely while preserving the memory required evaluating individuals. As the population evolves and grows, the first transitions in the individual tend to stabilize, but the model checker still has to evaluate them every generation. They save the resulting states of the first transitions in memory slots and use them as the starting point for next generations. The advantages are obvious: less memory and time are required to evaluate an individual and the path can maintain a constant growth without requiring more memory. There are, of course, disadvantages: part of the search space is being discarded, and a good solution might be in that part.

The algorithm is compared to standard exhaustive search algorithms. It is shown that it generally requires more time than standard algorithms to reach an error, but the error trails generated by this method are smaller and simpler to use in the debugging of the program. Besides, it can be used in larger problems, where standard methods cannot be used due to memory requirements.

Other works apply GAs for finding suitable test suites. In particular, mutation testing techniques are used in [22] and [7] to evaluate the fitness of test suites for FSMs and Haskell programs, respectively. GAs are also applied for finding temporal sequences leading a temporal system to reach a determined configuration [8]. In [26] a River Formation Dynamics approach is used for finding short tests reaching some edges or states in a FSM under the assumption that the system state can be saved and loaded later. In [27] the authors present a GA with the ability of designing specific-to-problem kernels.

4 Conclusions

In this paper we have presented an overview about the application of EC to Formal Testing and Model Checking. We have illustrated, with several examples, how theoretically based methods and heuristic methods can work together in this environment. In particular, EC methods provide efficient strategies to search for good solutions in the kind of problems appearing in Formal Methods.

References

1. Alba, E., Chicano, F.: Finding safety errors with ACO. In: GECCO 2007, pp. 1066–1073. ACM Press, New York (2007)
2. Alba, E., Chicano, F., Ferreira, M., Gomez-Pulido, J.: Finding deadlocks in large concurrent java programs using genetic algorithms. In: GECCO 2008, pp. 1735–1742. ACM Press, New York (2008)
3. Berndt, D.J., Watkins, A.: High volume software testing using genetic algorithms. In: HICSS 2005, p. 318. IEEE Computer Society Press, Los Alamitos (2005)
4. Briand, L.C.: On the many ways software engineering can benefit from knowledge engineering. In: SEKE 2002, pp. 3–6. ACM Press, New York (2002)
5. Burch, J.R., Clarke, E.M., McMillan, K.L., Dill, D.L., Hwang, L.J.: Symbolic model checking: 10ˆ20 states and beyond. Inf. Comput. 98(2), 142–170 (1992)
6. Davis, L. (ed.): Handbook of genetic algorithms. Van Nostrand Reinhold, New York (1991)
7. de la Encina, A., Hidalgo-Herrero, M., Rabanal, P., Rubio, F.: Applying evolutionary techniques to debug functional programs. In: IWANN 2009. Springer, Heidelberg (2009)
8. Derderian, K., Merayo, M., Hierons, R., Núñez, M.: Aiding test case generation in temporally constrained state based systems using genetic algorithms. In: IWANN 2009. Springer, Heidelberg (2009)

9. Doerner, K., Gutjahr, W.J.: Extracting test sequences from a markov software usage model by ACO. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 2465–2476. Springer, Heidelberg (2003)
10. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system:optimization by a colony of cooperating agents. IEEE Tran Systems,Man & Cybernetics B 26(1), 29–41 (1996)
11. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
12. Fritz, C.: Constructing büchi automata from linear temporal logic using simulation relations for alternating büchi automata. In: H. Ibarra, O., Dang, Z. (eds.) CIAA 2003. LNCS, vol. 2759, pp. 35–48. Springer, Heidelberg (2003)
13. Godefroid, P.: Verisoft: A tool for the automatic analysis of concurrent reactive software. 476–479 (1997)
14. Godefroid, P., Khurshid, S.: Exploring very large state spaces using genetic algorithms, pp. 266–280. Springer, Heidelberg (2002)
15. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Redwood City (1989)
16. Havelund, K.: Java pathfinder, a translator from java to promela. In: Dams, D.R., Gerth, R., Leue, S., Massink, M. (eds.) SPIN 1999. LNCS, vol. 1680, p. 152. Springer, Heidelberg (1999)
17. Holzmann, G.J.: The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley Professional, Reading (2003)
18. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: ICNN 1995, vol. 4 (1995)
19. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671 (1983)
20. Li, H., Lam, C.P.: Software test data generation using ant colony optimization. In: International Conference on Computational Intelligence, pp. 1–4 (2004)
21. Mcminn, P., Holcombe, M.: The state problem for evolutionary testing. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 2488–2498. Springer, Heidelberg (2003)
22. Molinero, C., Núñez, M., Andrés, C.: Combining genetic algorithms and mutation testing to generate testing sequences. In: IWANN 2009. Springer, Heidelberg (2009)
23. Palshikar, G.K.: An introduction to model checking. Technical report (2004), `http://www.embedded.com/`
24. Pedrycz, W., Peters, J.F.: Computational Intelligence in Software Engineering. World Scientific Publishing Co., Inc., Singapore (1998)
25. Rabanal, P., Rodríguez, I., Rubio, F.: Finding minimum spanning/distances trees by using river formation dynamics. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217. Springer, Heidelberg (2008)
26. Rabanal, P., Rodriguez, I.: Testing restorable systems by using RFD. In: IWANN 2009. Springer, Heidelberg (2009)
27. Rubio, G., Pomares, H., Rojas, I., Guillén, A.: Creation of specific-to-problem kernel functions for function approximation. In: IWANN 2009. Springer, Heidelberg (2009)
28. Whittaker, J.A., Thomason, M.G.: A markov chain model for statistical software testing. IEEE Transactions on Software Engineering 20(10), 812–824 (1994)

Applying Evolutionary Techniques to Debug Functional Programs[⋆]

Alberto de la Encina, Mercedes Hidalgo-Herrero, Pablo Rabanal,
and Fernando Rubio

Universidad Complutense de Madrid, 28040 Madrid, Spain
{`albertoe@sip`, `mhidalgo@edu`, `prabanal@fdi`, `fernando@sip`}`.ucm.es`

Abstract. Selecting an appropriate test suite to detect faults in a program is a difficult task. In the case of functional languages, altough there are some additional difficulties (due to the lack of state, laziness and its higher-order nature), we can also take advantage of higher-order programming to allow defining generic ways of obtaining tests. In this paper we present a genetic algorithm to automatically select appropriate criteria to generate tests for Haskell programs. As a case study, we apply the algorithm to deal with a program using red-black trees.

1 Introduction

Computational systems constructed nowadays have reached a high level of complexity. Since they may consist of a high number of complex components potentially built by different parties, no system developer can have a precise intuition of where errors could appear indeed. Thus, manual intuition-based testing procedures must be substituted by more systematic testing methods. *Formal Testing Techniques* [6,7,11] aim at providing (semi-)automatic methods to select test cases, apply these tests to the *system under test* (SUT), and draw (in-)correctness diagnoses based on collected observations. If a formal specification of what the SUT must (and must not) do is given, then we can use this specification to automatically *derive* tests capable to stimulate the SUT. The goal of these tests is traversing the SUT for some interaction sequences explicitly considered in the specification, in such a way that we can compare the responses provided by the SUT with those expected by the specification. Most formal testing techniques assume that specifications, as well as the SUT itself, can be modeled by using some simple formalism. Though this is a valid assumption for some application domains, it may not be realistic for some kinds of *software* applications, which cannot be precisely represented by these models. In fact, being provided with a precise specification of such software systems could imply the target system has been programmed *twice*. Thus, formal specifications are rarely available.

Still, methods allowing to semi-automatically extract test cases (but not from a formal specification) can be defined. For instance, we may consider available

---

[⋆] Research partially supported by projects TIN2006-15578-C02-01, PAC06-0008-6995, and CCG08-UCM/TIC-4142.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 318–326, 2009.
© Springer-Verlag Berlin Heidelberg 2009

testing utilities for imperative languages, such as Java [2,4]. However, dealing with *functional languages* have some additional difficulties: Due to laziness and the absence of *state*, tracing the effect of the system execution in functional languages may be harder than in imperative languages, and thus testing functional programs is not easy. Other characteristics of functional languages may be an advantage, rather than a handicap. For instance, the *high order computation* allows functions to modularly and easily manipulate other functions. Thus, making a functional program (in our case, a tester program) control the execution of another program (in our case, the SUT) is easier.

In this paper we propose a test derivation method to test programs written in a functional language, in particular *Haskell* [8], the *de facto* standard language of the lazy-evaluation functional community. This method allows to (1) automatically construct data values of some type such that these values have a *high capability* of detecting faults in the SUT, (2) apply these values to the SUT, and (3) compare the observed values with some expected values in such a way that, if an incorrect response is observed, then an *incorrectness diagnosis* is drawn. Our method to find data values with high fault detection capabilities will be based on integrating *Mutation Testing* techniques [5,1] with an Evolutionary Computation approach (in particular, *Genetic Algorithms* [3,10,12]). The Mutation Testing techniques allow to *measure* the capability of a set of tests to detect faults in a given kind of systems. They work as follows: Given a correct program belonging to the same domain as the SUT and dealing with the same kind of data, we generate some *mutants* of this program, that is, some programs whose code matches the original program code for all instructions but one (or a few of them). In these instructions, *mutations* (i.e. simple code modifications) are introduced (e.g. the boolean condition of an *if* statement is slightly modified, an instruction is duplicated or deleted, etc). Then, we say that the considered set of tests is *good* if, after applying these tests to all constructed mutants, responses allow to distinguish many (incorrect) mutants from the original correct program. That is, the suitability of a test suite is measured in terms of the capability of tests in the suite to *kill* wrong mutants.

Though Mutation Testing techniques allow to *compare* test suites, a method to find *good* test suites according to this criterion must be provided. This will be achieved by running a genetic algorithm. Our test derivation algorithm for Haskell programs will depend on some parameters, and these parameters will make the algorithm construct some test suites or others. We will use the genetic algorithm to *adjust* these parameters: Given a set of random combinations of parameters, we will measure the fitness of each combination by running the test derivation algorithm for these parameters and next applying the constructed test suite to our mutants. Then, the fitness of each parameter combination will be assessed in terms of the proportion of wrong mutants killed by this suite. After calculating the fitness of all parameter combinations, the best combinations will be taken, and these combinations will be *crossed over*, *mutated*, and *reproduced*. The procedure will be repeated for the new generation of parameter combinations, and so on.

We will present our test derivation method for Haskell programs and we will illustrate it with some examples. In particular, the algorithm integrating the mutation testing method with the genetic approach will be described, and some experimental results will be reported.

2 Creating Mutants

In this section we introduce a method to create mutations of a given Haskell function. Notice that we are not yet dealing with mutations of genotypes, but with the generation of mutations of the original program. Let us remind that our aim is creating a testing method capable of distinguishing correct functions and their incorrect mutants. Thus, as a first step to allow training our testing method, we need a simple way to check it against a relatively large number of mutations of programs.

To create mutations of a given function, we need to consider how to create mutations of its input parameters and how to create mutations of the results of the function. Thus, we start considering simple data types instead of functional data types. After considering those simple data types, we will show how to automatically infer mutations for functions by considering its inputs and output.

For each data type that can have mutations, we need to provide a method to create its mutations. Moreover, the interesting mutations that can occur in a data type are not exactly the same when it is used as input of a function or as output of it (although they use to be similar). Hence, we create a new data class with two basic operations (`mutOut` and `mutIn`) to create both input and output mutations:

```
class Mutant a where
  mutOut :: a -> [a]
  mutIn :: a -> [a]
  nOut :: (b -> a) -> Int
  nIn :: (a -> b) -> Int
```

where auxiliary functions `nOut` and `nIn` are only needed to deal with infinite lists. For the sake of clarity, we will not describe them and we will not use them in the following definitions.

Let us consider the most simple example: Boolean values. There are not many combinations of errors that the programmer can perform when passing parameters to a boolean function. In fact, for a single parameter, there are only three possible mutations (in addition to the identity mutation): (1) the parameter is negated; (2) the parameter is always True; or (3) the parameter is always False. In fact, the same mutations can take place when using it as output of a function. Thus, the instance we need to create is the following:

```
instance Mutant Bool
  where mutOut b = [b,not b,True,False]
        mutIn b = [b,not b,True,False]
```

Obviously, in case that several parameters are considered, then much more combinations are possible. But that issue can be easily solved in a functional setting by defining a generic way of combining them. In particular, we can create a new instance of the class `Mutant` to deal with any pair of values that are mutable (not only booleans):

```
instance (Mutant a, Mutant b) => Mutant (a,b)
  where mutOut (x,y) = [(x',y') | x'<- mutOut x, y'<- mutOut y]
        mutIn (x,y) = [(x',y') | x'<- mutIn x, y'<- mutIn y]
```

Let us consider a not so trivial data type. In order to do that, we use *red-black trees*. This data type can be defined in Haskell as follows:

```
data Color = R | B
data RB a = E | T Color (RB a) a (RB a)
```

where the first line introduces a data type `Color` that can be either Red or Black, while the second line defines a red-black tree as an empty tree or a tree with a color, a left child-tree, an element, and a right child-tree. As in the previous case, we have to provide a way to create mutations when it is used as input and as output of a function:

```
instance Mutant (RB a)
  where mutOut = mutate
        mutIn  = mutateRec
```

When this data type is the output of a function, the most relevant mutations are related to non empty trees. In this case, there can be errors in the color of the tree or on any of the cildren. Thus, we only need to combine the different possible mutations (those related to the color and those related to the children):

```
mutate :: RB a -> [RB a]
mutate E = replicate nOutRB E
mutate (T c t1 x t2)
  = [T c' t1' x t2' | c'<-[c,inverse c],t1'<-[t1,t2],t2'<-[t2,t1]]++[E]
```

However, in the case of using it as input of a function, the mutations must reflect possible ways in which the definition of the function fails when considering the inputs. Notice that, in case a function is not dealing properly with the pattern matching of the inputs, such an error does not only occur in the top-level structure of the input value, but also on all the recursive levels. That is, in case the definition of the function confuses the left and the right trees, it will confuse them in all the levels of the tree, not only on the first one. Thus, the same mutations must be applied to all the recursive levels of the data type:

```
mutateRec E = replicate nInRB E
mutateRec t = [mapT (fc!!cc) (fs!!c12) t | cc<-[0..3],c12<-[0..3]]
  where fc = [id,inverse,\_->B,\_->R]
        fs = [id,\(x,y)->(x,x),\(x,y)->(y,y),\(x,y)->(y,x)]
```

Let us remark that, when dealing with any other recursive data type, we must proceed exactly in the same way. In fact, it is straightforward to define functions mutOut and mutIn for any instance.

2.1 Generating Mutations of Functions

Once a `Mutant` class is defined, we can create functions working with any instance of the class. In particular, our main interest is to create mutations of functions. This can be done for any function having either as input or as output parameter a mutable type (or several of them). Let us start by considering the case where the output of the function is mutable, but not the input. In this case, we need to define a function with the following signature:

```
mutatorOuts :: Mutant a => (b -> a) -> [b -> a]
```

That is, given a function whose output is mutable, it returns a list of functions of the same type. Notice that using a functional language allows to define such a function, as it is just a higher-order function. However, it is not possible to define it in typical imperative languages.

The generation of the list of functions will be done in two steps. First, we will generate a single function that returns a list of values:

```
mut1o :: Mutant a => (b -> a) -> (b -> [a])
mut1o f x = mutOut (f x)
```

and then, we will extract a list of functions by considering the list of values returned by function `mut1o`:

```
mut2 :: (b -> [a]) -> [b -> a]
mut2 f = [\x -> (f x)!!i | i<-[0..]]
```

Thus, in order to obtain our function `mutatorOuts` we just need to apply function `mut2` after applying `mut1o`:

```
mutatorOuts f = take (nOut f) (mut2 (mut1o f))
```

where we use function `nOut` to obtain only a finite relevant portion from the infinite list produced by `mut2`.

Following an analogous procedure, we can also define a function `mutatorIn :: Mutant a => (a -> b) -> [a -> b]` creating mutations of functions when the input parameter is mutable. In fact, we only need to modify function `mut1o`, as function `mut2` can be used without any modification.

```
mut1i :: Mutant a => (a -> b) -> (a -> [b])
mut1i f t = [f t' | t' <- mutIn t]
```

Obviously, in case both the input and the output are mutable, we can combine all the possible mutations by considering function `mutatorInOut :: (Mutant a, Mutant b) => (a -> b) -> [a -> b]` that is defined in the same way as in the previous cases, but using a new function for the first step of the transformation:

```
mut1InOut :: (Mutant a, Mutant b) => (a -> b) -> (a -> [b])
mut1InOut f t = concat [mutOut (f t') | t' <- mutIn t]
```

Let us remark that the previous higher-order functions are defined only once, and they are automatically available to create mutants of any function dealing with mutable data types.

3 Creating Test Cases by Using a Genetic Algorithm

In order to create appropriate test cases for a program, we can take advantage of the previously described `Mutant` class. In case the program uses some data structures of any type belonging to class `Mutant`, we can create mutations of the program by using the facilities described in the previous section. Then, we need to find appropriate test cases to detect the possible mutations. The difficult part now consists in deciding appropriate *criteria* to obtain those test cases. We will do it by using a genetic algorithm. Thus, we will start by defining a list of properties that will be used to generate tests. Obviously, for each property, many concrete values (even infinite) are possible (e.g. in the case of red-black trees, a criterium can be the depth of the tree, or the proportion of red nodes, etc). Hence, a genotype (of type `TestGene`) will define such concrete values.

In our method, for each `TestGene` we provide a method to obtain a list of concrete tests fulfilling all the properties defined in the genotype. This task is quite easy, as we can take advantage again of the type class `Mutant` defined in the previous section. Thus, we only need to create a list of test candidates and then filter them to select only those fulfilling the properties. Notice that, due to lazy evaluation, filtering the list of candidates is done while starting to create such list. Thus, bad-candidates are discarded before actually constructing them. Our algorithm works as follows:

1. A list of mutants of the original *correct* specification is initially created to assess the effectivity of each test.
2. We randomly create an initial population `initialGenes :: [TestGene]`
3. For each genotype of our population, we randomly create a list of concrete tests. Thus, we have a list of lists of tests (one list of tests for each genotype).
4. We automatically apply the tests to all available mutants of the specification introduced (i.e. the list of mutations computed in step 1). The quality of each test will be given by the number of incorrect mutants that it is able to detect, while the quality of each genotype will be given by the total number of incorrect mutants detected by any of its instances.
5. We update our population by selecting only those genotypes that obtained the best results in the present step. Then, following a typical GA approach, the population is enlarged by considering mutations of them and by using crossover.
6. We repeat the process from step 3 on, until a given number of iterations is obtained.

The first and third steps are done by using the techniques described in the previous section to obtain mutants of a function, while the second step is just a simple random creation of simple values. In fact, the list to be defined in the second step could also be created in an *ad hoc* way in case the programmer prefers to do so.

Regarding the fourth step, we can check how many mutants have been detected by using a higher-order function to test the results of the original function against the results of each of the mutants. In particular, function `testerG1` returns a list of booleans indicating for each mutant whether a single test has detected the mutation or not. Then, function `testerG` applies `testerG1` for each test case of a genotype and computes the number of mutants that have not been detected by any test case. The lower this value is, the better the test suite is. The source code of these functions is as follows:

```
testerG1 :: Eq b => (a -> b) -> [a -> b] -> a -> [Bool]
testerG1 f fs t = map (\g->f t == g t) fs

testerG :: Eq b => (a -> b) -> [a -> b] -> [a] -> Int
testerG f fs ts
  = length (filter id (map and (transpose (map (testerG1 f fs) ts))))
```

In step 5, by taking into account the values returned by `testerG` for the test cases associated to each `TestGene`, it is trivial to select those with the lowest values. Then, some mutations[1] are randomly introduced in some elements of the selected population, and the elements are also reproduced by combining their characteristics. The concrete mutations are obtained by randomly selecting an element of the possible list of mutations (available due to class Mutant), while the crossover of two elements is trivially defined by considering a function `crossover :: [Bool] -> TestGene -> TestGene -> TestGene` that combines two genotypes by using a list of booleans (that was randomly obtained) to decide for each gene whether it must be obtained from the first genotype or from the second one.

As it usually happens when dealing with genetic algorithms, the size of the population, the number of iterations, and the (small) propensity of gene mutations in each iteration have to be fixed for each concrete problem.

3.1 A Concrete Example

As a running example, we have used the red-black trees described in the previous section. Suppose that we want to test a (potentially faulty) function dealing with such trees. In order to adjust the scheme to a concrete example, the first thing we need to do is to define the genotype to be used to create the tests. In our case, we consider the following type: `type TestGene = (Bool,Int,(Int,Int),Int)`. The boolean value represents whether we are interested in testing empty trees or not. The first integer represents the maximum depth of the trees to be tested. The tuple of integers represents the minimum and maximum percentage of red nodes that must appear in each test tree. Finally, the last integer represents the maximum unbalance that we allow between subtrees.

After that, we need to adjust the size of the population, the number of iterations, and the percentage of mutations to be introduced in each step. In our case, population size is 10, the number of iterations is 100, and the mutations percentage is 5%.

---

[1] Let us remark that these mutations are the typical mutations of genetic algorithms, not the generation of mutants of mutation testing.

We have performed several executions with the previously commented parameters. Each of the executions used a different function dealing with read-black trees, where some functions were simple calls to basic operations (e.g. insert or remove an element) while in other cases we used more complex functions combining several simple functions. Anyway, the results obtained in any case are very similar: The genotypes surviving the process usually restrict the depth of the test trees to two (there is only one exception on a very concrete ad-hoc problem, where trees of depth three were preferred). That is, it is enough to use *small* trees to detect most of the possible errors, although it is not enough to have a single recursive call. Notice that larger trees could detect the same number of mutations, but they require more time to test them. Thus, the algorithm chooses to use smaller tests in case they provide the same results.

Regarding the other properties codified in the genotypes, it seems that the percentage of red nodes is not very relevant, provided that there is at less one red and one blue node. However, it is important to remark that the maximum balance difference between subtrees is only one in nearly all the executions.

4 Conclusions and Future Work

In this paper we have presented a genetic algorithm such that, given a Haskell program, it (semi-)automatically obtains an appropriate test suite for testing this program, as well as other similar programs dealing with the same kind of input data. First, some program mutants are obtained by automatically introducing some *mutations* in the program. These mutations simulate the kind of faults a programmer could make. Then we run a genetic algorithm to automatically select some *criteria* allowing to produce good test suites. The fitness of criteria is assessed in terms of mutants: A set of criteria is good if test cases generated according to these criteria allow to *kill* a high number of mutants (i.e. they allow to detect wrong answers in these mutants). In order to use the proposed testing utility, a programmer just has to provide some simple (Haskell) functions fitting into the required interfaces. Thus, the required set up for using our method is simple and not very time-consuming. We have illustrated this with a case study dealing with red-black trees, and we have reported some experimental results.

We are currently working on developing alternative strategies to generate test cases. In particular, we are trying to improve the efficiency of the previously described genetic algorithm by developing a method based on river formation dynamics [9].

References

1. Bottaci, L., Mresa, E.S.: Efficiency of mutation operators and selective mutation strategies: An empirical study. Software Testing, Verification and Reliability 9(4), 205–232 (1999)
2. Cheon, Y., Leavens, G.T.: A simple and practical approach to unit testing: The JML and JUnit way. In: Magnusson, B. (ed.) ECOOP 2002. LNCS, vol. 2374, pp. 231–255. Springer, Heidelberg (2002)

3. Davis, L. (ed.): Handbook of Genetic Algorithms. Van Nostrand Reinhold (1991)
4. Do, H., Rothermel, G., Kinneer, A.: Prioritizing JUnit test cases: An empirical assessment and cost-benefits analysis. Empirical Software Engineering 11(1), 33–70 (2006)
5. Howden, W.E.: Weak mutation testing and completeness of test sets. IEEE Transactions on Software Engineering 8, 371–379 (1982)
6. Lee, D., Yannakakis, M.: Principles and methods of testing finite state machines: A survey. Proceedings of the IEEE 84(8), 1090–1123 (1996)
7. Petrenko, A.: Fault model-driven test derivation from finite state models: Annotated bibliography. In: Cassez, F., Jard, C., Rozoy, B., Dermot, M. (eds.) MOVEP 2000. LNCS, vol. 2067, pp. 196–205. Springer, Heidelberg (2001)
8. Peyton Jones, S.L., Hughes, J.: Report on the programming language Haskell 98. Technical report (February 1999), `http://www.haskell.org`
9. Rabanal, P., Rodríguez, I., Rubio, F.: Finding minimum spanning/distances trees by using river formation dynamics. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217, pp. 60–71. Springer, Heidelberg (2008)
10. Reeves, C.R., Rowe, J.E.: Genetic Algorithms: Principles and Perspectives: a Guide to GA Theory. Kluwer Academic Publishers, Dordrecht (2002)
11. Rodríguez, I., Merayo, M.G., Núñez, M.: $\mathcal{HOTL}$: Hypotheses and observations testing logic. Journal of Logic and Algebraic Programming 74(2), 57–93 (2008)
12. Rothlauf, F., Goldberg, D.E.: Representations for Genetic and Evolutionary Algorithms. Physica-Verlag (2006)

Aiding Test Case Generation in Temporally Constrained State Based Systems Using Genetic Algorithms[⋆]

Karnig Derderian[1], Mercedes G. Merayo[2],
Robert M. Hierons[1], and Manuel Núñez[2]

[1] Department of Information Systems and Computing, Brunel University
Uxbridge, Middlesex, UB8 3PH United Kingdom
derderian@karnig.co.uk, rob.hierons@brunel.ac.uk
[2] Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid, Madrid, Spain
mgmerayo@fdi.ucm.es, mn@sip.ucm.es

Abstract. Generating test data is computationally expensive. This paper improves a framework that addresses this issue by representing the test data generation problem as an optimisation problem and uses heuristics to help generate test cases. The paper considers the temporal constraints and behaviour of a certain class of (timed) finite state machines. A very simple fitness function is defined that can be used with several evolutionary search techniques and automated test case generation tools.

1 Introduction

Testing is an important part of the system development process that aims to increase the reliability of the implementation. Unfortunately, testing can be very expensive and difficult. In particular, the problem of generating test sequences for real-time systems is not trivial. One of the main problems to overcome is that it is not enough to test that the implementation system is doing what it is supposed to do, but it is necessary also to test that it is also taking the specified time to complete. In addition, the tests applied to implementations have to consider *when* to test [2]. This paper addresses the issues related to generating test sequences for temporally constrained FSM based systems. It focuses on generating transition paths with specific properties that can, in turn, be used to generate test inputs. Taking as a first step our previous work [3], where genetic algorithms (GAs) were used to aid the test sequence generation of FSMs, the problem of generating these paths is represented as a search problem and GAs are used to help automate test data generation. Other approaches to generate test data using GAs are reported in [4,5]

[⋆] An extended version of this paper, including a case study, can be found in [1]. Research supported by the Spanish projects WEST/FAST (TIN2006-15578-C02-01) and MATES (CCG08-UCM/TIC-4124).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 327–334, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Ordinary FSMs are not powerful enough for some applications where extended finite state machines (EFSMs) are used instead. EFSMs have been widely used in the telecommunications field, and are also now being applied to a diverse number of areas. Here we consider EFSMs with the addition of *time*: Timed EFSMs (TFSMs). In EFSMs, a transition path (TP) represents a sequence of transitions in M where every transition starts from the state where the previous transition finished. In order to execute a TP it is necessary to satisfy all of the transition guards involved, in addition to using a specific input sequence to trigger these transitions.

When FSM based systems are tested for conformance with their specification, often a fault can be categorised as either an *output fault* (wrong output is produced by a transition) or a *state transfer fault* (the state after a transition is wrong). When TEFSMs are tested for conformance there are time related faults that arise when a transition within the implementation takes longer to complete than the time specified by the TEFSM. Thus, test sequence generation is more complex than it is for FSMs. In FSMs all paths are feasible since there are no guards and actions do not affect the traversal [6]. With TEFSMs, however, in order to trigger the transition path it is necessary to satisfy the transition guards. A transition guard may refer to the values of the internal variables and the input parameters, which in turn can assume different values after each transition. Some transition paths might have no conditions, some might have conditions that are rarely satisfied and some transition paths will be infeasible. The existence of infeasible transition paths creates difficulties for automating the test generation process for TEFSMs. One way of approaching the test sequence generation problem is to abstract away the data part of the TEFSM and consider it as an FSM on its own. However, a transition sequence for the underlying FSM of a TEFSM is not guaranteed to be feasible for the actual TEFSM nor to satisfy the temporal constraints. This leads to the problem that an input sequence will only trigger a specific TP in a TEFSM if all the transition guards allow this and so an input sequence generated on the basis of the FSM may not follow the required path in the TEFSM. Another approach is to expand a TEFSM to an FSM and then use the techniques used for FSMs. However this can lead to a combinatorial explosion.

The problem of TP feasibility in EFSMs was considered in [7]. This paper uses the TP feasibility approach proposed in that work and extends it to consider TEFSMs and problems associated to testing the compliance of an implementation to its temporal aspects of the specification. In addition to estimating the feasibility of a TP, in this paper we examine how to consider the temporal properties of a TP and help in related test case generation. In order to generate a test case for a TEFSM M we can first consider the properties of the TP that this test case is expected to trigger. The general problem of finding a (an arbitrary) feasible transition sequence for a TEFSM is uncomputable, as is generating the necessary input sequence to trigger such a transition sequence. The task of finding a transition sequence with particular temporal conditions complicates the problem even further.

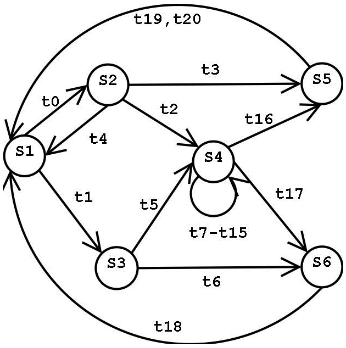


Fig. 1. Class 2 transport protocol TEFSM M_1. The transition table is on Fig. 2.

2 Timed Extended Finite State Machine (TEFSM) Model

To represent our timed EFSM assume that the number of different variables is m. If we assume that each variable x_i belongs to a domain D_i thus the values of all variables at a given point of time can be represented by a tuple belonging to the cartesian product of $D_1 \times D_2 \times ... \times D_m = \Delta$. Regarding the domain to represent time we define that time values belong to a certain domain **Time**.

Definition 1. *A TEFSM M can be defined as* $(S, s_0, V, \sigma_0, P, I, O, T, C)$ *where* S *is the finite set of logical states,* $s_0 \in S$ *is the initial state,* V *is the finite set of internal variables,* σ_0 *denotes the mapping from the variables in* V *to their initial values,* P *is the set of input and output parameters,* I *is the set of input declarations,* O *is the set of output declarations,* T *is the finite set of transitions and* C *is such that* $C \in \Delta$*.*

A transition $t \in T$ *is defined by* $(s_s, g_I, g_D, g_C, op, s_f)$ *where* s_s *is the start state of* t*;* g_I *is the input guard expressed as* (i, P^i, g_{P^i}) *where* $i \in I \cup \{NIL\}$*;* $P^i \subseteq P$*; and* g_{P^i} *is the input parameter guard that can either be NIL or be a logical expression in terms of variables in* V' *and* P' *where* $V' \subseteq V$*,* $\emptyset \neq P' \subseteq P^i$*;* g_D *is the domain guard and can be either NIL or represented as a logical expression in terms of variables in* V' *where* $V' \subseteq V$*;* $g_C : \Delta \rightarrow$ ***Time*** *is the time the transition needs to take to complete; op is the sequential operation which is made of simple output and assignment statements; and* s_f *is the final state of* t*.*

A TEFSM M is deterministic if any pair of transitions t *and* t' *initiating from the same state* s *that share the same input* x *have mutually exclusive guards. A TEFSM M is strongly connected if for every ordered pair of states* (s, s') *there is some feasible path from* s *to* s'*. A configuration for a TEFSM M is a combination of state and values of the internal variables* V *of* M*.*

We assume that any TEFSM considered is deterministic and strongly connected. For example, consider the Class 2 transport protocol [8] represented as a TEFSM in Figure 1. There are two transitions, t_2 and t_3, initiating from state S_2 with the same input declaration. However, they have mutually exclusive conditions: $opt\_ind \leq opt$ and $opt\_ind > opt$, respectively. For example, consider again Figure 1. There is a TP from the initial state S_1 to every other state.

The label of a transition in a TEFSM has two guards that decide the feasibility of the transition: the input guard g_I and the domain guard g_D. In order for a transition to be executed g_I, the guard for inputs from the environment must be satisfied. Some inputs may carry values or specific input parameters and M may guard those values with the input parameter guard g_P. Hence the values of the input parameters may determine whether a transition is executed and affect the output of M. The input guard $(NIL, \emptyset, NIL)$ represents no input being required (spontaneous transition). g_D is the guard, or precondition, on the values of the system variables (e.g. $v > 4$, where $v \in V$). Note that in order to satisfy the domain guard g_D of a transition t, it might be necessary to have taken some specific path to the start state of t. *op* is a set of sequential statements such as $v := v+1$ and $!o$ where $v \in V$, $o \in O$ and $!o$ means 'output o to the environment'. Literal outputs (output directly observable by the user) are denoted with ! and output functions (an output function may produce different output depending on the parameters it receives) without it (e.g. $!o$ and $u(v)$). The operation of a transition in a TEFSM has only simple statements such as output statements and assignment statements, no branching statements are allowed. In a TEFSM the time a transition takes to complete, represented by g_C, is also important. This time can depend on the current values of V' and P^i, where $V' \subseteq V$ and $P^i \subseteq P$.

We assume that none of the spontaneous transitions in a TEFSM are without any guards, $g_I = (NIL, \emptyset, NIL)$ and $g_D = NIL$, because they would be uncontrollable. When a transition in a TEFSM is executed, all the actions of the operation specified in its label are performed consecutively and only once and the transition must not take more than $g_C(C)$ time units to perform. The transition is considered to have failed (in terms of compliance to the specification or be considered void) if it is not completed within the required time. Such transitions may be considered void if they take longer than expected in some systems and in others they might still be executed even though they took longer than expected.

Now consider the problem of finding an input sequence that triggers a timed feasible transition path (TFTP) from state s_i to state s_j of a TEFSM M.

Definition 2. *A timed feasible transition path (TFTP) for state s_i to state s_j of a TEFSM M is a sequence of transitions initiating from s_i that is feasible for at least one combination of values of the finite set of internal variables V (configuration) of M and ends in s_j.*

In a transition path for FSMs each transition can be identified, and thus represented, by its start state and input (s_s, i). However, with TEFSMs this information is not sufficient because there can be more than one transition sharing the same start state and input due to having mutually exclusive guards. Instead, a transition t in a TEFSM M can be identified from its start state, input

t	input	output	feasibility rank	temporal rank
t_0	ICONreq	!CR	0	0
t_1	CC	!ICONconf	0	0
t_2	T_expired	!CR	2	0
t_3	T_expired	!IDISind	1	1
t_4	IDATreq	DT	0	0
t_5	AK		6	1
t_6	AK		6	1
t_7	AK	DT	5	0
t_8	AK	!IDISind	4	0
t_9	T_expired	DT	3	0
t_{10}	T_expired	!IDISind	2	1
t_{11}	DR	!IDISind	0	2
t_{12}	DR	!IDISind	0	2
t_{13}	DR	!IDISind	0	2
t_{14}	DR	!IDISind	0	2

Fig. 2. Temporal constraint ranking and feasibility ranking for all transitions in M_1

declaration, input parameter, the input parameter guard and the domain guard $(s_s, i, P^i, g_{P^i}, g_D)$.

A transition in a transition path of a TEFSM can be identified by a tuple (s_s, i, g_{P^i}, g_D) in which s_i is its start state, i is its input, g_{P^i} is its input guard and g_D is its domain guard. The input parameter P^i is not required in order to be able to uniquely identify a transition in M. Note how in this case some transitions with different domain guards share a common input predicate guard.

Since not all transitions in TEFSMs have input parameter guards and domain guards, they can be categorised in the following way: *simple transitions* are those transitions that have *no* input parameter guard and *no* domain guard, that is, $g_{P^i} = NIL$ and $g_D = NIL$; g_{P^i} *transitions* are those transitions that *have* input parameter guard but *not* a domain guard, that is, $g_{P^i} \neq NIL$ and $g_D = NIL$; g_D *transitions* are those transitions that *have* a domain guard but *not* an input parameter guard, that is, $g_D \neq NIL$ and $g_{P^i} = NIL$; g_{P^i}-g_D *transitions* are those transitions that *have* both an input parameter guard and a domain guard, that is, $g_{P^i} \neq NIL$ and $g_D \neq NIL$. Next we introduce some concepts that will be used later.

Definition 3. *An input sequence (IS) is a sequence of input declarations $i \in I$ with associated input parameters $P^i \subseteq P$ of a TEFSM M.*

A predicate branch (PB) is a label that represents a pair of g_{P^i} and g_D for a given state s and input declaration i. A PB identifies a transition within a set of transitions with the same start state and input declaration.

An abstract input sequence (AIS) for M represents an input declaration sequence with associated PBs that triggers a TP in the abstracted M.

PBs can label conditional transitions and be used to help simulate the behaviour of a potential input sequence for a TEFSM without the feasibility restrictions. The advantages of using AIS and simulating the execution of a TEFSM is that the configuration of the TEFSM is not considered. Hence, transition traversal evaluations that can be used to estimate the characteristics of a TP can be done without complex computation.

3 Using Genetic Algorithms to Aid Test Case Generation

In order to achieve our objectives we require an easy to compute fitness function that estimates the feasibility of a TP but also helps us to test our temporal constraints. Computing the actual feasibility of a transition path is computationally expensive, so we need a method to estimate this.

Some transition paths consist of transitions with difficult to satisfy guards. It is always possible to execute a *simple* transition in a transition path since there are no guards to be satisfied. The presence of g_{P^i} transitions could render a transition path infeasible because of its input predicate guard. However the conditions of this guard are more likely to be satisfiable than domain guards because the values of the input parameters $P^i \subseteq P$ can be chosen. When these conditions depend also on some internal variables $V' \subseteq V$ then such g_{P^i} transitions might not be easier to trigger than g_D transitions. In some cases the execution of a g_D transitions could require reaching its start state through a specific transition path. The feasibility of g_{P^i}-g_D transitions depends on both issues outlined above for g_{P^i} transitions and g_D transitions.

Since the presence of g_D transitions and g_{P^i}-g_D transitions seem to increase the chance of a transition path being infeasible such transitions can be penalised and *simple* transitions rewarded in a TP. In this paper we use the same feasibility estimation framework as in [7].

We may consider the temporal constraints of the transitions in a TP. For example a test sequence may be needed to stress test the temporal aspects of the system. Adequate test cases should be feasible (in order to be useful) and may focus on transitions with complex temporal constraints. However since the temporal constraints for every transition of M are dependant on the configuration of M (the current values if the internal variables) then it is difficult to know the exact temporal constraints without executing the system and verifying the configuration of M. If the temporal constraints for M are listed in a table then we can analyse the constraints and categorise different transitions in a similar way as we classified them according to their guards above. However if the temporal constraints are represented by one or more formulas the complexity of analysing all the possibilities the problem may become intractable.

We may try to estimate the different temporal constraints associated with each transition according to how the temporal constraint is defined. Note that the same transition may have different temporal constraints depending on the values of the internal variables of M. Some transitions may not have temporal constraints at all, while others might have fixed temporal constraints that are not dependant on the configuration of M. Other transitions may have temporal constraints that are expressed using tables while some may have the temporal constraints represented using formulas.

Based on these observations we may classify the transitions in a TEFSM M in the following way: *no-time transitions* are those transitions that have *no* temporal constraints; *fixed-time transitions* are those transitions that *have* temporal constraints that *are not* effected by the values of the internal variables V of M; *known-time transitions* are those transitions that *have* temporal constraints that *are* effected by the values of the internal variables V of M, but presented in an easy to analyse way; *variable-time transitions* are transitions that *have* temporal constraints that *are* effected by the values of the internal variables V of M, but are presented by one or more formulas and the temporal constraints are not easy to analyse without considering a subset of all the configurations of M.

A *transition ranking* process is completed before the fitness function can be used. This process first ranks each transition of the EFSM according to how many penalty points are assigned to the transition guards. A *simple* transition gets the highest rank (i.e. lowest amount of penalty points), a g_{Pi} transition is ranked next, etc. Transitions that have the same number of penalty points get the same rank. This algorithm in essence sorts $|T|$ elements and has complexity $O(|T| \cdot log|T|)$ where $|T|$ is the number of transitions in M. Then the process ranks each transition according to its temporal constraint category. In our case, if we are attempting to stress test the implementation then we can argue that $variable-time$ transitions can potentially have the most complex temporal constrains hence be ranked highest (i.e. lowest amount of penalty points), $known-time$ transitions can be ranked next as they still depend on the internal variables of M, and so on. The order can be reversed if the aim is to find a test sequence that will most likely work. Such test cases may be used in early stages of software development.

Definition 4. *The fitness is a function that given a TP of a TEFSM M, sums the penalty points (assigned through the transition ranking process for M and the temporal constrain ranking for M) for each of the transition of the TP.*

In our approach the two rankings are given equal weight before being combined, following the conclusions of a similar experiment in [7] where different weights were used for a similar multi-optimisation problem. The fitness algorithm used in this work can be used to reward a potential solution to a TP generation problem according to the combined ranks of the transitions in the sequence. This function reflects the belief that the fewer constraints a sequence contains, the more likely it is be feasible and the less we can analyse the temporal constraints of a transition, the more likely it is that they are more complex.

Estimating the feasibility of a TP is just the first part of the more difficult problem of generating actual IS for a TFTP that do not always represent the shortest path between two states. Also there may be other computationally inexpensive analysis of a TP that can be added to the existing fitness functions to make it more accurate. In our approach we focus on evaluating our feasibility and complex temporal conditions estimation fitness function.

As reported in the extended version of this paper [1], the two studied GAs performed well and generated very similar results. This indicates that the fitness function and the TP representation represent the problem of TFTP generation reasonably well.

4 Conclusions and Future Work

This paper defines a computationally inexpensive method to address an important problem in test data generation for TEFSMs. It extends some work on feasibility of EFSMs [7] and considers temporal constraints. We defined the problem of finding a transition sequence that is likely to be feasible and to satisfy some temporal criteria as a search problem. We defined a computationally efficient fitness function that is used to guide a GA. In the extended version of the paper [1] a case study of a communication protocol shows how the defined fitness function yields some positive results when GA search is applied to the problem.

Future work may focus on refining the fitness function to take into account loops and other difficulties to estimate transitions. Further analysis of the conditions might also help. Further evaluation of the fitness function on other larger TEFSMs may also be beneficial to evaluate further how well the method scales. Different TEFSMs may present the need for alternative temporal constraint classification, which will be very interesting to investigate.

References

1. Derderian, K., Merayo, M.G., Hierons, R.M., Núñez, M.: Aiding test case generation in temporally constrained state based systems using genetic algorithms (2009), `http://kimba.mat.ucm.es/manolo/papers/aiding-extended.pdf`
2. Merayo, M., Núñez, M., Rodríguez, I.: Generation of optimal finite test suites for timed systems. In: 1st IEEE & IFIP Int. Symposium on Theoretical Aspects of Software Engineering, TASE 2007, pp. 149–158. IEEE Computer Society Press, Los Alamitos (2007)
3. Derderian, K., Hierons, R.M., Harman, M., Guo, Q.: Automated Unique Input Output Sequence Generation for Conformance Testing of FSMs. The Computer Journal 49(3), 331–344 (2006)
4. Jones, B.F., Eyres, D.E., Sthamer, H.H.: A strategy for using genetic algorithms to automate branch and fault-based testing. The Computer Journal 41(2), 98–107 (1998)
5. Michael, C.C., McGraw, G., Schatz, M.A.: Generating software test data by evolution. IEEE Transactions on Software Engineering 27(12), 1085–1110 (2001)
6. Duale, A.Y., Uyar, M.Ü.: Generation of feasible test sequences for EFSM Models. In: TestCom 2000: Proceedings of the IFIP TC6/WG6.1 13th International Conference on Testing Communicating Systems, Deventer, The Netherlands, pp. 91–112. Kluwer, B.V (2000)
7. Derderian, K.: Automated test sequence generation for Finite State Machines using Genetic Algorithms. PhD thesis, Brunel University (2006)
8. Ramalingom, T., Thulasiraman, K., Das, A.: Context independent unique state identification sequences for testing communication protocols modelled as extended finite state machines. Computer Communications 26(14), 1622–1633 (2003)

Creation of Specific-to-Problem Kernel Functions for Function Approximation

Ginés Rubio[1], Héctor Pomares, Ignacio Rojas, and Alberto Guillén

Department of Computer Architecture and Technology, University of Granada,
C/ Periodista Daniel Saucedo sn, 18071 Granada (Spain)
`grubio@atc.ugr.es`

Abstract. Although there is a large diversity in the literature related to kernel methods, there are only a few works which do not use kernels based on Radial Basis Functions (RBF) for regression problems. The reason for that is that they present very good generalization capabilities and smooth interpolation. This paper studies an initial framework to create specific-to-problem kernels for application to regression models. The kernels are created without prior knowledge about the data to be approximated by means of a Genetic Programming algorithm. The quality of a kernel is evaluated independently of a particular model, using a modified version of a non parametric noise estimator. For a particular problem, performances of generated kernels are tested against common ones using weighted k-nn in the kernel space. Results show that the presented method produces specific-to-problem kernels that outperform the common ones for this particular case. Parallel programming is utilized to deal with large computational costs.

1 Introduction

Kernel methods such as Least Square Support Vector Machines (LS-SVMs) [1] have been successfully applied to regression and function approximation problems. Although they reach a good performance, obtaining very accurate models with good generalization properties, they present also some drawbacks: the selection of the kernel function could be difficult; the optimization of the parameters of the kernel function is computationally intensive, because they require the evaluation of some cross validation procedure or some Bayesian criteria with a complexity of $O(N^3)$, where N is the number of training points; and the generated models could be huge, because they include all training data inside.

In the literature, almost all kernel functions used are variants of Radial Basis Functions and the analysis of the problem is centred upon feature selection. In [2] (pp. 118-120, chapter 5) and [3] a different approach is used, based on the analysis by an expert on the specific data to extract some features to be used as a guide to create some kernel functions by combinating some well-known ones. In this paper, we focus on the creation of specific-to-problem kernels without prior knowledge.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 335–342, 2009.
© Springer-Verlag Berlin Heidelberg 2009

In [4], Koza proposed a class of evolutionary computation methodology, named Genetic Programming (GP), that are used to evolve computer programs to solve a range of problems. The programs evolved are in the form of parse trees. The use of parse trees has several advantages since it prevents syntax errors, which could lead to invalid individuals, and the hierarchy in a parse tree resolves any issues regarding function precedence.

Genetic Programming (GP) has been applied to upgrade some models [5] [6], and it has recently been used to generate automatically problem-adapted kernels for classification models without prior knowledge about the data. In [7] training errors of SVM models are optimized, avoiding overfitting by a tiebraker that evaluates the SVM margin (i.e., an approximation to its generalization capacities); in [8] the model optimized is called the Kernel-NN classification rule, and its fitness inspired on the classifier margin concept; in [9] and [10] the k-fold crossvalidation error of SVM models are optimized; in [11] the error on a validation set for SVM models is used. The problem of getting a kernel is really the problem of getting symmetric functions that follow the Mercer Theorem, a basic requirement of classical kernels methods. Getting symmetric functions is quite simple and most authors used a representation that directly produces symmetric kernels, but to deal with the Mercer condition each author applies different approaches: in [7] the condition is ignored due to computational overhead it entails and because non-Mercer kernels can give good results too; in [8] the condition is ignored because the used model is specially tolerant to non-Mercer kernels; in [9] and [10] the operators and terminal nodes used to create the kernels are chosen to always generate Mercer kernels; in [11] the created kernels can be even non symmetrical, but the conditions are evaluated after kernel creation.

The strategy used in this work differs, not only because the application to regression problems, but also because it uses the non parametric noise estimator (NNE) delta test [12] in the kernel induced space (see section 2) to evaluate the quality of each created kernel independently from the particular model in which it could be applied (SVM for regression, LSSVM, Gaussian Processes, etc.). The created kernels are a posteriori tested using the `kernelized` version of the weighted k-nearest neighbours algorithm (KWKNN), that it is less computational expensive than other options and also more robust with respect to the non-Mercer kernels.

Thus, the work presented in this paper illustrates the application of a kernel method for function approximation, specializing the kernel functions to a given problem by means of a Genetic Programming Algorithm. The rest of the paper is organized as follows: Section 2 will briefly introduce the NNE delta test, as well as the modification it is needed to use kernel-based distances instead of Euclidean; Section 3 will present the Genetic Programming Algorithm designed to adapt the kernels; in Section 4, the experiments performed on specific data are described and the results commented; and, finally, in Section 5, some conclusions will be drawn.

2 Delta Test

The estimation of the amount of noise present on a set of Input/Output data has proved to be specially useful for regression problems since it can provide an estimation of the best results that any model can reach for these specific data, and also it can be used to perform a selection of the most relevant input variables for the system [12]. The delta test for a given set of input vectors $\boldsymbol{X}_i, i = 1...n \in R^d$ with their corresponding output Y_i, can be defined as:

$$\delta_{n,k} = \frac{1}{2n}\sum_{i=1}^{n}(Y_i - Y_{N[i,k]})^2 \tag{1}$$

where $N[i,k]$ is the index of the k-th nearest neighbour to X_i in a distance measure, usually the Euclidean distance. Since $\delta_{n,1} \approx V_n$, where V_n is the variance of the noise in the output, it is possible to use $\delta_{n,1}$ as an estimation of the best mean quadratic error that could be obtained by a model designed using X, Y.

The adaptation of the algorithm to work in feature space is quite simple and is obtained by redefining the distance measure D as:

$$D\left(x, x'\right) = \sqrt{k(x,x) + k(x',x') - 2k(x,x')} \tag{2}$$

where k is a kernel function. To get the kernel-distance-based version of the delta test it is only needed to use the expression (2) instead of Euclidean distance to compute the nearest neighbours. We will refer to this as the `kernelized` delta test.

3 Genetic Programming Algorithm

The first parameter to define in a Genetic Programming algorithm is the representation of the solutions. In the case of kernels, the classical parse tree representation is adequate, leaving the main choice to the set of operators and terminal nodes that will be used. In [9] [10], the nodes are chosen to generate Mercer Kernels, using as terminal nodes known Mercer kernels and operators that guaranteed that the result will remain a Mercer Kernel (kernel closure); in [7] a resulting function $k(x,z)$ is used to create a kernel $K(x,z) = k(x,z)k(z,x)$, ensuring the symmetry of the function; in [8] the symmetry is ensured by using a set of terminal nodes in which each of them is a symmetric function of inputs; in [11] the conditions of symmetry and Mercer theorem are tested after the creation of each kernel.

In our case, as in [8], the nodes are chosen to ensure the creation of kernel functions $k(x_1, x_2; P)$ that are symmetrical with respect to the inputs x_1, x_2, and P is the set of parameters of its constituting node operators, which become the parameters of the kernel. These nodes can be seen in Table 1. Most of the operators have an arity (i.e, number of children) fixed to 1 or 2, but nodes 2 and 4 can have more children (this is a parameter *max. arity* of our algorithm that can be defined).

Table 1. Operators used in the proposed GP algorithm. *p_i is a parameter of the node, $input_i$ is the result of the evaluation of the i-th child of the node, x_1, x_2 are inputs from X, $x_1(d)$ means access to the dimension d of x_1.*

Index	Node	Index	Node
1	p_1	10	$log(input_1)$
2	$\sum_j^n input_j$	11	$exp(input_1)$
3	$input_1 - input_2$	12	$tanh(input_1)$
4	$\prod_j^n input_j$	13	$sin(input_1)$
5	$(input_1)^{input_2}$	14	$p_1(x_1(d) + x_2(d))$
6	$par_1 \cdot input_1$	15	$p_1(x_1(d) - x_2(d))^2$
7	$(input_1)^{par_1}$	16	$p_1 x_1(d) \cdot x_2(d)$
8	$-input_1$	17	$p_1 sin(\pi \cdot p_1 \cdot (x_1(d) - x_2(d)))^2$
9	$(input_1)^2$		

To limit the search space, we apply to every new tree some simplification and normalization procedures. The chosen operators for the GP algorithm are:

Initialization: the population is initialized using the *grow* procedure [13], as in [7] [11], to ensure variety of tree depths in the initial population.

Selection: the parents will be chosen by the classical roulette wheel selection procedure [14].

Cross-over: the classical GP operator *subtree exchange cross-over* is used [4]. This operator selects a random subtree of each parents and swaps them.

Mutation: *subtree prune mutation*, that substitutes a randomly selected subtree of the tree by a terminal node.

Replacement: the worst individuals of each population are replaced each generation.

A major concern of GP is the bloat effect: the growing of trees in population without fitness increase. Since the cross-over operator make the trees grow, in our algorithm the mutation operator is chosen to compensate this effect by reducing the size of the trees. Additionally, the maximum depth, number of nodes and node arity in a valid tree are parameters of the algorithm, and only valid trees are evaluated and used in reproduction.

Our GP algorithm will receive a set of input vectors $\boldsymbol{X}_i,\ i = 1...n \in \mathbb{R}^d$ with their corresponding output Y_i of a regression problem, and will compute kernels that can be used in models for its approximation. Each individual of the implemented GP algorithm is a kernel whose fitness value is computed as the square root of the `kernelized` delta test with this kernel for the training data of a regression problem divided by the variance of the output. Since all kernels have their own parameters, they are optimized with respect to the `kernelized` delta test using a Variable Neighborhood Search (VNS) [15]. The use of the VNS instead of the classical conjugate gradient is justified by the fact that the expression needed to compute the delta test is not differentiable.

The software was developed in Matlab on a personal computer and cross-compiled to a cluster platform. To access the MPI library from our codes, the MPImex toolbox is used[16][17]. Most of the computational cost is the optimization of the kernel parameters w.r.t. the `kernelized` delta test. To reduce the execution time, a load balancing scheme was implemented: a process maintains the population and executes the main loop (i.e.: generational process, individual creation, etc.); while each of the rest executes a loop waiting for receiving from the master node a kernel, applies VNS to optimize its parameters and returns the result to the master node.

4 Experiments

Once the GP algorithm used has been presented, this section will describe the particular settings for the specific experiments carried out, the resulting kernels and the evaluation of the performances of these specific-to-problem kernels versus the commonly used kernels in the literature.

4.1 Data

In our experiments we will try to model and predict the monthly river flow time series of the Snake river[1]. Note how in this case, the series presents a high level of noise, although with a recognizable periodic pattern which should be taken into account by the kernels. The data set has a total of 1092 data. The first half will be used for training and the rest for test. The indices of the time series, $t = 1 \cdots 1092$, will be used as inputs and the values y_t as outputs. The training and test data sets are defined each as half of the total data.

4.2 GP Settings

The values of the parameters of the GP algorithm in the experiments are those shown in Table 2. These settings were chosen to avoid too long execution times but without limiting too much the trees (i.e. kernels) that can be generated.

Table 2. Parameter values of the GP algorithm in the experiments

Parameter	Value	Parameter	Value
population size	60	max. tree depth	4
number of generations	100	max. tree nodes	10
subst. by generation	30	max. tree node arity	5

---

[1] `http://www-personal.buseco.monash.edu.au/hyndman/TSDL/hydrology.html`

4.3 Results

For comparison purposes, KWKNN ($k = 10$) using classical kernels is applied to the problem: the kernel Linear (3), Gaussian or RBF(4), Rational Quadratic (5) and Periodic (6).

$$k_{linear}(x_1, x_2; \{\alpha\}) = \alpha \langle x_1, x_2 \rangle \, . \tag{3}$$

$$k_{gauss}(x_1, x_2; \{\sigma\}) = \exp\left(-\frac{2}{{\sigma_2}^2}\|x_1 - x_2\|^2\right) . \tag{4}$$

$$k_{ratquad}(x_1, x_2; \{\alpha, \beta\}) = \left(1 + \frac{\|x_1 - x_2\|^2}{2\alpha\beta^2}\right)^{-\alpha} . \tag{5}$$

$$k_{periodic}(x_i, x_j; \{\sigma, \lambda\}) = \exp\left(-\frac{2}{\sigma^2}\sin\left(\frac{\pi}{\lambda}(x_i - x_j)\right)^2\right) . \tag{6}$$

Table 3. Results using common kernels with KWKNN

kernel	training error	test error	best training error	best test error
linear	7.27e-01 ± 4.87e-14	1.32e+00 ± 1.62e-15	7.27e-01	1.32e+00
gauss	7.27e-01 ± 2.10e-06	1.32e+00 ± 4.42e-04	7.27e-01	1.32e+00
ratquad	7.27e-01 ± 1.74e-06	1.32e+00 ± 2.87e-04	7.27e-01	1.32e+00
periodic	6.37e-01 ± 6.33e-02	8.91e-01 ± 5.56e-02	5.43e-01	8.45e-01

In Table 3 the results for the KWKNN models using common kernels are presented, 10 models for each kernel were generated and for each of them the kernel parameters were optimized versus the 10 fold cross validation error on half the data (the training data) by a VNS procedure. The Normalized Root Mean Square Error (nrmse, $\sqrt{mse/\sigma_y^2}$) is used, an NRMSE = 1 means a prediction as good as predicting the mean of the output. In Table 4 the results for the models using the kernels generated by one execution of the GP algorithm are shown, as well as the results of the models using the three best kernels generated by the GP algorithm (according to their fitness). It is worth to recall that the values of the kernel parameters of the three best kernels generated by the GP algorithm are those obtained optimizing the `kernelized` delta test, i.e. the parameters of the kernels were not specifically optimized in these cases for the KWKNN models, but they get good results. The results of 10 KWKNN models using the three best kernels optimizing the kernels parameters as in common kernel case are also shown.

The results show that the performances reached by the specifically-generated kernels are better than those obtained by the usual and more general kernels. The expressions of the three best kernels 4 reveal that the main feature of the data is its strong periodicity, confirmed both by the fact that from common kernels the one with best results is the called Periodic Kernel, Equation (6).

Table 4. Results of KWKNN models generated from the final population of GP

fitness	training error	test error
4.71e-01 ± 6.12e-02	6.57e-01 ± 1.84e-01	9.10e-01 ± 1.82e-01

Kernels of the 3 best individuals
$K(u,v;\{p_1,p_2,p_3,p_4\}) = \left(p_1(u_1-v_1)^2\right)\cdot\left(-\left((p_2)-\left(p_3 sin(\pi p_4(u_1-v_1))^2\right)\right)\right)$
$K(u,v;\{p_1,p_2,p_3\}) = (p_1)\cdot\left(p_2 sin(\pi p_3(u_1-v_1))^2\right)$
$K(u,v;\{p_1,p_2,p_3,p_4\}) = \left(-\left((p_2)-\left(p_3 sin(\pi p_4(u_1-v_1))^2\right)\right)\right)^{p_1}$

Models using the 3 best kernels and values of kernel parameters from GP		
fitness	training error	test error
3.26e-01	4.98e-01	7.82e-01
3.86e-01	5.57e-01	8.27e-01
3.89e-01	4.57e-01	8.30e-01

Models using the 3 best kernels			
training error	test error	best training error	best test error
7.56e-1 ± 1.32e-1	1.028e-1 ± 1.18e-1	4.98e-1	7.82e-1
7.12e-1 ± 1.47e-1	9.470e-1 ± 1.38e-1	4.36e-1	5.98e-1
6.62e-1 ± 1.33e-1	8.683e-1 ± 1.45e-1	4.20e-1	5.86e-1

5 Conclusions

The use of specific-to-problem kernels are strongly recommended in the literature of kernel methods, but it requires a deep understanding of the data and the kernels themselves to be effective. This paper has presented a Genetic Programming Algorithm capable of designing specific-to-problem kernels that use a `kernelized` version of the non parametric noise estimator delta test. The kernels obtained for a time series data show that they were able to model the periodicities of the series, which is its main feature and the key to make a successful prediction. Another element to remark is the utility of parallel programming as an almost obligated use in order to obtain practical execution times.

Acknowledgment

This work was carried out using the facilities of the Edinburgh Parallel Computing Centre[2] (EPCC) and under the HPC-EUROPA project (RII3-CT-2003-506079), with the support of the European Community - Research Infrastructure Action under the FP6 "Structuring the European Research Area" Programme. This work has been partially supported by the Spanish CICYT Project TIN2007-60587 and Junta Andalucia Project P07-TIC-02768.

---

[2] `http://www.epcc.ed.ac.uk/`

References

1. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific Publishing, Singapore (2002)
2. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2005)
3. Rubio, G., Guillen, A., Herrera, L.J., Pomares, H., Rojas, I.: Use of specific-to-problem kernel functions for time series modeling. In: ESTSP 2008: Proceedings of the European Symposium on Time Series Prediction, pp. 177–186 (2008)
4. Koza, J.: Genetic Programming. MIT Press, Cambridge (1992)
5. Rubio, G., Pomares, H., Rojas, I., Guillén, A.: A basic approach to reduce the complexity of a self-generated fuzzy rule-table for function approximation by use of symbolic regression in 1d and 2d cases. In: IWINAC (2), pp. 143–152 (2005)
6. Rubio, G., Pomares, H.: A basic approach to reduce the complexity of a self-generated fuzzy rule-table for function approximation by use of symbolic interpolation. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 34–41. Springer, Heidelberg (2005)
7. Howley, T., Madden, M.: The genetic kernel support vector machine: Description and evaluation. Artificial Intelligence Review 24, 379–395 (2005)
8. Gagné, C., Schoenauer, M., Sebag, M., Tomassini, M.: Genetic programming for kernel-based learning with co-evolving subsets selection. DANS PPSN 06, 1008 (2006)
9. Methasate, I., Theeramunkong, T.: Kernel Trees for Support Vector Machines. IEICE Trans. Inf. Syst. E90-D(10), 1550–1556 (2007)
10. Sullivan, K.M., Luke, S.: Evolving kernels for support vector machine classification. In: GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 1702–1707. ACM Press, New York (2007)
11. Diosan, L., Rogozan, A., Pecuchet, J.P.: Evolving kernel functions for svms by genetic programming. In: ICMLA 2007: Proceedings of the Sixth International Conference on Machine Learning and Applications, Washington, DC, USA, pp. 19–24. IEEE Computer Society Press, Los Alamitos (2007)
12. Lendasse, A., Corona, F., Hao, J., Reyhani, N., Verleysenp, M.: Determination of the mahalanobis matrix using nonparametric noise estimations. In: ESANN, pp. 227–232 (2006)
13. Koza, J.R.: A genetic approach to the truck backer upper problem and the intertwined spiral problem. In: Proceedings of IJCNN International Joint Conference on Neural Networks, vol. IV, pp. 310–318. IEEE Press, Los Alamitos (1992)
14. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning . Addison-Wesley, New York (1989)
15. Mladenović, N., Hansen, P.: Variable neighborhood search. Comps. in Opns. Res. 24, 1097–1100 (1997)
16. Guillén, A.: MATLAB Interface for Sun MPI functions, `http://atc.ugr.es/~aguillen/MPI/MPI_MATLAB.htm`
17. Guillen, A., Rojas, I., Rubio, G., Pomares, H., Herrera, L.J., Gonzalez, J.: A new interface for mpi in matlab and its application over a genetic algorithm. In: ESTSP 2008: Proceedings of the European Symposium on Time Series Prediction, pp. 37–46 (2008)

Combining Genetic Algorithms and Mutation Testing to Generate Test Sequences*

Carlos Molinero, Manuel Núñez, and César Andrés

Dept. Sistemas Informáticos y Computación
Universidad Complutense de Madrid, 28040 Madrid, Spain
molinero@fdi.ucm.es, mn@sip.ucm.es, c.andres@fdi.ucm.es

Abstract. The goal of this paper is to provide a method to generate efficient and short test suites for Finite State Machines (FSMs) by means of combining Genetic Algorithms (GAs) techniques and mutation testing. In our framework, mutation testing is used in various ways. First, we use it to produce (faulty) systems for the GAs to learn. Second, it is used to sort the intermediate tests with respect to the number of mutants killed. Finally, it is used to measure the fitness of our tests, therefore allowing to reduce redundancy. We present an experiment to show how our approach outperforms other approaches.

1 Introduction

Software testing is an expensive and time consuming task. If a formal approach is used, tests are derived from a specification. For derived test sets to be complete, the tester needs to assume a given set of assumptions and hypotheses, allowing to reduce the space of the possible implementations. Our intention is to combine GAs and mutation testing to create a new approach capable of deriving a test suite that, with a short amount of execution time, finds the 95% of the faulty implementations. We call our methodology GAMuT (Genetic Algorithm and MUtation Testing) and it is composed of 3 main phases: Learning through evolution, learning through specialization, selection and reduction of the test cases.

GAs have shown to have a good performance in search and optimization problems. There exists a number of papers where GAs are used in testing (e.g.[4,5,2]). They usually represent the test data generation problem as an optimization problem and heuristics are used to generate test cases. Mutation Testing has been widely used for checking the performance in test suites, by measuring its capability to kill mutants, for details into applications developed through *mutation testing* see [3,8] for some formal approaches and [1] for a critical discussion on its use. We propose to use mutation as a way to provide the GA with enough learning examples in an automated way, by modifying the specification and subsequently creating simulations of faulty implementations. We compile several populations into a *community*. Therefore, each community has several *populations*, with several *inhabitants* each, and each individual has a DNA that directly

---

[*] Research supported by the Spanish projects WEST (TIN2006-15578-C02-01) and MATES (CCG08-UCM/TIC-4124).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 343–350, 2009.
© Springer-Verlag Berlin Heidelberg 2009

represents a test sequence, that is, a chain of inputs that will be applied to the implementation under test (in short, IUT). In our methodology, mutation testing is used in three ways. In the first phase of GAMuT, called *learning through evolution*, the community is presented with a set of mutants. In our approach we use 100 IUTs and each of the populations is confronted with all the IUTs. The fitness function is a heuristic based on the percentage of mutants killed. The second phase of GAMuT, *learning through specialization*, also uses mutants but in a different way. In this case, each population inside the community (after having evolved to kill the biggest number of mutants) is given one IUT, mutated from the specification, and each population evolves to try and minimize the length of the sequence needed to find the error inside that mutant. Finally, we select the fittest individual from each population (from both phases) and create a set of tests, which we confront with the final and biggest set of mutants (500 in our methodology), to be able to find the number of mutants killed by each of the selected individuals. Then, as the last step, we make a subdivision with the smaller set of tests that kills the highest number of mutants.

The rest of the paper is organized as follows, in Section 2 we present the language used to define specifications, the mutation operators, and define the operations and objects of GA. In Section 3 we give a schema of the different phases of our methodology. In Section 4 we present experiments and the results of the comparisons of our technique with random testing. Finally, in section 5 we give our conclusions and some ideas for further development.

2 Preliminaries

We introduce hereby some general notions, about Finite State Machines, and about our GA. Specifications and its mutants are given by means of finite state machines.

Definition 1. A *Finite State Machine*, in short FSM, is a tuple $M = (S, \mathcal{I}, \mathcal{O}, Tr, s_0)$ where S is a finite set of states, $\mathcal{I}$ is the set of input actions, $\mathcal{O}$ is the set of output actions, Tr is the set of transitions, and s_0 is the initial state. We say that M is *input-enabled* if for all state $s \in S$ and input $i \in \mathcal{I}$, there exist $s' \in S$ and $o \in \mathcal{O}$ such that $(s, s', i, o) \in Tr$. We say that M is *deterministic* if for all $s \in S$ and $i \in I$ there do not exist two different transitions $(s, s_1, i, o_1), (s, s_2, i, o_2) \in Tr$. □

A transition belonging to Tr is a tuple (s, s', i, o) where $s, s' \in S$ are the initial and final states of the transition, and $i \in \mathcal{I}$ and $o \in \mathcal{O}$ are the input and output actions, respectively. Intuitively, a transition (s, s', i, o) of a FSM indicates that if the machine is in state s and receives the input i, then the machine emits the output o and moves to s'. Along the rest of this paper we assume that all the machines are input-enabled and deterministic. The first restriction, as usual, is needed to ensure that implementations will react to any input provided to them. The second limitation is not vital to our approach, although it simplifies the implementation of the algorithm.

Next we define two mutation operators: One will modify the output of a certain transition and the other will modify the state at which the transition arrives.

Definition 2. Let $M = (S, I, O, Tr, s_0)$ be a FSM. The application of the operator $moper_1$ to M produces a mutant by choosing a transition $tr = (s, s', i, o) \in TR$ and an output $o' \neq o$, and by returning the FSM $M' = (S, I, O, Tr', s_0)$ such that $Tr' = Tr \backslash \{tr\} \cup \{tr'\}$, where $tr' = (s, s', i, o')$. The application of the operator $moper_2$ to M produces a mutant by choosing a transition $tr = (s, s', i, o) \in TR$ and a state $s'' \neq s$, and by returning the FSM $M' = (S, I, O, Tr', s_0)$ such that $Tr' = Tr \backslash \{tr\} \cup \{tr'\}$, where $tr' = (s, s'', i, o)$. □

A usual problem in mutation testing is that the application of mutation operators may produce an *equivalent mutant*, that is, a mutant equivalent to the specification. Therefore, when applying mutation testing it is normal that tests do not kill all the mutants since some of them (the equivalent ones) are not even supposed to be killed.

GAs are an AI technique that uses metaphors of mechanisms present in nature for organism to develop, adapt and reproduce, to try to survive in the system in which they act and live. Its main operators are mutation, reproduction (genetic crossing) and selection of the fittest individuals. GAs are a good method to use with black-box testing since if we do not have any information regarding the internal structure of the IUT, then the testing problem can be expressed as a search problem, guided by a heuristic. In our case, the search space is the set of all possible implementations that may have mutated from the given specification. GAs can adapt themselves to find this optimum, as long as the fitness function is correctly defined. Let us note that if we were to leave a genetic algorithm running indefinitely, the whole population will converge to the same inhabitant, that would be the one that maximizes the fitness function.

3 Description of GAMuT

Next we describe our GA. Since we do not want to find a single solution and for the sake of genetic diversity, we have added another component to usual GAs: A *community* holds several *populations*, having in turn several inhabitants. We do not allow genetic crossing between populations. This can be seen as a parallelism to what is sometimes called in the literature *species*. An inhabitant of a specific population has a DNA sequence that is formed by genes, that codifies the test to be applied. The value that each gene can take is any input $i \in I$. The DNA sequence can be modified through mutation and recombination, that is, by mating of two inhabitants. In addition, it can also mutate the length of its sequence. The community holds the specification that we are trying to check as well as a set of examples of mutated specifications that we will call $\mathrm{ex}_{\mathrm{IUT}}$, and each population holds a mutated IUT specific for it, used in the specialization phase.

In order to initiate our algorithm, all genes are randomly initialized, so that a random number of gens is available for each possible DNA. The size of the population, the number of populations, and the fitness function are decisions to be made by the tester, that has to take into account the number of states of the machine. An example of fitness function used in GAMuT is shown in Section 4. The search finished by two means, one will be to reach the maximum number of generations specified for the GA, the other is that the test sequences are able to detect a specific percentage of the faulty IUTs. We count with two mutation operators for DNA: To change the value of one or more gens and to modify the length of the DNA sequence. DNAs can be modified also by combining them with the DNA of another inhabitant through mating. In our approach we have used a single point crossover, with random position for the division.

As usual, we will use *elitism* as a way to control that only the best solutions actually get to mate and reproduce. We will use the *roulette wheel* selection technique that allows to choose the inhabitants in a proportional scale to its fitness. In our case, we also add after reproduction the original set of elected best inhabitants from the population, not to loose solutions. In order to decide who are the best, we find the top/lowest fitness scores, and we take a percentage of the top to be chosen as the best.

Our algorithm is divided into several phases that are graphically represented in Figure 1. These phases are:

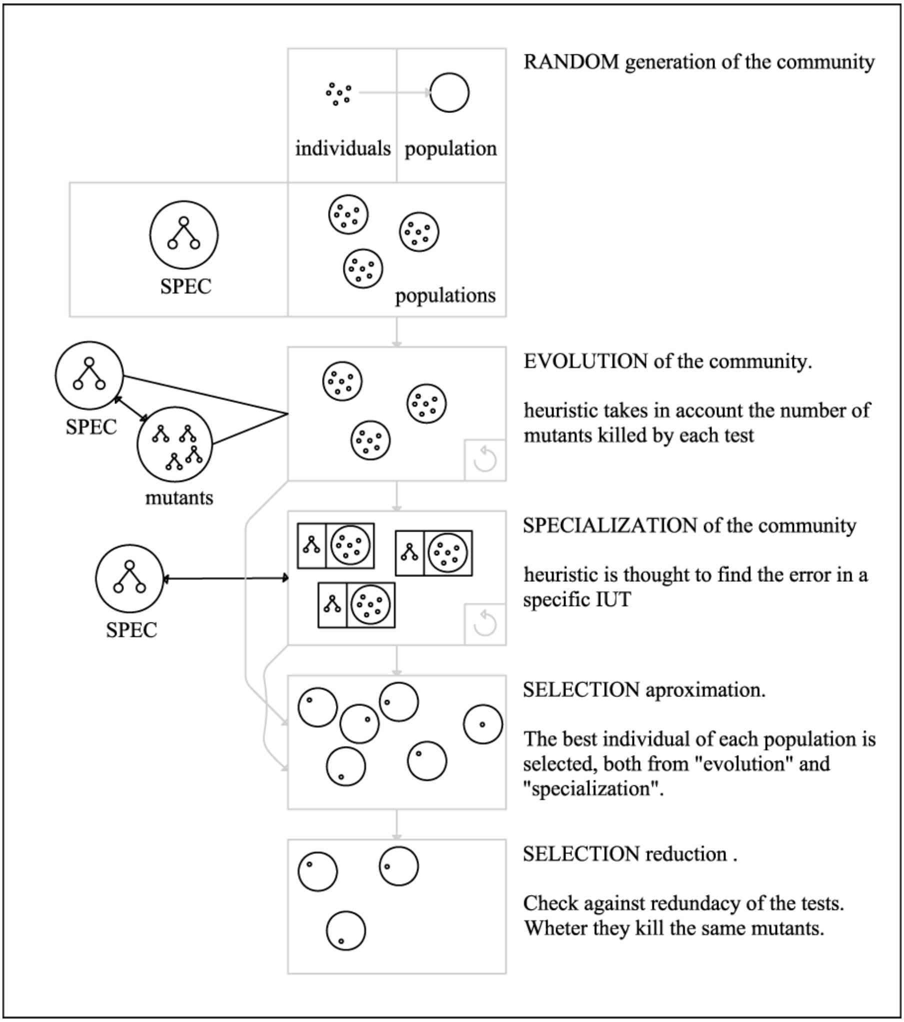

RANDOM generation of the community
individuals
population
SPEC
populations
EVOLUTION of the community.
heuristic takes in account the number of mutants killed by each test
SPEC
mutants
SPECIALIZATION of the community
heuristic is thought to find the error in a specific IUT
SPEC
SELECTION aproximation.
The best individual of each population is selected, both from "evolution" and "specialization".
SELECTION reduction .
Check against redundacy of the tests. Wheter they kill the same mutants.

Fig. 1. Scheme of our approach

1. *Generation*. First, the community is generated. A specified number of populations are stored into the community. Second, inhabitants for each population are introduced into the different populations. Each inhabitant holds in its DNA a sequence of inputs that could be applied as a test. Third, the specification is introduced and a set of mutants is generated by applying the two mutation operators.
2. *Evolution*. In the evolution phase each inhabitant tries to kill mutants from a reduced set of candidates. The fitness function is as simple as the percentage of mutants killed from the set. These populations evolve as every other GA during a number of generations, to try and find a solution. The best individuals of each population are selected and saved for further use.
3. *Specialization*. In this phase the already evolved population is modified by inserting one specific faulty IUT into each of the populations, and making the GA try to find suitable individuals that will find the mistake present in the mutation. In this case the fitness function is thought to find one mistake in the IUT with the shortest DNA sequence. The individuals with the shortest test sequence while detecting the fault in the IUt of each population are selected and saved for further use.
4. *Selection*. We sort the inhabitants generated after the previous phases to get the ones that kill more mutants (trying to shorten the overall testing sequence length). More precisely, we initially select the ones that are needed to kill the mutants that did not die when applying the first test, and we repeat the process. Thus, by eliminating redundant tests we have to find a tradeoff between killing the highest number of mutants and getting the shortest sequence. In our framework, we eliminate every test that kills less than 1% of the mutants.

4 Application of GAMuT: Results and Comparisons

We have developed a tool to implement our methodology. We have used Java Technology (JDK 1.6) and the Netbeans software, and made usage of the MVC architecture to enable ease of maintenance, and use session facade, singleton and light-weight design patterns. FSMs are implemented as a set of adjacency matrices, one for each input; instead of having 1 to denote that a connection exists, the corresponding output is specified. The singleton community class holds the objects that must be available at all times. There is a class for each GA object, 3 heuristics that implement a common interface and a couple of auxiliary classes, like matrix and errors.

We have worked with a specification having 50 states. The corresponding FSM has been randomly generated to avoid that its specific structure produces any bias in the application of our methodology. Our set of inputs is $I = \{a, b\}$ and the set of outputs is $O = \{1, 2\}$, but bigger sets could be considered without affecting the results. From the specification we derive 500 mutants, from which 100 are chosen as the learning set. The used specification in the experiment reported in this paper is shown in Figure 2. The number of populations in the community is 30.The size of each population is of 50 inhabitants. The maximal number of generations allowed for the phase of evolution is 200. The maximal number of generations for specialization is set to 200.

DNA are sequences $< i_1\, i_j\, \ldots\, i_n >$ of inputs. We mutate the length of the DNA sequence in a 33% of the cases, randomly deciding whether to enlarge or to shorten it.

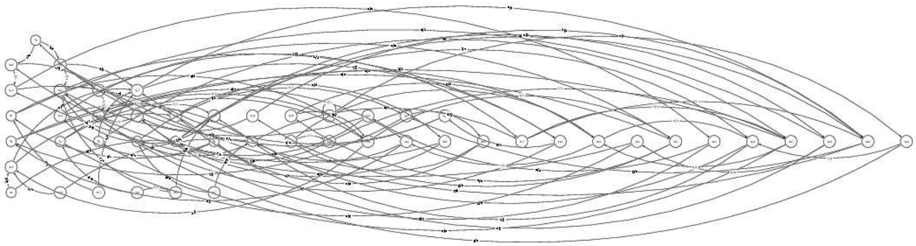


Fig. 2. Example of specification

We mutate the value of various genes simultaneously, being the number of mutations randomly chosen, but proportionally to the inverse of its fitness function, that is, $n = \frac{random[1,n_g/3]}{f}$, being n the number of genes modified, n_g the total number of genes, and f the value given by the fitness function.

There are two fitness functions: One for the *evolution* phase and another one for the *specialization* phase.

1. Fitness for evolution: $f = \frac{\mid mutantskilled \mid \cdot 100}{\mid mutants \mid}$ is the percentage of mutant implementations, for which an inhabitant finds an error.
2. Fitness for specialization: $f = e - l \cdot \alpha$, where e is equal to 1 if an error is found, and l is the length of the genetic sequence starting to count from the point where the error is found; otherwise, starting to count from the beginning of the sequence, and $0 \leq \alpha \leq 1$ is its weight. This heuristic tries to approximately find the shortest test sequence that detects an error in the IUT that we are checking.

Mating is done through selection with the *roulette wheel* selection technique and it uses a single point *crossover* for DNA reconversion.

In order to check convergence in the *evolution* step, we take into account the fittest individuals from each population, and consider them as a test suite, that is applied to the number of learning IUTs. If the total number of killed mutants is over 85%, then we allow the program to continue towards its next step. For the *specialization* phase we set finding 75% of the errors as a good number to stop the algorithm and go to the selection process. This number does not represent the mutants that the set will kill, but how many of the test found an error in a specific IUT. Obviously, these values can be modified to find fitter tests, but our experience shows that these limits behave good enough. If convergence is not reached before the total number of generations allowed, then the system stops the process and continue with its next step.

In order to compare our methodology we have implemented a random testing tool. Random testing is a technique argued to be as valid as any other testing technique (see [7,9,6]). As we can see in Figure 3, the test suite resulting after selection is the shortest one and detects up to 94.6% of mutants as faulty. Compared to the randomly generated test suite, it outperforms it both by number of mutants killed and by having a shorter test sequence. Even though to produce our test suite takes more time than a random generation, the extra performance of our approach to find errors, even though not very

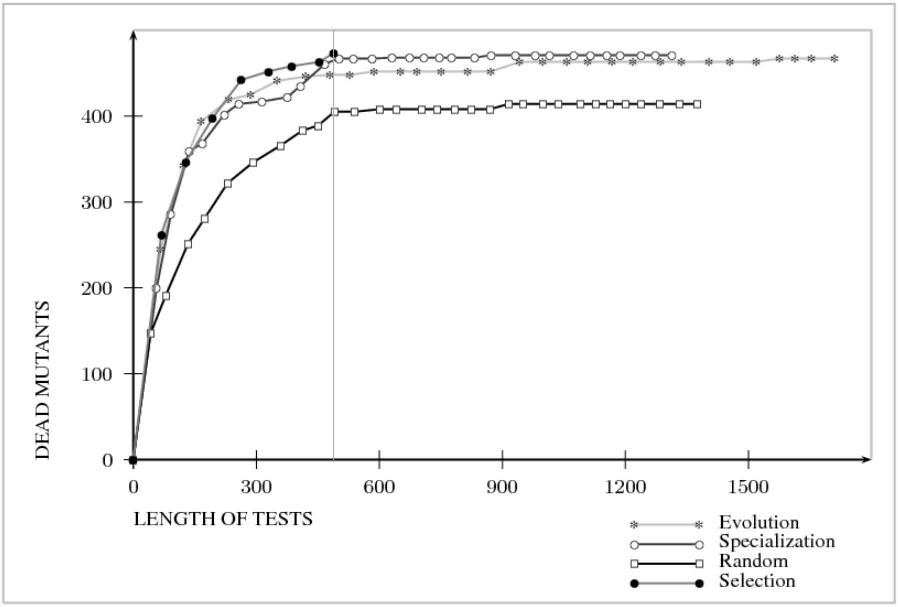


Fig. 3. Killed mutants/length of test sequence

significant, is worth the additional computations. Actually, the time spent in testing is a valuable asset. Thus, we have to minimize the number of applied tests (in other words, the length of the test suite) even if we have to make additional *off-line* computations, that are very cheap when compared to the cost of testing. Actually, in order to test a system we usually have to stop it. Thus, the more time the system is halted, the most expensive the testing process is, since the system cannot be producing what it is supposed to produce. In the case of the evolution and specialization test suites, they are between 1% and 2% below in test coverage (i.e. in proportion of killed mutants) because we have chosen the best tests of both test suites, and they are a lot longer to apply due to their high redundancy on the kind of errors found. This is even so if we eliminate, from the selected tests, those that kill under 1% of the mutants, because the time to apply those tests, compared to the benefits of the number of mutants detected, makes it not worthy to retain them in the final test suite.

5 Conclusions and Future Work

The results after experimentation with the implementation of GAMuT have led us to claim that the existence of these different phases is crucial. GAMuT combines the *evolution* as a massive learning process (thanks to mutation testing techniques), in which we try to reach a maximum, and a more local search, *specialization*, that creates elements in a more sophisticated way by creating a variety of viewpoints, seen as genetic diversity, that confronts the same problem. It is also important to remark the order in which the phases are applied, because this modifies the starting point of the population for the second GA step (*specialization*), making it a good starting point to be able to locate an optimum. Furthermore, *selection* is crucial as well because we need to get rid of the redundancy that the overlapping solutions offers, a normal consequence of the stochastic transformations.

We believe that this approach is feasible and since it tends to create a close-to-complete set of tests, with a short amount of testing time, it is suitable for its application on a number of systems, once it is developed for other languages. Due to this fact, one line of future work will be to study how to codify more languages into the DNA sequences and any derivation that will cause this on the rest of the elements of the system, like the mutation operators and the fitness functions.

Another idea for future development is to provide the GA algorithm with memory for its inhabitants. Then, they could foresee the best mutation rates and the ones that are leading them to a better heuristic value, maybe through the use of a neural network or simply by probabilistic methods. In this way we can reach convergence faster.

Acknowledgements. We would like to thank the anonymous reviewers of this paper for the careful reading and their suggestions.

References

1. Andrews, J.H., Briand, L.C., Labiche, Y.: Is mutation an appropriate tool for testing experiments. In: 27th Int. Conf. on Software Engineering, ICSE 2005, pp. 402–411. ACM Press, New York (2005)
2. Derderian, K., Hierons, R.M., Harman, M., Guo, Q.: Automated Unique Input Output sequence generation for conformance testing of FSMs. Computer Journal 49(3), 331–344 (2006)
3. Fabbri, S.C.P.F., Delamaro, M.E., Maldonado, J.C., Masiero, P.C.: Mutation analysis testing for finite state machines. In: 5th IEEE Int. Symposium on Software Reliability Engineering, ISSRE 1994, pp. 220–229. IEEE Computer Society Press, Los Alamitos (1994)
4. Fatiregun, D., Harman, M., Hierons, R.M.: Evolving transformation sequences using genetic algorithms. In: 4th IEEE Int. Workshop on Source Code Analysis and Manipulation, SCAM 2004, pp. 65–74. IEEE Computer Society Press, Los Alamitos (2004)
5. Guo, Q., Hierons, R.M., Harman, M., Derderian, K.: Computing unique input/ouput sequences using genetic algorithms. In: Petrenko, A., Ulrich, A. (eds.) FATES 2003. LNCS, vol. 2931, pp. 169–184. Springer, Heidelberg (2004)
6. Hamlet, D.: When only random testing will do. In: 1st International Workshop on Random Testing, pp. 1–9. ACM Press, New York (2006)
7. Hamlet, R.: Random testing. In: Marciniak, J. (ed.) Encyclopedia of Software Engineering, pp. 970–978. Wiley, Chichester (1994)
8. Sugeta, T., Maldonado, J.C., Wong, W.E.: Mutation testing applied to validate SDL specifications. In: Groz, R., Hierons, R.M. (eds.) TestCom 2004. LNCS, vol. 2978, pp. 193–208. Springer, Heidelberg (2004)
9. Zhu, H., Hall, P.A.V., May, J.H.R.: Software unit test coverage and adequacy. ACM Computing Surverys 29(4), 366–427 (1997)

Testing Restorable Systems by Using RFD[⋆]

Pablo Rabanal and Ismael Rodríguez

Dept. Sistemas Informáticos y Programación
Universidad Complutense de Madrid, 28040 Madrid, Spain
prabanal@fdi.ucm.es, isrodrig@sip.ucm.es

Abstract. Given a finite state machine denoting the *specification* of a system, finding some short interaction sequences capable to reach some/all states or transitions of this machine is a typical goal in testing methods. We study the problem of finding such sequences in the case where configurations previously traversed can be *saved* and *restored* (at some cost). Finding optimal sequences for this case is an NP-hard problem. We propose an heuristic method to approximately solve this problem based on an evolutionary computation approach, in particular *River Formation Dynamics*. Some experimental results are reported.

1 Introduction

Testing constructed systems is an industrial necessity in any IT developing process. In particular, *formal testing techniques* [10,12,3,17] allow to perform testing tasks in a systematic and (semi-)automatic way. Typically, testers refuse to pursue the *completeness*, that is, they do not aim at applying some tests such that observed results allow to *precisely* determine whether the *implementation under test* (IUT) is correct or not. Instead, some *coverage testing* [19,11,8] criteria are pursued. For instance, given a specification represented by a *finite state machine* (FSM), we could be interested in applying a test suite that allows to reach, in the IUT, all transitions or all nodes defined in the specification (assuming that the IUT were actually defined as the specification). Reaching a specification configuration by following a given path does not imply that the same configuration is reached in the IUT. Therefore, a test suite that allows to reach each specification state or transition by following a single path is *not* a complete test suite in general. However, such an incomplete test suite could detect a high number of possible faults. Thus, incomplete test suites can be very useful in practice.

Several graph theory techniques have been applied to find short test sequences allowing to reach, at least once, all states or transitions in an FSM specification [1,4,18]. In this paper we generalize testing methods allowing to reach some/all FSM states or transitions to the case where the tester can *restore* any previous configuration of the system. Let us assume that the IUT is a software

---

[⋆] Research partially supported by projects TIN2006-15578-C02-01, PAC06-0008-6995, and CCG08-UCM/TIC-4124.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 351–358, 2009.
© Springer-Verlag Berlin Heidelberg 2009

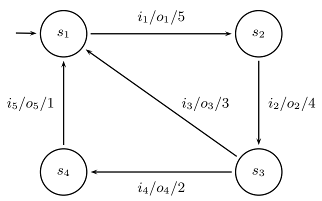


Fig. 1. Example of cost-weigthed FSM

system and the tester can *save* the complete current configuration of the system at any time. Then, at any subsequent time, the tester could *restore* such configuration and execute the system from that configuration on. In particular, after restoring the configuration, he could follow a different path to the one he followed the previous time. Notice that, if configurations can be saved/restored, then the tester can use this feature to avoid *repeating* some execution sequences. Thus, some time assigned to testing activities can be saved. Let us also note that saving/restoring the complete configuration of a system could be time-expensive. In particular, these costs are similar to the costs of executing *process suspending* and *process restoring* operations in any operating system, which in turn are due to copying the program state from/to RAM to/from hard disk, respectively. Thus, a testing technique using these features should take these costs into account. For instance, let us consider the (cost-weighted) FSM depicted in Figure 1. The initial state is s_1 and transitions are labelled by an input, an output, and the time required to take the transition. Let us suppose transitions $s_3 \xrightarrow{i_3/o_3/3} s_1$ and $s_4 \xrightarrow{i_5/o_5/1} s_1$ are *critical*, i.e. they must be tested. The interaction plan $s_1 \xrightarrow{i_1} s_2 \xrightarrow{i_2} s_3 \xrightarrow{i_3} s_1 \xrightarrow{i_1} s_2 \xrightarrow{i_2} s_3 \xrightarrow{i_4} s_4 \xrightarrow{i_5} s_1$ covers these transitions at cost $5+4+3+5+4+2+1 = 24$ time units. Let us suppose that saving/loading a previously traversed configuration costs 7 time units. Then, the interaction $s_1 \xrightarrow{i_1} s_2 \xrightarrow{i_2} s_3 \xrightarrow{i_3} s_1 \xrightarrow{load\ \ s_3} s_3 \xrightarrow{i_4} s_4 \xrightarrow{i_5} s_1$ costs $5+4+3+7+2+1 = 22 < 24$, so loading is preferable in this case. In general, saving/restoring operations should be used only when the cost of repeating a path is bigger than the cost of saving and restoring a configuration. Let us note that this is actually the case of many software systems that are computation-intensive but that do not use much memory (e.g. control systems, embedded systems, etc).

In this paper we present a method that, given (i) the cost of saving/restoring a configuration; (ii) an FSM specification explicitly denoting the cost of performing each transition; and (iii) a set of *critical* system configurations that are required to be observed, it returns a plan to interact with the system (possibly including saving/restoring operations) that allows to reach all critical configurations in a low overall time. The problem of finding the *optimal* interaction sequence fulfilling these conditions will be called the *Minimum Load Sequence* problem (`MLS`). We can observe that `MLS` is an NP-complete problem[1]. Thus, we will provide a heuristic method to approximate the optimal path. Let us note that an interaction plan where saving/restoring configurations is allowed (under a certain cost) is, in fact, a plan where each save/restore point represents a *bifurcation*: Initially, we will take a path, and then we will return to a previous point and follow another path. Thus, interaction plans found by our method will not be represented by a *sequential* path, but by a *tree* covering all critical points where the root is the initial state of the FSM. Hence, our goal will be finding a tree where the sum of the costs of all the transitions of the tree (including the sum of the costs of saving/restoring operations) is minimal[2].

In order to heuristically solve this NP-hard problem, we use an *evolutionary computation* approach [7,9]. In particular, we consider *River Formation Dynamics* (RFD) [13,14,15]. Briefly, RFD consists in simulating how drops transform the landscape while they flow down to the sea. Drops erode the ground when they traverse high decreasing gradients, and they deposit carried sediments in flatter areas. By increasing/decreasing the altitude of nodes, gradients are modified, which in turn affects movements of subsequent drops. Eventually, formed decreasing gradients will depict *paths* from the point/s where it rains to the sea. Since drop movements are driven by the *difference* of some values (in particular, node altitudes) rather than on these values themselves, RFD can be seen as a *gradient-oriented* version of *Ant Colony Optimization* (ACO) [6,5,2]. However, there are other important differences between RFD and ACO (see [14,15] for details).

The rest of the paper is structured as follows. In Section 2 we introduce our general heuristic method, while in Section 3 we show how to apply it two our particular problem. Afterwards, in Section 4 we report the results obtained in some experiments. Finally, in Section 5 we present our conclusions and some lines of future work.

2 Brief Introduction to River Formation Dynamics

In this section we briefly introduce the basic structure River Formation Dynamics (RFD) (for further details, see [13,15]). Given a working graph, we associate *altitude* values to nodes. *Drops* erode the ground (they reduce the altitude of nodes) or deposit the sediment (increase it) as they move. The probability of the drop to take a given edge instead of others is proportional to the gradient of the down slope in the edge, which in turn depends on the difference of altitudes between both nodes and the distance (i.e. the *cost* of the edge). At the beginning, a flat environment is provided, that is, all nodes have the same altitude. The exception is the destination node, which is a hole (the *sea*). Drops are unleashed

---

[1] The proof of its NP-completeness is given in the extended version of this paper [16].

[2] A formal presentation of `MLS`, in particular proving that focusing on trees rather than on sequences is a right approach, can also be found in [16].

at the origin node/s, and they spread around the flat environment until some of them fall in the destination node. This erodes adjacent nodes, which creates new down slopes, and in this way the erosion process is propagated. New drops are inserted in the origin node to transform paths and reinforce the erosion of promising paths. After some steps, good paths from the origin to the destination are found. These paths are given in the form of sequences of decreasing edges from the origin to the destination. Several improvements are applied to this basic general scheme (see [13,15]).

Compared to a related well-known evolutionary computation method, *Ant Colony Optimization*, RFD provides some advantages. On the one hand, local cycles are not created and reinforced because they would imply an *ever decreasing cycle*, which is contradictory. Though ants usually take into account their past path to avoid repeating nodes, they cannot avoid to be led by pheromone trails through some edges in such a way that a node must be repeated in the next step[3]. However, *altitudes* cannot lead drops to these situations. Moreover, since drops do not have to worry about following cycles, in general drops do not need to be endowed with *memory* of previous movements, which releases some computational memory and reduces some execution time. On the other hand, when a shorter path is found in RFD, the subsequent reinforcement of the path is fast: Since the same origin and destination are concerned in both the old and the new path, the difference of altitude is the same but the distance is different. Hence, the edges of the shorter path necessarily have higher down slopes and are immediately preferred (in average) by subsequent drops. Finally, the erosion process provides a method to avoid inefficient solutions because sediments tend to be cumulated in blind alleys (in our case, in *valleys*). These nodes are filled until eventually their altitude matches adjacent nodes, i.e., the valley disappears. This differs from typical methods to reduce pheromone trails in ACO: Usually, the trails of *all* edges are periodically reduced at the same rate. On the contrary, RFD intrinsically provides a *focused* punishment of bad paths where, in particular, those nodes blocking alternative paths are modified.

3 Applying RFD to Solve `MLS`

In this section we show how the general RFD scheme, briefly introduced in the previous section, is adapted to solve `MLS`. Due to the lack of space, some details are omitted here (for further details and more complete explanations, see [16]). First, let us note that our goal will be constructing load trees where the root is the *initial state* of the given `WFSM`. In terms of RFD, we will make this node be the final goal of all drops, i.e. the *sea*. In order to make drops go in this direction, each transition of the `WFSM` will be represented in the working graph of RFD by an edge leading to the *opposite* direction. Thus, final trees constructed by RFD will have to be inverted in order to constitute valid `MLS` solutions. During the execution of RFD, new drops will be introduced in nodes representing critical

---

[3] Usually, this implies either to repeat a node or to *kill* the ant. In both cases, the last movements of the ant were useless.

configurations (i.e. it *rains* in these nodes). After executing RFD for some time, a solution tree will be constructed by taking, for each critical node, the path leading to the sea through the *highest* available decreasing gradient.

Let us note that load trees may include repeated states. In particular, a repeated state denotes that we return to a previously traversed state by taking transitions instead of by loading (a load is simply represented by a bifurcation). Thus, solutions constructed by RFD must be able to include repetitions as well. In order to allow this, *additional edges* connecting each state with the rest of states through the *shortest available path* will be added to the graph. Let us suppose that we wish to go from A to an (unvisited) critical node B, and the cheapest way to do it is traversing some (repeated and/or not critical) nodes and transitions $N_1 \rightarrow \ldots \rightarrow N_m$ and next taking a transition from node N_m to B. Rather than doing this, we will take a single *direct* transition from A to B whose cost will be the addition of costs of transitions $A \rightarrow N_1 \rightarrow \ldots \rightarrow N_m \rightarrow B$. Technically, no state or transition will be repeated in the working graph of RFD by taking this direct edge.

In order to compute the cost of these additional edges, before launching RFD we will execute the *Floyd* algorithm for the graph representing our `WFSM`. Given a graph, the Floyd algorithm finds the shortest paths connecting each pair of nodes.

The second main modification of the general RFD scheme concerns load costs. The general RFD scheme does not include any negative incentive to form bifurcation points, i.e. points where two or more flows join together. However, a negative incentive should be included because these points will represent *loads* in our solutions, which imply some additional cost. Negative incentives should be proportional to the load cost, in such a way that loads are preferred only if repeating nodes is more expensive than loading. We consider two (negative) incentive approaches. On the one hand, let us suppose that a drop can take an edge connecting its current node to a node N where some drop has already moved this turn. Note that moving to N would imply that the hypothetical solution formed by both drops would include bifurcation (i.e. a load) at N. In order to take load costs into account, we will bias the *perception* of drops in such a way that, when the drop makes the (probabilistic) decision of where to move next, it will perceive as if the edge leading to N were C units longer than it actually is, where C is the load cost. Since the chances to take a path depend on the gradient, which in turn depends on the edge cost, this will reduce the probability to go to N. In fact, the drop will choose to go to N only if it is a good choice *despite of* the additional cost. Moreover, the erosion introduced after the drop moves to N, which in turn also depends on the edge gradient, will be calculated as if the edge cost were C units longer.

On the other hand, when a drop reaches the *sea*, some additional erosion will be introduced in nodes traversed so far by the drop (in addition to the erosion already produced by the drop in previous movements). This additional erosion will be proportional to the suitability of the (possibly partial) *solution* found by the drop. In order to reduce the number of individual drops movements, drops reaching some node at the same step will be joined into a single *bigger* drop. Thus, a drop reaching the sea is actually the fusion of several drops that departed at different nodes. By considering the edges traversed by each of these subdrops, we will depict a *tree*. We can evaluate the suitability of this tree, based on two factors: (a) the *proportion* of critical nodes and transitions, out of the total amount of critical nodes an transitions, actually traversed by this tree; and (b) the cost of the tree. If the proportion of critical nodes is close to 1, then the traversed tree is near to be a solution. Thus, edges in the tree are further reinforced (eroded). Besides, if this solution is competitive indeed (i.e. it has lower cost) then this incentive is further increased.

4 Experimental Results

In this section we report the experimental results we collected after applying our method to solve `MLS` for some weighted finite state machines. Our experiments have two goals: (a) showing that being able to load previously traversed states may allow to reduce the time needed to cover some states and transitions; and (b) showing that solutions provided by our heuristic method are good enough though not being optimal. Regarding (a), we compare the time required to cover some critical configurations in the cases where load operations are allowed and not allowed, respectively. This comparison is made for three load cost assumptions: the load cost is similar to the cost of taking a few edges (so loading is usually preferable); the load cost is a bit less than the cost of the shortest path between two distant nodes (so loading is seldom preferable); and an intermediate approach. In the alternative case where we cannot restore configurations, we will assume that a *reliable reset button* is available, which is a typical assumption in testing methods. In fact, this is equivalent to assuming that we can restore only the initial state of the machine. An adapted version of RFD, similar to the one given in [14], is used to find solutions in this alternative case. Regarding (b), for some small graphs we compare the performance and optimality of results given by our method with those given by an optimal branch and bound (B&B) strategy.

All experiments were performed in an Intel T2400 processor with 1.83 Ghz. RFD was executed thirty times for each of the graphs during one minute, while the B&B method was executed (only once) during one hour (note that B&B is deterministic, so running it more times is pointless). For each method, the following table summarizes the best solution found by each algorithm, the arithmetic mean, the variance, and the best solution found for the alternative case where we cannot load any state different to the initial state.

Method	Graph size	Best solution	Arithmetic mean	Variance	Reset best
RFD	50	1616.81	1815.56	6414.07	2459.65
B&B	50	2158.45	-	-	-
RFD	100	221.00	326.40	3355.51	378.13
B&B	100	4346.45	-	-	-

The '-' symbol denotes a non-applicable case. The input of both algorithms is some randomly generated graph with 50 and 100 nodes where the cost of edges is between 0 and 100. The graph with 50 nodes is sparse (each node is connected with 2-4 nodes) while the graph with 100 nodes is dense (each node is connected with 80% of the nodes). Due to the lack of space, here we only present the results for the case where the load cost is relatively cheap. Similar results have been obtained for other load costs, though the differences between the best solutions found for the cases where we can and we cannot load are reduced as long as the load cost increases. These results can be found in the extended version of the paper [16].

As we can see, cheaper testing plans can be obtained when load operations are available. Besides, though B&B can eventually find the optimal solution, the performance of RFD and B&B for similar execution times is incomparable: Even if we execute B&B for much longer times than RFD, solutions found by RFD clearly outperform those given by B&B. Besides, solutions provided by RFD for the case when (general) loads are forbidden (i.e. we can only reset) are also competitive compared with B&B.

5 Conclusions and Future Work

In this paper we have studied a testing scenario where we can copy and restore previously traversed configurations of the IUT at some cost. This feature is specially interesting if operations performed by the IUT are time-expensive but the memory used by it is, in comparison, not so high. In this case, using load/restore operations allows to reduce the time needed to cover some given critical IUT configurations, because loading will often be cheaper than repeating some already traversed path. Finding optimal plans to interact with the IUT for this particular testing case is an NP-hard problem (we present a proof in [16]), so an heuristic approach is required to find good test sequences (in particular, good test *trees*). Thus, in order to solve this problem, we have applied an evolutionary computation approach, in particular *River Formation Dynamics*. Experimental results show that (a) loading is a good choice even in scenarios where the cost of loading is not particularly low; and (b) the river formation dynamics implementation provides a good tradeoff between (partial) optimality of results and the time needed to achieve them. As future work, we want to design an adaptation of ACO to the `MLS` problem and compare its performance with RFD.

References

1. Aho, A., Dahbura, A., Lee, D., Uyar, M.Ü.: An optimization technique for protocol conformance test generation based on UIO sequences and Rural Chinese Postman tours. In: Protocol Specification, Testing and Verification VIII, pp. 75–86. North-Holland, Amsterdam (1988)
2. Alba, E., Chicano, F.: Finding safety errors with ACO. In: GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 1066–1073. ACM Press, New York (2007)

3. Brinksma, E., Tretmans, J.: Testing transition systems: An annotated bibliography. In: Cassez, F., Jard, C., Rozoy, B., Dermot, M. (eds.) MOVEP 2000. LNCS, vol. 2067, pp. 187–195. Springer, Heidelberg (2001)
4. Dahbura, A., Uyar, M.: Optimal test sequence generation for protocols: The chinese postman algorithm applied to Q.931. In: Conformance testing methodologies and architectures for OSI protocols, pp. 347–351. IEEE Computer Society Press, Los Alamitos (1995)
5. Dorigo, M.: Ant Colony Optimization. MIT Press, Cambridge (2004)
6. Dorigo, M., Gambardella, L.M.: Ant colonies for the traveling salesman problem. BioSystems 43(2), 73–81 (1997)
7. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
8. Huang, G.-D., Wang, F.: Automatic test case generation with region-related coverage annotations for real-time systems. In: Peled, D.A., Tsay, Y.-K. (eds.) ATVA 2005. LNCS, vol. 3707, pp. 144–158. Springer, Heidelberg (2005)
9. De Jong, K.A.: Evolutionary computation: a unified approach. MIT Press, Cambridge (2006)
10. Lee, D., Yannakakis, M.: Principles and methods of testing finite state machines: A survey. Proceedings of the IEEE 84(8), 1090–1123 (1996)
11. Miller, R.E., Arisha, K.A.: Fault coverage in networks by passive testing. In: International Conference on Internet Computing 2001, IC 2001, pp. 413–419. CSREA Press (2001)
12. Petrenko, A.: Fault model-driven test derivation from finite state models: Annotated bibliography. In: Cassez, F., Jard, C., Rozoy, B., Dermot, M. (eds.) MOVEP 2000. LNCS, vol. 2067, pp. 196–205. Springer, Heidelberg (2001)
13. Rabanal, P., Rodríguez, I., Rubio, F.: Using river formation dynamics to design heuristic algorithms. In: Akl, S.G., Calude, C.S., Dinneen, M.J., Rozenberg, G., Wareham, H.T. (eds.) UC 2007. LNCS, vol. 4618, pp. 163–177. Springer, Heidelberg (2007)
14. Rabanal, P., Rodríguez, I., Rubio, F.: Finding minimum spanning/distances trees by using river formation dynamics. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217. Springer, Heidelberg (2008)
15. Rabanal, P., Rodríguez, I., Rubio, F.: Applying river formation dynamics to solve NP-complete problems. In: Choing, R. (ed.) Nature-Inspired Algorithms for Optimisation. Studies in Computational Intelligence, vol. 193. Springer, Heidelberg (2009)
16. Rabanal, P., Rodríguez, I., Rubio, F.: Testing restorable systems by using RFD: Extended version (2009), `http://kimba.mat.ucm.es/prabanal/`
17. Rodríguez, I., Merayo, M.G., Núñez, M.: $\mathcal{HOTL}$: Hypotheses and observations testing logic. Journal of Logic and Algebraic Programming 74(2), 57–93 (2008)
18. Shen, Y., Lombardi, F.: Graph algorithms for conformance testing using the rural chinese postman tour. SIAM Journal on Discrete Mathematics 9, 511–528 (1996)
19. Zhu, H., Hall, P.A.V., May, J.H.R.: Software unit test coverage and adequacy. ACM Computing Surverys 29(4), 366–427 (1997)

RCGA-S/RCGA-SP Methods to Minimize the Delta Test for Regression Tasks

Fernando Mateo[1], Dušan Sovilj[2], Rafael Gadea[1], and Amaury Lendasse[2]

[1] Institute of Applications of Information Technologies and Advanced
Communications, Polytechnic University of Valencia, Spain
`fermaji@upvnet.upv.es, rgadea@eln.upv.es`
[2] Laboratory of Information and Computer Science,
Helsinki University of Technology, Finland
`dusans@cis.hut.fi, lendasse@hut.fi`

Abstract. Frequently, the number of input variables (features) involved in a problem becomes too large to be easily handled by conventional machine-learning models. This paper introduces a combined strategy that uses a real-coded genetic algorithm to find the optimal scaling (RCGA-S) or scaling + projection (RCGA-SP) factors that minimize the Delta Test criterion for variable selection when being applied to the input variables. These two methods are evaluated on five different regression datasets and their results are compared. The results confirm the goodness of both methods although RCGA-SP performs clearly better than RCGA-S because it adds the possibility of projecting the input variables onto a lower dimensional space.

Keywords: real-coded genetic algorithm, global search, variable selection, delta test, input scaling, input projection.

1 Introduction

The size of datasets often compromises the models that can employ them for a determined regression or classification task. A linear increase in the number of variables results in an exponential increase in the necessary number of samples to successfully represent the solution space. This burden is called curse of dimensionality [1] and affects many real-life problems usually characterized by a high number of features. In these cases, it is highly convenient to reduce the number of involved features in order to reduce the complexity of the required models and to improve the interpretability.

In the recent years, many studies have intended to address variable selection for regression using a variety of search strategies and convergence criteria. One of the most successful criteria to determine the optimal set of variables in regression applications is a nonparametric noise estimator called Delta Test (DT) ([2], [3]).

With regard to the search strategy, some authors propose local search strategies for DT minimization (e.g. forward search [4], backward search, forward-backward search ([5], [6])), because of their high speed, but they suffer from

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 359–366, 2009.
© Springer-Verlag Berlin Heidelberg 2009

severe sensitivity to local minima. do not explore all the input search space efficiently. Global search strategies (e.g. exhaustive search, tabu search [7], genetic algorithms (GA) [8]) explore more efficiently the solution space but are much slower too. A tabu based approach to DT minimization has been reported in [6] and [9]. Parallel schemes that combine tabu and GAs have also been implemented to ease the slow convergence drawback [6].

This paper aims to optimize the choice of relevant inputs in an automated manner, by introducing a combination of a GA-based global search strategy with two different fitness approaches: scaling and scaling enhanced with projection. The use of real-valued scaling factors is already a great improvement that minimizes the DT beyond the limit imposed by pure selection, because a variable can not only be selected or not, but also be given a weight according to its relative importance. Projection takes a further step as it includes the possibility of projecting the input vectors into a lower dimensional space. Both methods have been compared in [4] using a forward search method but their integration in a global search framework remains unexplored so far.

This paper is organized as follows: Section 2 introduces the DT and its theoretical background. Section 3 describes the developed genetic algorithm and its main parameters, paying special attention to the two custom fitness functions created. Section 4 presents a performance study of both methods on a variety of datasets and discusses the results. Finally, Section 5 summarizes the conclusions.

2 The Delta Test

The DT, firstly introduced by Pi and Peterson for time series [2] and proposed for variable selection in [10], is a technique to estimate the variance of the noise, or the mean squared error (MSE), that can be achieved without overfitting. Given N input-output pairs $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, the relationship between $\boldsymbol{x}_i$ and y_i can be expressed as

$$y_i = f(\boldsymbol{x}_i) + r_i, \qquad i = 1, ..., N \ , \tag{1}$$

where f is the unknown function and r is the noise. The DT estimates the variance of the noise r.

The DT is useful for evaluating the nonlinear correlation between input and output variables. It can also be applied to input variable selection: the set of input variables that minimizes the DT is the selected one. The DT is based on hypotheses coming from the continuity of the regression function. If two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ are close in the input variable space, the continuity of regression function implies the outputs $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ will be close enough in the output space. If this is not accomplished, it is due to the influence of the noise.

The DT can be interpreted as a particularization of the Gamma Test (GT) [11] considering only the first nearest neighbor. This yields a fully nonparametric method as it removes the only hyperparameter (number of neighbors) that had to be chosen for the GT. Let us denote the nearest neighbor of a point $\boldsymbol{x}_i \in \mathbb{R}^d$ as $\boldsymbol{x}_{NN(i)}$. The nearest neighbor formulation of the DT estimates $\mathrm{Var}[r]$ by

$$\mathrm{Var}[r] \approx \delta = \frac{1}{2N} \sum_{i=1}^{N} (y_i - y_{NN(i)})^2 , \tag{2}$$

where $y_{NN(i)}$ is the output of $\boldsymbol{x}_{NN(i)}$. For a proof of convergence the reader should refer to [11].

3 Real-Coded Genetic Algorithms for Global Search

The use of GAs for variable selection has been widely reported in the literature ([12], [13], [14], [15], [16]). The purpose of the GA in this work is the global optimization of the scaling weights and projection matrix that minimize the DT when applied to the input vectors. This study intends to find the optimal DT value in a fixed number of generations. Pure selection would clearly outperform scaling in terms of speed but the best DT found is often sub-optimal. Scaling or projection are necessary to get closer to the optimal set of solutions. For that reason, a real-coded GA (RCGA) is proposed to optimize a population of chromosomes that encode arrays of potential solutions. The two following Subsections describe the fitness functions that were built and applied to the RCGA: one for scaling and another combining scaling and projection.

3.1 Real-Coded Genetic Algorithm with Scaling: RCGA-S

The target of performing scaling is to optimize the value of the DT beyond the minimum value that can be obtained with pure selection. When performing scaling, the selected variables are weighted according to their influence on the output variable. Let us consider f as the unknown function that determines the relationship between the N input-output pairs of a regression problem, $y = f(\boldsymbol{x}) + r$, with $\boldsymbol{x} \in \mathbb{R}^d, y \in \mathbb{R}$ and $r \in \mathbb{R}$ is a random variable that represents the noise. Thus, the estimate of the output, $\hat{y} \in \mathbb{R}$, can be expressed as $\hat{y} = g(\boldsymbol{x_s}) + r$, with $\boldsymbol{x_s} = \boldsymbol{s} \cdot \boldsymbol{x} \in \mathbb{R}^d$ and g is the model that best approximates the function f. The objective is to find a scaling vector $\boldsymbol{s} \in \mathbb{R}^d$ such that

$$\hat{y} = g(s_1 x_1, s_2 x_2, \ldots, s_d x_d) + r \tag{3}$$

minimizes $\mathrm{Var}[r]$ for the given problem.

In the existing variable selection literature there are several applications of scaling to minimize the DT, but often keeping a discrete number of weights ([4], [5], [6]) instead of using unconstrained real values like in this study. In each generation, each individual (array of scaling factors) is multiplied element by element by the i-th input sample from the dataset:

$$X^i_{S(1\times d)} = X^i_{(1\times d)} \times S_{(1\times d)}, \quad i = 1, \ldots, N , \tag{4}$$

where X is the $N \times d$ input matrix, X_S is the scaled version of X and S is the scaling vector.

The DT is calculated by obtaining the Euclidean distances among the weighted input samples X_S. Once done this, the first nearest neighbor of each point is selected and the DT is obtained from their corresponding outputs, according to Eq. 2. When a predefined number of generations has been evaluated, the GA returns the fittest individual and its corresponding DT.

3.2 Real-Coded Genetic Algorithm with Scaling + Projection: RCGA-SP

A projection can be used to reduce the number of variables by applying a linear (idempotent) transformation, represented by a matrix $P_{(d\times k)}$, to the matrix of input samples $X_{(N\times d)}$, resulting in a lower dimensional matrix $X_{P(N\times k)}, k < d$:

$$X_{P(N\times k)} = X_{(N\times d)} \times P_{(d\times k)} \ . \tag{5}$$

Although it might seem counterproductive, the idea if the developed method that combines scaling and projection is to add a new variable to the input space (the projection of the input vectors on one dimension, i.e. with $k = 1$). Equations 6 and 7 describe this approach:

$$X_{P(N\times 1)} = X_{(N\times d)} \times P_{(d\times 1)} \ , \tag{6}$$

$$X_{SP(N\times(d+1))} = [X_{S(N\times d)}, X_{P(N\times 1)}] \ , \tag{7}$$

where X_S is the scaled version of X as calculated in equation 4, X_P is the projected version of X and X_{SP} is the new scaled/projected input matrix. In this case, the length of the chromosome will be twice the length of the ones used for the scaling approach, i.e. $2d$, as a global optimization of the projection vector P must be carried out along with the optimization of the scaling vector S.

4 Experiments

The experiments were carried out using MATLAB 7.5 (R2007b, The Mathworks Inc., Natick, MA, USA), partly using the Genetic Algorithm and Direct Search Toolbox v2.2, and several custom functions. The parts of the code that are critical for speed, like the computation of pairwise distances among points, were coded in C++ and compiled as MATLAB executables (MEX).

The populations are initially created using a custom function that assigns a uniform initialization to a percentage of the population and the rest can be customized by the user, specifying how many of the remaining individuals are initialized randomly and how many of them are left as zeros. The function is flexible in the sense that the custom percentage of the initial population can be further split into more subsets, each one with a customizable percentage of randomly initialized individuals.

The crossover and mutation operators have also been implemented as custom functions. The mutation operator is a pure random uniform function whereas the crossover operator was BLX-α [17] because of its better performance compared to the one-point, two-point and uniform crossover operators [8]. Regarding the selection operator, the binary tournament was chosen because it outperformed the roulette wheel. Three population sizes were tested: 50, 100 and 150 individuals. Values higher than 150 were discarded in order to keep reasonable runtimes. To sum up, the GA parameters were set as follows:

- Number of averaged runs: 10
- Number of generations evaluated: 50
- Population sizes: 50, 100, 150
- Population initialization: 20% uniform / 80% custom (with 90% zeros and 10% random genes)
- Crossover operator: BLX-α (α=0.5)
- Selection function: Binary tournament
- Crossover rate: 0.85
- Mutation rate: 0.1
- Elitism: 10%
- Mutation function: Random uniform
- Fitness function: S/SP

4.1 Datasets

The described methods (RCGA-S and RCGA-SP) have been evaluated on five regression datasets with different sample/variable ratios to assess their performance in different types of scenarios. The dimensionality and number of samples of each dataset are listed in Table 1. Santa Fe and ESTSP 2007 are time series, so regressors of 12 and 55 variables, respectively, were built.

Table 1. Datasets used in the experiments

Dataset	Instances	Input variables
Boston Housing[1]	506	13
Tecator[2]	215	100
Anthrokids [3]	1019	53
Santa Fe[4]	987	12
ESTSP 2007[3]	819	55

[1] http://archive.ics.uci.edu/ml/datasets/Housing
[2] http://lib.stat.cmu.edu/datasets/tecator
[3] http://www.cis.hut.fi/projects/tsp/index.php?page=timeseries
[4] http://www-psych.stanford.edu/∼andreas/Time-Series/SantaFe.html

All datasets were normalized to zero mean and unit variance. Therefore, all DT values shown in this paper are normalized by the variance of the output. The normalization was done variable-wise for all datasets except for Tecator, in which variable selection works better with sample-wise normalization.

4.2 Results

The results of the experiments appear listed in Table 2. In all tests, the population size of 50 chromosomes gave worse results than 100 or 150. The population size of 150 minimized the DT for most datasets, either with RCGA-S or RCGA-SP. Nonetheless, the values of DT obtained with 100 individuals are often very similar to the ones obtained with 150, and the high increase in computational time might not always be worthwhile.

The average, minimum and maximum DT values are improved in all cases by using RCGA-SP instead of RCGA-S. The rate of improvement depends on each particular dataset, and is specially noticeable for Tecator (>64%) or Santa Fe (>20%). Another important result is that the RCGA-S method is more precise than RCGA-SP as the standard deviation is generally lower. Predictably, the fact of doubling the chromosome size increases runtimes too.

An analysis of the initialization function was carried out using the GAs with the best specifications among the tested (Population size = 150). The results, for several custom/uniform initialization ratios, are listed in Tables 3 and 4. The best mean DT for each dataset is marked in bold. As before, 90% of the custom part was composed of zeros while the remaining 10% was randomized.

The results confirm the goodness of the custom initialization with respect to the pure uniform, as the mean DT is reduced in most cases when a high rate of custom-initialized genes is used. Again, the best improvement is found for Tecator dataset (>68% in some cases). Comparing with the existing methodologies (forward-backward search, tabu search or combined tabu + GA search), the results have improved for all tested datasets as it can be observed in [6].

Table 2. Performance of RCGA-S and RCGA-SP after 50 generations

Method	Pop. size	Measurement	Housing	Tecator	Anthrokids	Santa Fe	ESTSP
RCGA-S	50	Mean±StDev DT ($\times 10^{-4}$)	570±14	108±11	75±3	107±12	127.5±1.3
		Min DT ($\times 10^{-4}$)	544.54	92.60	71.91	89.99	125.01
		Max DT ($\times 10^{-4}$)	583.09	132.75	79.37	119.73	128.74
	100	Mean±StDev DT ($\times 10^{-4}$)	559±7	98±13	72.0±1.9	92±12	123.8±2.1
		Min DT ($\times 10^{-4}$)	544.78	75.74	69.13	77.33	120.65
		Max DT ($\times 10^{-4}$)	569.82	109.52	74.83	111.41	126.78
	150	Mean±StDev DT ($\times 10^{-4}$)	553±16	98±12	71.8±1.2	85±8	123.7±2.1
		Min DT ($\times 10^{-4}$)	528.47	83.60	69.81	75.46	121.43
		Max DT ($\times 10^{-4}$)	578.18	113.47	72.98	98.54	128.93
RCGA-SP	50	Mean±StDev DT ($\times 10^{-4}$)	570±60	39±3	71±5	83±15	125±5
		Min DT ($\times 10^{-4}$)	523.56	35.02	66.08	69.96	119.02
		Max DT ($\times 10^{-4}$)	692.22	43.53	77.49	111.66	132.3
	100	Mean±StDev DT ($\times 10^{-4}$)	537±22	38.1±2.4	69±4	72±8	123±4
		Min DT ($\times 10^{-4}$)	512.47	35.14	62.54	62.15	116.71
		Max DT ($\times 10^{-4}$)	583.14	43.36	75.33	82.85	128.97
	150	Mean±StDev DT ($\times 10^{-4}$)	530±17	36.8±2.1	69±4	68±5	122±4
		Min DT ($\times 10^{-4}$)	514.15	33.12	61.82	62.83	115.62
		Max DT ($\times 10^{-4}$)	557.88	40.74	73.71	77.86	129.18

Table 3. Mean and standard deviation of DT values ($\times 10^{-4}$) calculated by RCGA-S for several initialization ratios (Population size = 150)

Custom/Uniform	Housing	Tecator	Anthrokids	Santa Fe	ESTSP
0%/100%	554±5	135.2±1.2	83±3	100±7	123.0±0.9
10%/90%	551±4	133.3±2.0	79±3	99±9	**121.7**±1.9
20%/80%	553±8	127±9	77±3	96±10	123.4±1.0
30%/70%	554±11	124.5±7	76±3	99±11	123.0±1.3
40%/60%	550±9	118±10	75±3	91±9	123.0±1.6
50%/50%	552±8	109±12	72.7±2.0	91±9	122.8±1.3
60%/40%	549±11	110±7	72.7±2.0	91±9	122.4±1.8
70%/30%	**548**±11	105±9	73.3±2.1	87±6	123.5±1.0
80%/20%	553±16	98±12	71.8±1.2	85±8	123.7±2.1
90%/10%	549±10	92±11	**70.9**±0.9	**80**±5	123.0±1.2
100%/0%	605±40	**87**±10	72.60±0.10	**80**±6	125.6±2.1

Table 4. Mean and standard deviation of DT values ($\times 10^{-4}$) calculated by RCGA-SP for several initialization ratios (Population size = 150)

Custom/Uniform	Housing	Tecator	Anthrokids	Santa Fe	ESTSP
0%/100%	543±19	42.2±1.4	72±3	77±12	**119**±3
10%/90%	537±18	40.8±1.6	78±8	83±13	120±3
20%/80%	535±16	41.3±1.9	82±11	76±10	119.8±1.7
30%/70%	539±18	40.0±2.5	86±14	85±21	120±3
40%/60%	531±15	40±4	80±8	76±10	119.2±1.8
50%/50%	541±22	38.7±1.9	80±7	67±4	120±3
60%/40%	536±13	38.9±2.1	76±8	71±8	122±3
70%/30%	540±21	40±3	77±8	68±5	**119**±3
80%/20%	**530**±17	36.8±2.1	69±4	68±5	122±4
90%/10%	539±23	36.2±2.0	66±6	68±5	123±3
100%/0%	555±21	**34.8**±1.6	**65**±6	**66**±3	124±4

5 Conclusions

The methodology presented is a combination of a real-coded GA with custom fitness functions that perform scaling (RCGA-S) and scaling + projection (RCGA-SP), which has proved to accurately minimize the DT in a variety of scenarios. In particular, the RCGA-SP method has proved to find lower values of DT than RCGA-S in all tests. Furthermore, the proposed custom initialization enables a refinement of the final value of DT obtained and performs better than the uniform initialization in most cases. The minimum DT values found are lower than the lowest values attained in previous works, either using local or global search strategies ([4], [6]) for the tested datasets. The main drawback of RCGA-SP is its higher computational time, but this issue could be alleviated in the future using parallel implementations. Scaling + projection to higher dimensional spaces ($k > 1$) is also to be further examined.

Acknowledgments. Fernando Mateo received financial support from the Spanish Ministry of Science and Innovation (projects AGL 2004-07549-C05-02/ALI, FPA 2007-65013-C02-02 and a research grant).

References

1. Verleysen, M., François, D.: The curse of dimensionality in data mining and time series prediction. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 758–770. Springer, Heidelberg (2005)
2. Pi, H., Peterson, C.: Finding the embedding dimension and variable dependencies in time series. Neural Computation 6(3), 509–520 (1994)
3. Liitiäinen, E., Corona, F., Lendasse, A.: On nonparametric residual variance estimation. Neural Processing Letters (to be published) (2008)
4. Yu, Q., Séverin, E., Lendasse, A.: A global methodology for variable selection: application to financial modeling. In: Proc. of MASHS 2007, ENST-Bretagne, France (May 2007)
5. Mateo, F., Lendasse, A.: A variable selection approach based on the delta test for extreme learning machine models. In: Proc. of ESTSP 2008, European Symposium on Time Series Prediction, Porvoo, Finland, September 2008, pp. 57–66 (2008)
6. Guillén, A., Sovilj, D., Mateo, F., Rojas, I., Lendasse, A.: Minimizing the delta test for variable selection in regression problems. Int. J. on High Performance Systems Architecture (in press, 2009)
7. Glover, F., Laguna, F.: Tabu Search. Kluwer Academic Publishers, Norwell (1997)
8. Holland, J.: Adaption in natural and artificial systems. University of Michigan Press (1975)
9. Sovilj, D., Sorjamaa, A., Miche, Y.: Tabu search with delta test for time series prediction using OP-KNN. In: Proc. of ESTSP 2008 European Symposium on Time Series Prediction, Porvoo, Finland, pp. 187–196 (September 2008)
10. Eirola, E., Liitiäinen, E., Lendasse, A., Corona, F., Verleysen, M.: Using the Delta Test for Variable Selection. In: Proc. of ESANN 2008, European Symposium on Artificial Neural Networks, Bruges, Belgium, pp. 25–30 (April 2008)
11. Jones, A.: New tools in non-linear modelling and prediction. Computational Management Science 1(2), 109–149 (2004)
12. Oh, I.S., Lee, J.S., Moon, B.R.: Local search-embedded genetic algorithms for feature selection. In: Proc. of the 16th Int. Conference on Pattern Recognition, vol. 2, pp. 148–151 (2002)
13. Oh, I.S., Lee, J.S., Moon, B.R.: Hybrid genetic algorithms for feature selection. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(11), 1424–1437 (2004)
14. Punch, W., Goodman, E., Pei, M., Chia-Shun, L., Hovland, P., Enbody, R.: Further research on feature selection and classification using genetic algorithms. In: Forrest, S. (ed.) Proc. of the Fifth Int. Conf. on Genetic Algorithms, San Mateo, CA, pp. 557–564. Morgan Kaufmann, San Francisco (1993)
15. Raymer, M., Punch, W., Goodman, E., Kuhn, L., Jain, A.: Dimensionality reduction using genetic algorithms. IEEE Trans. on Evolutionary Computation 4(2), 164–171 (2000)
16. Saeys, Y., Inza, I., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
17. Eshelman, L., Schaffer, J.: Real-coded genetic algorithms and interval schemata. In: Darrell Whitley, L. (ed.) Foundation of Genetic Algorithms, vol. 2, pp. 187–202. Morgan-Kauffman Publishers, Inc., San Francisco (1993)

An Evolutionary Hierarchical Clustering Method with a Visual Validation Tool

José A. Castellanos-Garzón, Carlos Armando García,
and Luis A. Miguel-Quintales

University of Salamanca, Department of Computer Science and Automatic,
Faculty of Sciences, Plaza de los Caídos s/n, 37008 Salamanca, Spain
{`jantonio,CarlosGarcia,lamq`}`@usal.es`
`http://informatica.usal.es`

Abstract. In this paper, we propose a novel hierarchical clustering method based on evolutionary strategies. This method leads to gene expression data analysis, and shows its effectiveness with regard to other clustering methods through cluster validity measures on the results. Additionally, a novel visual validation interactive tool is provided to carry out visual analytics among clusters of a dendrogram. This interactive tool is an alternative for the used validity measures. The method introduced here attempts to solve some of the problems faced by other hierarchical methods. Finally, the results of the experiments show that the method can be very effective in the cluster analysis on DNA microarray data.

Keywords: Hierarchical clustering, DNA microarray, evolutionary algorithm, cluster validity, visual analytics.

1 Introduction

Evolutionary algorithms ([1,2]) are stochastic search methods based on the natural selection principle, where the genetic information of the individuals (candidate solutions) is recombined in order to create fitter individuals. *Genetic algorithms* (GAs) in evolutionary algorithms, are the paradigm with better theoretical basis of the whole *evolutionary computation*. GAs represent an efficient, adaptive, and robust search for the optimization processes that are usually applied on very large, complex and multimodal search spaces.

On the other hand, the *hierarchical cluster analysis* ([3,4]) is a powerful solution for showing data (particularly, biological data) using visual representations, where data can easily be interpreted through a special type of tree structures that show cluster relationships, such as a *dendrogram* representation.

GAs application to *data mining* is very important in knowledge discovery through the study of classification problems, where an optimization criterium is required. Consequently, GAs applied to clustering problems ([5]) are very important in the *DNA microarray data analysis* [6]. The above researches have shown understanding of the gene function, gene regulation, cellular processes, and subtypes of cells.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 367–374, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The goal of this research is to explore the dendrogram search space on DNA-microarray data in search of the best dendrogram. The above problem is *NP-Complete*, and thereby arises the need of introducing methods that do not consider every solution in the search space, such as evolutionary techniques. In addition, a visual validation interactive tool was implemented to explore the quality of the dendrograms returned by the method. This tool is based on the principal component analysis (PCA, [7]) to reduce the data dimensionality, so that a first approximation of data distribution is shown. Furthermore, a visualization on parallel coordinates ([8]) and a view of DNA-microarray data, assigning a scale of colors to gene expression levels in each cluster, is presented in an interactive way.

2 A Genetic Algorithm

In order to model this problem, the individuals (*chromosomes*) represent dendrograms on a given data set, encoded as an ordered set of nested clusterings, where each clustering has an order number, called level, [9]. Initially, each dendrogram of a population is built up from an initial level to a higher level by joining two clusters chosen randomly in a level, thereby, the next level is built. Thus, this method is aimed at the search of high quality clustering dendrograms.

Definition 1. *Dendrogram length.*
Set $\mathfrak{P}_n$ be a data set of size n and set $\mathfrak{G}$ be a dendrogram on $\mathfrak{P}_n$, then the length of $\mathfrak{G}$ is the clustering number of it and is defined as:

$$|\mathfrak{G}| = n - \lfloor n \cdot \delta \rfloor - 2, \tag{1}$$

where δ[1], is the part of $\mathfrak{G}$ to remove, assuming $\delta \geq 1/2$.

2.1 Fitness Function

In every GA it is necessary to measure the goodness of the candidate solutions. In this problem, the fitness of a dendrogram must be evaluated. namely, based on one of the given definitions of cluster in [9] (*the objects inside of a cluster are very similar, whereas the objects located in distinct clusters are very different*), the fitness function is defined according to the concepts of both, *homogeneity* and *separation*, introduced in [10].

Definition 2. *Clustering homogeneity.*
If $\mathfrak{D} = [d(i,j)]$ is the proximity matrix on the $\mathfrak{P}_n$ data set, being d the defined metrics on this data set, $\mathfrak{C}$ a clustering of objects in $\mathfrak{P}_n$, C a cluster into $\mathfrak{C}$ and $m = |C|$, then the homogeneity of C is:

$$h(C) = \frac{2}{m \cdot (m-1)} \sum_{i \neq j}^{m \cdot (m-1)/2} d(i,j), (\forall i, j \in C), \tag{2}$$

---

[1] Is the fraction of $\mathfrak{G}$ that does not give information.

if $k = |\mathfrak{C}|$ then the homogeneity of $\mathfrak{C}$ is:

$$\mathcal{H}(\mathfrak{C}) = \frac{1}{k} \sum_{i=1}^{k} h(C_i). \tag{3}$$

Definition 3. *Clustering separation.*
Set $\mathfrak{C}$ be a clustering of $\mathfrak{P}_n$, set C_1 and C_2 be two clusters of $\mathfrak{C}$, then the distance d_m between these clusters is defined as:

$$d_m(C_1, C_2) = \min\{d(i,j)/i \in C_1, j \in C_2\}, \tag{4}$$

if $k = |\mathfrak{C}|$ then the $\mathfrak{C}$ separation is:

$$\mathcal{S}(\mathfrak{C}) = \frac{2}{k \cdot (k-1)} \sum_{i \neq j}^{k \cdot (k-1)/2} d_m(C_i, C_j), \quad (\forall i, j \in [1, k]). \tag{5}$$

Definition 4. *Dendrogram fitness function.*
Set $\mathfrak{C}$ and $\mathfrak{D}$ be a clustering of objects in $\mathfrak{P}_n$ and the proximity matrix of $\mathfrak{P}_n$ respectively, then the fitness function of $\mathfrak{C}$ is defined as:

$$f_c(\mathfrak{C}) = \max \mathfrak{D} + \mathcal{S}(\mathfrak{C}) - \mathcal{H}(\mathfrak{C}). \tag{6}$$

If $\mathfrak{C}_i$ is a clustering of a dendrogram $\mathfrak{G}$, then the fitness function of $\mathfrak{G}$ is:

$$f_d(\mathfrak{G}) = \frac{1}{|\mathfrak{G}|} \sum_{i=1}^{|\mathfrak{G}|} f_c(\mathfrak{C}_i). \tag{7}$$

Definition 5. *Agglomerative coefficient.*
Set $\mathfrak{G}$ and $\mathfrak{C}_i$ be a dendrogram on $\mathfrak{P}_n$ and a clustering of $\mathfrak{G}$, respectively. The agglomerative coefficient of $\mathfrak{G}$ is defined as:

$$ac(\mathfrak{G}) = \arg_{i \in [1, |\mathfrak{G}|]} \max f_c(\mathfrak{C}_i), \tag{8}$$

the level i whose clustering has the maximum fitness of the whole dendrogram.

2.2 Improving the Fitness Function Cost

Due to the large runtime of the fitness function defined in ([4]), it was required to decrease the runtime of this function. Hence, the fitness function defined in ([4]) can be transformed in an equivalent but more efficient one, as shown in the following lemmas:

Lemma 1. *Recurrent homogeneity.*
Set $\mathfrak{C}_i$ be a clustering of level i and set $\mathfrak{C}_{i+1}$ be a clustering of level $i+1$, both in the dendrogram $\mathfrak{G}$; C_j and C_l, the two clusters of level i such that its join forms a new clustering $\mathfrak{C}_{i+1}$ of level $i+1$, then the homogeneity of $\mathfrak{C}_{i+1}$ ($\mathcal{H}_1(\mathfrak{C}_{i+1})$) is computed in the following expression:

for $i = 1$, that is, for the first clustering, $\mathcal{H}_1(\mathfrak{C}_1) := \mathcal{H}(\mathfrak{C}_1)$ and for $i > 1$,

$$\begin{aligned}&(k-1)\cdot\mathcal{H}_1(\mathfrak{C}_{i+1}) = k\cdot\mathcal{H}_1(\mathfrak{C}_i) - h(C_j) - h(C_l) + \tfrac{1}{k_3}[k_1\cdot h(C_j) + \\ &+ k_2\cdot h(C_l) + l_1\cdot l_2\cdot d_p(C_j, C_l)],\end{aligned} \tag{9}$$

where $k = |\mathfrak{C}_i|, l_1 = |C_j|, l_2 = |C_l|, k_1 = \binom{l_1}{2}, k_2 = \binom{l_2}{2}$, $k_3 = \binom{l_1\cdot l_2}{2}$, and d_p is the average distance between clusters C_j and C_l.

Lemma 2. *Recurrent separation.*
Keeping the same conditions of the previous lemma, we can obtain the recurrent separation of a clustering $\mathfrak{C}_{i+1}$ $(\mathcal{S}_1(\mathfrak{C}_{i+1}))$:

for $i = 1, S_1(\mathfrak{C}_1) := S(\mathfrak{C}_1)$ and for $i > 1$,

$$\begin{aligned}&(g-k+1)\cdot S_1(\mathfrak{C}_{i+1}) = g\cdot S_1(\mathfrak{C}_i) - d_m(C_j, C_l) - \textstyle\sum_{t\neq j\wedge t\neq l}^{k-2}[d_m(C_j, C_t) + \\ &+ d_m(C_l, C_t) - \min\{d_m(C_j, C_t), d_m(C_l, C_t)\}],\end{aligned} \tag{10}$$

where $k = |\mathfrak{C}_i|, g = \binom{k}{2}$, being g the number of distances among the clusters of $\mathfrak{C}_i$.

Definition 6. *Dendrogram recurrent fitness.*
The fitness function of a clustering $\mathfrak{C}_{i+1}$ of $\mathfrak{G}$, according to $\mathcal{H}_1$ and S_1, is defined as:

$$g_c(\mathfrak{C}_{i+1}) = \max\mathfrak{D} + S_1(\mathfrak{C}_{i+1}) - \mathcal{H}_1(\mathfrak{C}_{i+1}), \tag{11}$$

known $\mathcal{H}(\mathfrak{C}_i)$ and $S(\mathfrak{C}_i)$. The recurrent fitness function of $\mathfrak{G}$ is defined as:

$$g_d(\mathfrak{G}) = \frac{1}{|\mathfrak{G}|-1}\sum_{i=1}^{|\mathfrak{G}|-1} g_c(\mathfrak{C}_i). \tag{12}$$

2.3 Mutation Operator

The mutation of a dendrogram is performed according to the following steps:

1. We will consider two parameters τ and ϵ for each dendrogram $\mathfrak{G}$ where:
    - τ is the percentage of choosing cluster pairs into level i to build the following level $i + 1$;
    - ϵ is a small value that represents the similarity between two clusters, according to the homogeneity measure.
2. A random number $i \in [1, |\mathfrak{G}|]$ is generated, being it the level where the mutation of $\mathfrak{G}$ is carried out.
3. For the clustering of level i in the previous step, one of the following conditions is chosen:
    - the most homogeneous join of cluster pairs of τ% of random cluster pairs is chosen;
    - the cluster pair with a difference less or equal than ϵ from the cluster pair chosen in the above condition is chosen.
4. The cluster pair chosen in the previous step is joined in order to form a new cluster so that the clustering of next level $i + 1$ can be built.
5. The steps 3 and 4 are repeated at the new level, until i reaches level $|\mathfrak{G}|$.

2.4 Crossover Operator

The crossover is carried out on two dendrograms to obtain a child dendrogram, based on the idea of [11], that is:

1. Given two dendrograms $\mathfrak{G}_1$ and $\mathfrak{G}_2$ (parents), a random number i in $[1, |\mathfrak{G}_1|]$ is generated to choose the level where it can carry out the crossover between both dendrograms.
2. Through a strategy of *greedy algorithm*, the best $\lfloor k/2 \rfloor$ clusters[2] of level i of both dendrograms of the above step are chosen, being k the number of clusters of level i. A new clustering is formed by repairing the chosen clusters, [11].
3. As soon as the new clustering for level i is built, it can build up the new dendrogram:
    - levels higher than level i are built using the MO;
    - levels lower than level i are built in a divisive way, that is, for each level less than i, the less homogenous cluster is chosen to be split in two. This process is repeated until reaching the first level of the dendrogram.
4. The parent of the less fitness value is replaced by the child dendrogram of step 3.

3 Case Study on Gene Expression Data

In this section we study the behavior of the GA on a real data set of gene expression data (called, *Sorlie*) and compare the results with other methods according to some cluster validity measures [12,10,13]. Gene expression levels of 456 genes are evaluated under 85 tissue samples. The original data had missing values scattered throughout the matrix. 10-nearest neighbours were used for imputation, as described in [14,15]. This data set was normalized with mean 0 and variance 1. GA was implemented on the *R language* (R Development Core Team), and the visualization tool was developed on *Java Programming Language.*

3.1 Homogeneity, Separation and Silhouette Width

In what follows we have carried out a cluster validity process to compare the results of the GA. Therefore, we have focused on the quality of clusters in terms of *homogeneity* (Homog), *separation* (Separ) [10] and *silhouette width* (SilhoW, [13]), for three non-deterministic methods of hierarchical clustering; *Eisen* [3], *HybridHclust* [16] and *Tsvq* [17].

The GA was initialized with a population of 10 individuals, $\delta = 20/21$, the values of the crossover and mutation likelihood were assigned around 0.60 and 0.05 respectively, the generation number to 10^4, $\tau \in [0.15, 0.40]$, $\epsilon = 0.03$, $x = 0.90$ and the *euclidian distance* was used on the data set. The GA output was extracted to make comparisons, such as listed in table 1, where the GA is compared

---

[2] The most homogenous clusters of both dendrograms.

Table 1. Cluster validity of GA vs. three hierarchical clustering methods

Method	g_d	Cluster (ac)	f_c	Homog	Separ	SilhoW
Eisen	33.03	20	34.85	16.05	17.52	-0.08
HybridHclust	36.14	21	39.53	14.46	17.46	0.01
Tsvq	36.21	21	40.70	14.28	17.52	0.02
GA	47.14	6	50.70	17.02	21.38	0.12

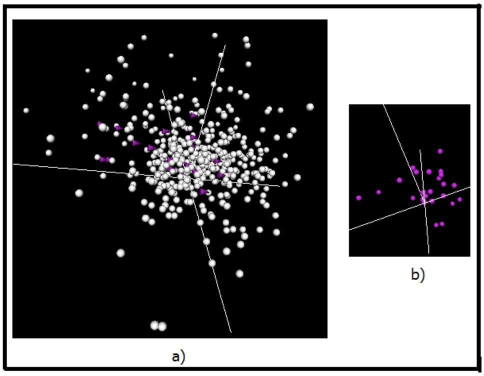


Fig. 1. a) 3D data points of sorlie data set, b) 3D data points of a cluster of 24 genes of sorlie data

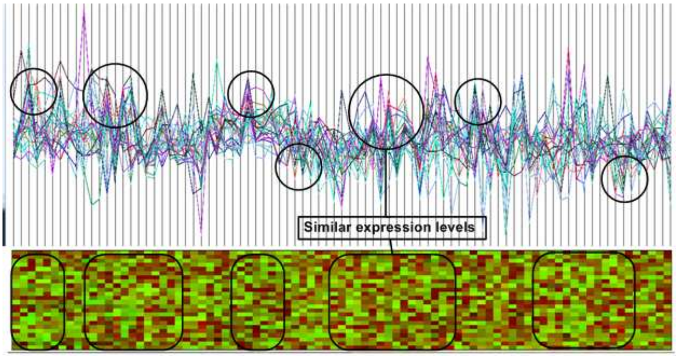


Fig. 2. Cluster of Figure 1-b), shown on parallel coordinates and its correspondent view of sorlie microarray data, rows=genes and columns=samples

with three methods according to four measures: the *Cluster* column is the number of the cluster of the best clustering in a dendrogram (using *ac* coefficient). Then, for that same clustering, the other measures located on the right side of the *Cluster* column of the table were computed.

Most clustering methods focus on the goals on homogeneity indicator and not in the cluster global quality. Consequently, table 1 emphasizes that GA performed well for all measures, except for the homogeneity indicator. The introduced GA looks for both indicators, homogeneity and separation.

Consequently, we applied our visualization tool to represent the distribution of the Sorlie dataset in the space and a cluster of the same dataset, as shown in figures 1-a) and 1-b). In addition, a view of the data cluster is displayed in figure 2 using parallel coordinates (curves representing genes and vertical lines representing samples) and as microarray intensity levels, where intensity of color (red=up-regulation, green=down-regulation, black=constitutive expression) is proportional to the differential gene expression. Moreover, in this figure, similar values of gene expression data are highlighted.

4 Conclusion

The main goal of this paper has been to present a new clustering method with GAs, as well as an interactive visualization tool to aggregate, summarize and visualize the information on the clusters of a dendrogram. To conclude, the most important outcomes are:

1. Two fundamental lemmas for reducing the computation complexity of the fitness function;
2. A visualization tool that allows interactivity to observe cluster relationships in a dendrogram;
3. The method performed well almost for all validity measures. Therefore, it can be very useful in the process of gene expression data analysis.

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley Longman, Inc., Amsterdam (1989)
2. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer, New York (1999)
3. Eisen, M., Spellman, T., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences, USA 95, 14863–14868 (1998)
4. Korte, B., Vygen, J.: DHC: A density-based hierarchical clustering method for time series gene expression data. In: Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering, BIBE (2003)
5. Ma, P.C.H., Chan, K.C.C., Yao, X., Chiu, D.K.Y.: An evolutionary clustering algorithm for gene expression microarray data analisys. IEEE Transactions on Evolutionary Computation 10, 296–314 (2006)
6. Berrar, D.P., Dubitzky, W., Granzow, M.: A Practical Approach to Microarray Data Analysis. Kluwer Academic Publishers, Dordrecht (2003)
7. Jolliffe, I.T.: Principal Component Analysis. Springer, Heidelberg (2002)
8. Inselberg, A., Dimsdale, B.: Parallel coordinates: a tool for visualizing multidimensional geometry. In: VIS 1990: Proceedings of the 1st conference on Visualization 1990, pp. 361–378 (1990)

9. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1998)
10. Jiang, D., Tang, C., Zhang, A.: Cluster analysis for gene expression data: A survey 16, 1370–1386 (2004)
11. Greene, W.A.: Unsupervised hierarchical clustering via a genetic algorithm. In: Congress on Evolutionary Computation, CEC 2003, vol. 2, pp. 998–1005. IEEE, Los Alamitos (2003)
12. Handl, J., Knowles, J., Kell, D.B.: Computational cluster validation in post-genomic data analysis, vol. 21, pp. 3201–3212. Oxford University Press, Oxford (2005)
13. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data. In: An Introduction to Clustering Analysis, John Wiley & Sons, Inc., Hoboken (2005)
14. Chipman, H., Hastie, T., Tibshirani, R.: Clustering microarray data. Statistical Analysis of Gene Expression Microarray Data (2003)
15. Sorlie, T., et al.: Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. PNAS 98, 10969–10974 (2001)
16. Chipman, H., Tibshirani, R.: Hybrid hierarchical clustering with applications to microarray data. Biostatistics 7, 302–317 (2006)
17. Macnaughton-Smith, P., Williams, W.T., Dale, M.B., Mockett, L.G.: Dissimilarity analysis: a new technique of hierarchical subdivision. Nature 202, 1034–1035 (1965)

An Adaptive Parameter Control for the Differential Evolution Algorithm

Gilberto Reynoso-Meza, Javier Sanchis, and Xavier Blasco

Instituto Universitario de Automática e Informática Industrial.
Universidad Politécnica de Valencia.
Camino de Vera s/n, 46022 Valencia, España.
`gilreyme@posgrado.upv.es, {jsanchis,xblasco}@isa.upv.es`

Abstract. The Differential Evolution is a floating-point evolutionary algorithm that has demonstrated good performance on locating the global optima in a wide variety of problems and applications. It has mainly three tuning parameters and their choice is fundamental to ensure good quality solutions. Because of this, adaptive parameter control and self-adaptive parameter control had been object of research. We present a novel scheme for controlling two parameters of the Differential Evolution using fitness information of the population in each generation. The algorithm shows outstanding performance on a well known benchmark functions, improving the standard DE and comparable with similar algorithms.

Keywords: Differential Evolution, Evolutionary Algorithms, Adaptive control parameter, evolutionary optimization.

1 Introduction

Differential Evolution Algorithm (DE) is an evolutionary algorithm for global optimization which has demonstrated an outstanding performance in a variety of problems. It was introduced by Price and Storn [1], and has become popular due to its simplicity and powerful capabilities for converging into global solutions.

However, finding suitable parameter values generally requieres experience from the user and a good problem knowledge. There exist recommendations for parameter tuning [2], but choosing the most appropriate values for a particular application still represents a problem [3].

Adaptation mechanisms has been presented to solve the parameter tuning problem. According to [4], the adaptation schemes for evolutionary algorithms can be classified into three categories:

- *Deterministic Parameter Control (DPC):* where the tuning parameter is adjusted by a deterministic rule, without taking account any kind of feedback from the search (for example, changes in parameter values according to a number of elapsed generations).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 375–382, 2009.
© Springer-Verlag Berlin Heidelberg 2009

- *Adaptive Parameter Control (APC):* takes place when information from the search is used to change, refine or redirect a parameter value. Examples in this category are the strategy proposed by Zaharie [5,6], where a diversity control in population is used and the scheme presented by Liu [7,8] named FADE (Fuzzy Adaptive Differential Evolution) which uses fuzzy reasoning for parameter control.
- *Self-Adaptive Parameter Control (SAPC):* represents an idea where the evolution strategy used to find the global optima, uses another level of evolution scheme to propagate promising parameter values among the population. For example, SaDE [9,10,11] is an algorithm that pass throught a learning period in order to determine the probabilities of selecting one of different known version of the DE algorithm (*vid.* section 2) to generate the offspring; jDE [12] and jDE-2 [13] are algorithms that codifies information of F and Cr into each individual in order to adjust those values in the evolution mechanism itself, under the premise that better parameter values lead to better individuals. Recently, a SaDESA hybrid algorithm [14] was presented which incorporates Simulated Annealing and Differential Evolution strategies.

In this work, an Adaptive Parameter Control for DE Algorithm (APC-DE-A) is proposed to improve the DE performance and the quality solution. The rest of the paper is organized as follows: in section 2 the standard DE algorithm is presented and in section 3 related work for adaptive scheme would be briefly presented. The proposal of this paper is described in section 4. Section 5 refers to the experiments carried to show the performance of the algorithm. Finally, conclusions and remarks are given in section 6.

2 The Differential Evolution Algorithm

There are several versions for the DE algorithm. In this work, we present the *DE/rand/1/bin* strategy, which is considered the standard one [1]. Let it be:

- D the dimensionality of the problem
- NP the size of the population
- G the generation
- $x_{i,G}$ a vector which $x_i \in \Re^D$ and $i \in 1, 2, 3 \ldots NP$ at generation G

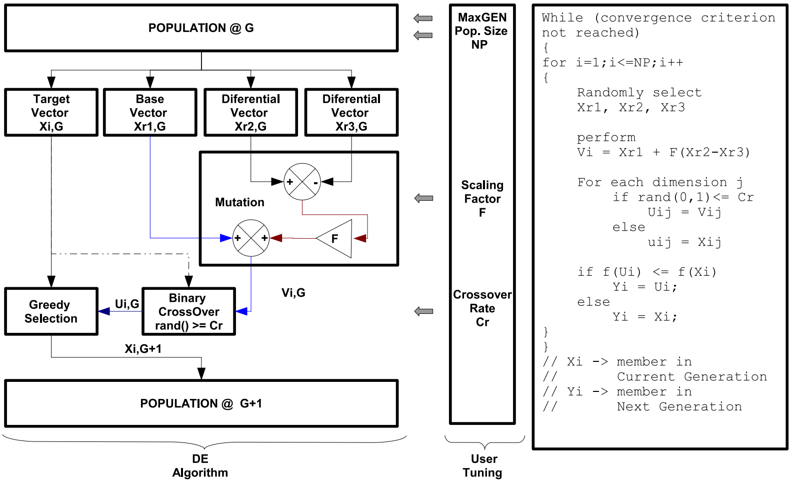
The evolution procedures use three operators: Mutation, Crossover and Selection (*vid.* figure 1).

2.1 Mutation

For each target vector $x_{i,G}$, a mutant vector $v_{i,G}$ is generated according to:

$$v_{i,G} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}) \tag{1}$$

Where $r_1 \neq r_2 \neq r_3 \neq i$ and F is known as the Scaling Factor of the Difference Vector $(x_{r2,G} - x_{r3,G})$ used to perturb a Base Vector $x_{r1,G}$. The vectors $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$ are randomly selected from the population at generation G.



```
While (convergence criterion
not reached)
{
for i=1;i<=NP;i++
{
     Randomly select
     Xr1, Xr2, Xr3

     perform
     Vi = Xr1 + F(Xr2-Xr3)

     For each dimension j
         if rand(0,1)<= Cr
             Uij = Vij
         else
             uij = Xij

     if f(Ui) <= f(Xi)
         Yi = Ui;
     else
         Yi = Xi;
}
}
// Xi -> member in
//       Current Generation
// Yi -> member in
//       Next Generation
```

Fig. 1. The Differential Evolution Algorithm

2.2 Crossover

For each target vector $x_{i,G}$ and its associated mutant vector $v_{i,G}$, a trial vector $u_{i,G}$ is created by sharing information:

$$u_{i,j,G} = \begin{cases} v_{i,j,G} \; if \; rand(0,1) \leq Cr \\ x_{i,j,G} \quad otherwise \end{cases} \tag{2}$$

where $j \in 1, 2, 3 \ldots D$ and Cr is the Crossover probability rate. To garantize a correct performance, at least one dimension need to be shared between target and mutant vector for creating the trial one. Because of the random selection for the Base Vector, the use of one Difference Vector at the mutation step and the binary probability for crossover, the standard DE algorithm is coded as $DE/rand/1/bin$. Other commonly versions used are: $DE/rand/2/bin$, $DE/best/1/bin$, $DE/current - to - best/2/bin$ among others.

2.3 Selection

A greedy selection is used:

$$x_{i,G+1} = \begin{cases} u_{i,G} \; if \; f(u_{i,G}) < f(x_{i,G}) \\ x_{i,G} \quad otherwise \end{cases} \tag{3}$$

According to [1], it is more important to have a good selection on F either than Cr. The regular suggestions for parameter tuning are: $F \in [0.5, 1.0]$, $Cr \in [0.8, 1.0]$ and $NP = 10D$.

3 Adaptive Parameter Control in DE

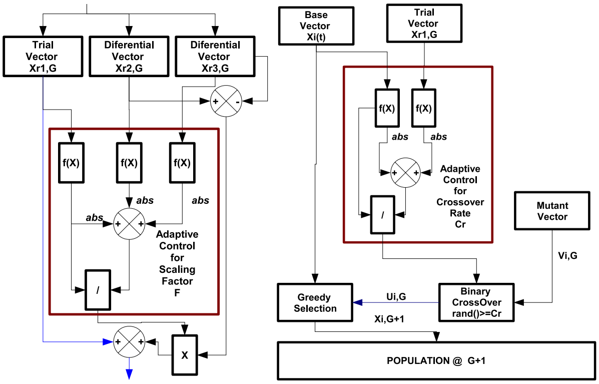
This section presents some Adaptive Parameter Control strategy for the DE.

In [5,6], Zaharie presented theorical results proving that diversity of the trial vector population in one step generation is a function of the target vector population diversity and parameters F, Cr and NP. Those results were used to design an adaptive strategy to maintain population diversity by controlling F value.

Liu and Lampinen [7,8] presented a fuzzy controller scheme for parameter values F and Cr. The design was motivated due to problems of high dimensionality and was tested in a set of ten well known benchmarks functions. The fuzzy adaptive differential evolution outperformed the standard DE with tuned values of $F = 0.9$ and $Cr = 0.9$ when: (a) only F is controlled, (b) only Cr and (c) both of them. The NP parameter was fixed to $10D$.

4 Proposal

Because of the importance and necessity to select parameter values to reach the global minima, this work proposes a novel scheme to adapt F and Cr parameters (*vid.* figure 2) for each mutant and Trial Vector in a given generation, according to information obtained from their fitness in the objective function.



(a) (b)

Fig. 2. Adaptive Parameter Control Scheme for Scaling Factor F (a) and Crossover Rate (b)

4.1 Scaling Factor Control

The objective of the Mutation operator is to exploit the given solution of vectors $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$. The first one, the Base Vector, is perturbed in some amount by the Difference Vector. The idea for controlling parameter F consist in accepting an scaled Difference Vector according to the Base Vector fitness. This can be achieved by:

$$F = \left(\frac{f(x_{r1,G})}{|f(x_{r1,G})| + |f(x_{r2,G})| + |f(x_{r3,G})|} \right)^n \tag{4}$$

According to this, a small value of F in a minimization problem, it is associated to a better possibility for the Base Vector to offer a better fitness compared with x_{r2} and/or x_{r3}. On the other hand, a higher value of F, indicates that there is more possibilities for x_{r2} and x_{r3} to conform a mutant with better fitness. Parameter n represents a fine control adjustment that regulates the convergence speed in the algorithm. In a future work, this parameter will be used to evolve this proposal from APC to a SAPC for DE algorithm. In this work it is not considered, then $n = 1$.

4.2 Crossover Rate Control

The objective in the crossover operator, is to achieve exploration by exchanging information between Target and Mutant Vector. The better the mutant fitness the higher influence of it in the crossover operation. To achieve this an additional evaluation of the objective function $f(v_{i,G})$ would be requiered. Instead, an alternative just using the Target and Base Vector is proposed as follows:

$$Cr = \left(1 - \frac{|f(x_{r1,G})|}{|f(x_{r1,G})| + |f(x_{i,G})|} \right)^m \tag{5}$$

Where parameter m performs the same purpose as n and for the rest of the work $m = 1$.

5 Testing

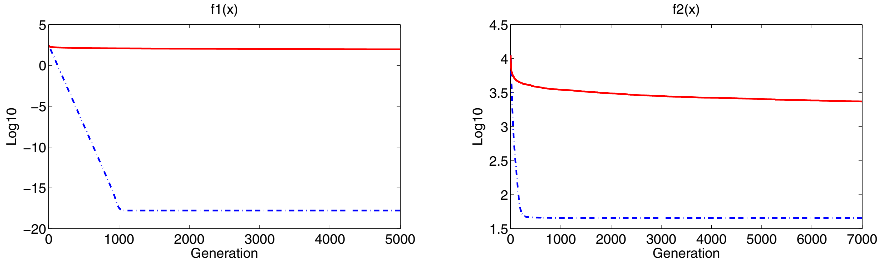
In order to test the performance of the proposed algorithm and for comparison purposes, well known benchmark functions used in [8] have been selected (table 1). They are used to test convergence capabilities and search for global minima of optimization algorithms. In all cases, $NP = 500$ and the test were carried out as presented in the same work.

Table 1. Set of Benchmark Functions

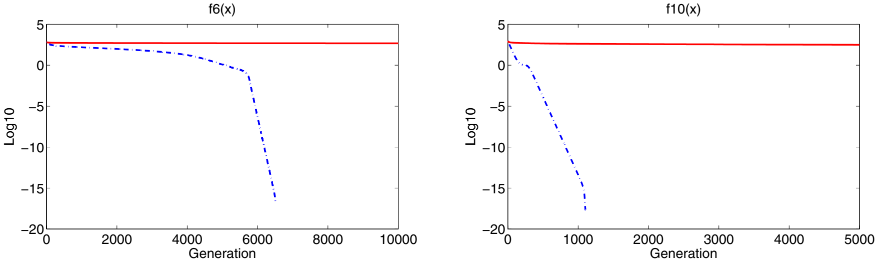
$f_n(x)$	$f_n(x)$	D	Interval
$f_1(x)$	$\sum_{i=1}^{n-1} x_i^2$	50	$x_i \in [-5.12, 5.12]$
$f_2(x)$	$\sum_{i=1}^{n-1} 100 * (x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	50	$x_i \in [-2.048, 2.047]$
$f_6(x)$	$10n + \sum_{i=1}^{n} (x_i^2 - 10 \cos 2\pi x_i)$	50	$x_i \in [-5.12, 5.12]$
$f_{10}(x)$	$\sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos \frac{x_i}{\sqrt{i}} + 1$	50	$x_i \in [-600, 600]$

Table 2. Mean and Variance of Best Fitness Value from 100 independents runs. † The t value of 99 degree of fredom is significant at a 0.05 level of significance by a two-tailed t-test.

$f_n(x)$	DE mean(var)	Proposal mean(var)	FADE mean(var)
$f_1(x)$	9.1351e+1 (5.9625e+1)†	1.7102e-18(2.004e-37)	2.3508 e-10 (2.967e-21)†
$f_2(x)$	2.3381e+3 (6.8718e+4)†	4.5418e+1 (2.797e-1)	4.1566 e+1 (1.82e-2)†
$f_6(x)$	4.6988e+2 (2.0887e+2)†	0(0)	2.5849 e+2 (9.1682e+1)†
$f_{10}(x)$	3.0997e+2 (7.1541e+2)†	0(0)	5.7763 e-1 (3.5e-3)†



(a) $f_1(x)$ (b) $f_2(x)$

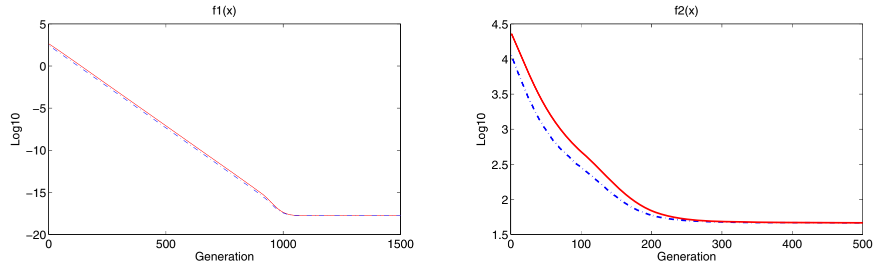


(c) $f_6(x)$ (d) $f_{10}(x)$

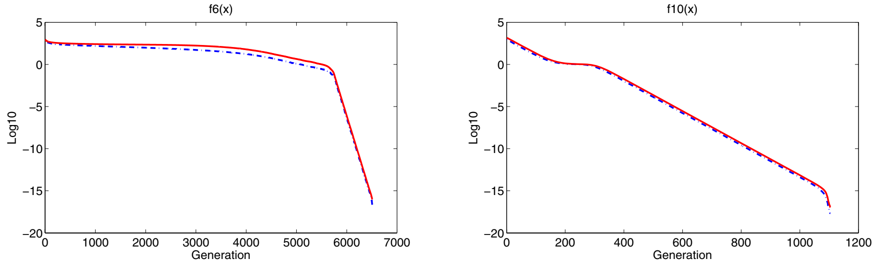
Fig. 3. Benchmark function $f_1(x)$, $f_2(x)$, $f_6(x)$ and $f_{10}(x)$. Fitness average for best individual: DE(-) and Proposal (-.-) .

Function $f_1(x)$ is the simplest test function and it is unimodal, convex and continuous; $f_2(x)$ is commonly known as *Rosenbrock Valley* or *Banana Function*, which optimun is inside a long flat valley; $f_6(x)$ is the Rastrigin's function, based on the sphere model, but adding a cosine modulation ir order to produce many local minima regularly distributed. Finally, test function $f_{10}(x)$ is the Griewangk function, which is multimodal, with many local minima regularly distributed. In all cases, $f_i(x) = 0$ is the minimal value to be reached.

Except one case, ($f_2(x)$) the proposal achieves better mean solution and a narrow variance (table 2 and figure 3) than the results reported by FADE



(a) $f_1(x)$ (b) $f_2(x)$



(c) $f_6(x)$ (d) $f_{10}(x)$

Fig. 4. Benchmark function $f_1(x)$, $f_2(x)$, $f_6(x)$ and $f_{10}(x)$. Fitness average for best individual (-.-) and for population (-) .

algorithm [8]. In all cases it is observed that the average of best individual in each generation is close to population average (figure 4) and with fast convergence to global minimun. This fact reveals the robustness and consistency of the proposed adaptation technique.

6 Conclusion

A proposal for an Adaptive Parameter Control scheme for the Differential Evolution Algorithm has been presented. The adaptation mechanism use fitness evaluation information to control the Scaling Factor and the Crossover Rate of the standard DE algorithm in each member of the population at a given generation. With the exception in one case, simulations studies performed on benchmark functions shows better acurracy than the standard DE and similar algorithms for finding the global minima and avoiding local minima. This fact is supported by the statistical evidence. As next step, this algorithm will be evolved to Self-Adaptive Parameter Control.

Acknowledgments. This work is partially funded by DPI2008-02133/DPI, Ministerio de Ciencia e Innovación, Gobierno de España.

References

1. Storn, R., Price, K.: Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
2. Storn, R.: On the usage of differential evolution for function optimization. 3(2), 519–523 (1996)
3. Storn, R.: Differential Evolution Research - Trends and Open Questions. In: Chakraborty, U.K. (ed.) Advances in Differential Evolution. SCI 143, pp. 1–31. Springer, Heidelberg (2008)
4. Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms 3(2), pp. 124–141 (1999)
5. Zaharie, D.: Critical values for the control parameters of differential evolution algorithms. In: Proceedings of 8th international Mendel conference on soft computing, pp. 62–67 (2002)
6. Zaharie, D.: Parameter adaptation in differential evolution by controlling the population diversity. In: Proceedings of 4th international workshop on symbolic and numeric algorithms for scientific computing, pp. 385–397 (2002)
7. Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm, vol. I, pp. 606–611 (2002)
8. Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm 9(6), 448–462 (2005)
9. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization 2, 1785–1791 (2005)
10. Huang, V.L., Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for constrained real-parameter optimization 2, 324–331 (2006)
11. Huang, V.L., Qin, A.K., Suganthan, P.N., Tasgetiren, M.F.: Multi-objective optimization based on self-adaptive differential evolution algorithm, pp. 324–331 (2007)
12. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, vol. 10(6), pp. 646–657 (2006)
13. Brest, J., Boskovic, B., Greiner, S., Zumer, V., Maucec, M.S.: Performance comparison of self-adaptive and adaptive differential evolution algorithms 11(7), 617–629 (2007)
14. bo Hu, Z., hua Su, Q., wu Xiong, S., gao Hu, F: Self-adaptive hybrid differential evolution with simulated annealing algorithm for numerical optimization, pp. 1189–1194 (2008)

Parallelizing the Design of Radial Basis Function Neural Networks by Means of Evolutionary Meta-algorithms

M.G. Arenas[2], E. Parras-Gutiérrez[1], V.M. Rivas[1], P.A. Castillo[2],
M.J. Del Jesus[1], and J.J. Merelo[2]

[1] Department of Computer Sciences Campus Las Lagunillas s/n, 23071, Jaen, Spain
{eparrasg,vrivas}@vrivas.es
[2] Department of Computer Architecture and Technology
C/Periodista Daniel Saucedo s/n, 18071,CITIC-UGR, Granada, Spain
{maribel,jmerelo,pedro}@atc.ugr.es

Abstract. This work introduces SymbPar, a parallel meta-evolutionary algorithm designed to build Radial Basis Function Networks minimizing the number of parameters needed to be set by hand. Parallelization is implemented using independent agents to evaluate every individual. Experiments over classifications problems show that the new method drastically reduces the time took by sequential algorithms, while maintaining the generalization capabilities and sizes of the nets it builds.[1]

Keywords: Neural networks, evolutionary algorithms, parallelization, meta-evolution.

1 Introduction

Designing Artificial Neural Networks (ANNs) is an important tasks nowadays from the point of view of knowledge extraction algorithms. Optimal design of ANNs can lead to improve their generalization capabilities, to provide an explanation for the outputs (descriptive data mining), and to easily integrate the ANN into hardware components. Radial Basis Function Networks (RBFNs) have shown their ability in classification problems, function approximation, and time series prediction, among others. RBFNs' main property is that once their structure has been fixed, the optimal weights of the links between neurons can be analytically computed. For this reason, researchers have applied data mining techniques looking for the optimal RBFNs that solves a given problem. As any other kind of algorithms, but these techniques also need to be given a set of values for their parameters tuned to every problem they face, thus it would be very helpful a method to find these parameters.

---

[1] This work has been partially supported by the Spanish project MYCYT NoHNES (Ministry of Education and Science - TIN2007-6083), the CICYT Spanish TIN2005-08386-C05-03 project, and the Junta de Andalucía CICE projects PC06-TIC-02025 and TIC-3928

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 383–390, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The EvRBF [14] method was an evolutionary algorithm developed to automatically design asymmetric RBFNs. Used to build RBFNs that solved various kind of problems, its main disadvantage was that needed to be given a large number of parameters that changed for every problem. In order to automatically assess the values for the parameters, Meta_CHC_RBF [12] was developed. As the name suggest, it was a Meta-Evolutionary Algorithm, able to establish many different values for parameters of EvRBF, making them to evolve towards the optimal ones. Based on EvRBF and Meta_CHC_RBF, in this paper the method SymbPar, a parallel meta-algorithm, is proposed. Thus, the main goal of SymbPar is to automatically design RBF neural networks reducing the computation cost. The method tries to find an optimal configuration of parameters for the EvRBF method, adapted automatically to every problem.

The remaining is organized as follows: section 2 describes the state of the art, section 3 introduces the method SymbPar, section 4 shows the experimentation process and results, and finally, section 5 describes conclusions and future works.

2 State of the Art

Since Mercer and Sampson made the first approach to calibrate parameters of a genetic algorithm [11], there have been a lot of attempts to develop methods in order to give support for parameter tuning. Nevertheless, their research had two principal problems, the big computation cost and the limited search space. After that, Greffenstette proposed what is known as “meta genetic algorithm” [6], where the algorithm whose parameters are calibrated does not have to be changed for the application of a meta-algorithm, the search space was composed of a widely parameter combination, and it gave as result a better optimization. On the other hand, Greffenstette’s meta-GA had three main limitations: was necessary to choose a particular parameterized subclass of the GA to explore, the GA considered were essentially unconstrained optimization procedures, and the metalevel experiments represent a sizable number of CPU hours.

This method was followed by others researches in the same way. Shahookar and Mazumder [16] used a meta-GA approach in order to solve a cell placement problem for industrial circuits, which resulted in an important decrease of the number of configurations which had to be evaluated.

Subsequent research apply this approach to a wide variety of task. For example, a model-based on meta-GA for the interpretation of cardiac beats by evolutionary algorithms [7]; or a meta-genetic algorithm for times series forecasting [5], where a two-level architecture was proposed: a meta-level GA, used for the model selection task, and a low-level GA, with parameter estimation purposes.

Laoutaris et al. [10] faced the problem of hierarchical web caches; thus, in order to investigate caching a local copy in all intermediate caches on the reverse path, employed meta-algorithms to operate on groups of caches organized in hierarchies, whereas earlier work on self-organizing linear search studied the operation of meta algorithms on isolated linear lists. Rex et al. [13] implemented the Brain Extraction Meta-Algorithm (BEMA) in order to study the accurate identification of brain tissue and cerebrospinal fluid in a whole-head Magnetic Resonance Image (a critical task in many neuroimaging studies). Automating this procedure can eliminate intra and interrater variance and greatly increase throughput for a labor-intensive step. For this, they developed a meta-algorithm which uses four freely available brain extraction algorithms and a linear volume registration procedure, concluding that the method was able to produce results that were of higher accuracy and increased robustness when compared to any of the brain extraction algorithms used in its processing.

More recently, Samsonovich and De Jong [15] try to calibrate genetic algorithms applied to three different problems: 2D optimization, the eight puzzle, and binary dentritic tree construction problems. They concluded that the general idea of meta-evolution may improve the performance of an evolutionary algorithm if: first, it is understood as co-evolution of reproductive operators and, second, the *fitness* is properly defined at the meta-level(s).

From the point of view of parallelization, ANNs' design by means parallel algorithms was studied by Castillo et al. [4]. In their work, they proposed a parallel version of G-Prop [3] where several machines worked in parallel searching for the solution, interchanging (sending/receiving) some population individuals after few generations. This model uses a ring migration scheme: sending some individuals to the next island, and getting the best individuals from the previous island (being every island a computer). In this parallel system the population is distributed between several evolutionary algorithm (executed in several processors), and any of these algorithms conforms to the sequential model of GProp.

3 Method Overview

SymbPar is a parallel evolutionary algorithm, whose main goal is to automatically design the RBFNs, by means of finding a suitable set of parameters for the EvRBF.

Parallelizing an algorithm typically involves finding out the critical points of the sequential evolutionary algorithm in order to success the operation. There are two important properties of sequential genetic algorithms that determine their running time. First one is the population size; usually, this parameter is high in order to get a suitable diversity, so the running time is larger than using a few individuals. Second one is the evaluation function used to set the *fitness* of the individuals; evaluation functions tends to be the bottleneck of evolutionary algorithms. Therefore, sequential evolutionary algorithms sometimes need a huge amount of memory, because uses a big populations. Both the requirement of memory and computation time are two good reasons to attempt to speed up the execution of an evolutionary algorithm by means of parallelization. In our case, the sequential version of the algorithm is able to achieve good solutions with small populations (50 individuals), so the parallelization tries to minimize the total execution time. Candidate operations to be parallelized are those that can be independently applied.The most commonly chosen operation is the individual evaluation [17], since it is the most time consuming operation of the algorithm.

In order to distribute the computation load following the *Master-Worker* scheme [2], SymbPar has two important classes: *Master* class, whose function is to organize the work to evaluate all the individuals, to create the independent tasks that make this evaluation and to collect the results once the tasks have ended; and *Worker* class, whose function is to carry out the tasks commanded by *Master*, and to send the obtained result. Then, the algorithm SymbPar creates a type *Master* instance for every generation, just before executing the evaluation of each individual of the population. The *Master* object is an independent agent that provokes the stop of the algorithm, and creates as many objects of the type *Worker* as necessary to evaluate every individual. Figure 1(b) shows the execution scheme of every generation.

In this paper, a study to determine the type of parallelization and distribution of tasks has to be done. The Master class assigns and distributes tasks, using a program running on Java virtual machine called DRM [8]. DRM distributes tasks between the set of available machines in a completely random way. Such distribution is quick to implement but sets a not completely balanced load. However, it reduces programming effort as the user might have no knowledge about the available machines. DRM distributes task code to be run through the network, using the Java's serialization of code and data.

4 Experiments and Results

In order to test SymbPar performance 11 different problems have been evaluated with the following data sets: Flag, German, Glass, Haberman, Ionosphere, Newthyroid, Pima, Postoperative, Sonar, Vehicle, and WDBC from UCI data set repository[2]. The problems characteristics and size can be seen on table 1.

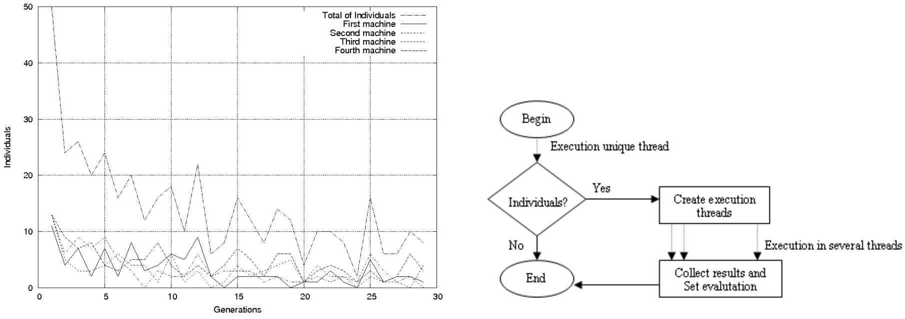
A 10-crossfold validation method has been used for every data set, so that every one has been divided into 10 different sets of training-test patterns. For every training-set couple, the algorithms have been executed three times, and the results show the average over these 30 executions (3 times * 10 crossfold). So, since we have 11 data sets, a total of 330 executions have been performed using 1, 2 and 4 computations machines, that is, 990 executions. All experiments have been executed on virtual computers of the Scientific Computer Center of Andalucía[3] using four machines with four cores any of them with Ubuntu 7.10 operating system (Linux 2.6.22-14-generic SMP Kernel).

Figure 1(a) shows an example of distribution using four machines. It can be seen how in those generations where there are fewer individuals to be evaluated, some computing machines are idle. Those machines have no task assigned, and that causes a waste of resources and a loss in speedup. However, in those generations where there are many individuals to be evaluated, distribution tends to be fair. This can be seen in the first generation, where the number of individuals to be evaluated is equal to 50 and available machines show a similar computing load (lines showing machines load exhibit approximately the same value).

---

[2] http://www.ics.uci.edu/∼mlearn/MLRepository.html

[3] https://eciencia.cica.es/



a b

Fig. 1. (a)Distributions picture shows task distributions along four computation machines for Vehicle problem. (b)Execution scheme for one generation.

Results yielded by both Meta_CHC_RBF and SymbPar using a single machine can be seen in table 2. The table shows for both algorithms and from left to right: the number of neurons (Neurons) (cols 2 and 5), the percent of correct classified patterns (Test) (cols 3 and 6), and the computation time (minutes) (cols 4 and 7). All of them show the average and standard deviation. Results show that small and “easy to learn”(Postoperative, Glass and Flag) problems are solved using few minutes, so no advantage of the parallelization is obtained. Nevertheless, the parallelization is advantageous for all the remaining problems.

Results for execution time show a high dependency on the size of the databases being classified. For this reason, tables 1 (ordered by the size of databases) and 2 (ordered by the time took to learn every database) show the databases in a relatively similar order. The explanation comes from the fact that the hardest task performed by the algorithm is the network training, and this task mainly depends on the number and size of patterns. Nevertheless, the lasting of the training task also depends on the size of the nets, but this size is not directly

Table 1. Number of features, instances, and size (product of features and instances)

Database	Features	Instances	Size
Postoperative	9	90	810
Haberman	4	306	1224
New-thyroid	6	215	1290
Glass	10	214	2140
Flag	29	194	5626
Pima	9	768	6912
Ionosphere	35	351	12285
Sonar	61	208	12688
Vehicle	19	846	16074
WDBC	31	570	17670
German	21	1000	21000

Table 2. Results of Meta_CHC_RBF and SymbPar. Results show that small problems are solved using few minutes, so no advantage of the parallelization is obtained. Nevertheless, the parallelization is advantageous for all the remaining problems.

Database		Meta_CHC_RBF			SymbPar	
	Neurons	Test(%)	T(min)	Neurons	Test(%)	T(min)
Postoperative	02 ± 01	82.96 ± 6.99	**05** ± 01	02 ± 02	82.23 ± 8.56	10 ± 0
Glass	01 ± 01	92.22 ± 2.19	**11** ± 02	01 ± 00	92.08 ± 2.11	**11** ± 0
Flag	01 ± 00	87.39 ± 15.56	**10** ± 02	02 ± 01	87.39 ± 16.15	11 ± 0
Haberman	07 ± 03	82.55 ± 5.28	57 ± 12	06 ± 03	81.66 ± 4.67	**17** ± 1
Sonar	06 ± 04	71.79 ± 6.66	130 ± 32	06 ± 04	70.99 ± 8.86	**18** ± 2
New-thyroid	07 ± 03	99.22 ± 1.78	60 ± 08	06 ± 03	98.44 ± 2.57	**20** ± 2
Ionosphere	13 ± 07	97.64 ± 2.47	316 ± 82	12 ± 06	97.46 ± 2.82	**33** ± 4
WDBC	08 ± 06	95.02 ± 3.29	310 ± 55	09 ± 07	95.27 ± 3.36	**36** ± 3
Pima	09 ± 06	80.06 ± 3.24	309 ± 53	16 ± 11	80.99 ± 2.90	**49** ± 5
German	06 ± 05	73.20 ± 2.43	259 ± 36	08 ± 07	73.47 ± 2.39	**50** ± 7
Vehicle	49 ± 20	94.99 ± 1.80	1903 ± 442	44 ± 21	93.77 ± 2.24	**119** ± 16

related to the size of the database. This is probably due to the information quantity provided by every attribute in the database. Thus, German database, even being bigger than Vehicle database, is easier to classify since their attributes are able to define disjoint hyper-volumes in the input space, every one including patterns that belong to the same class. As the evolutionary algorithm locates neurons close to the center of those hyper-volumes, the classification ability increases without the need to increase the number of neurons.

Table 3 shows SymbPar results for distributed evaluations in 2 and 4 machines. The table shows from left to right: the number of neurons (Neurons) (cols 4-5), the percent of correct classified patterns (Test) (cols 2-3), the computation time (minutes) (cols 6-7), the speedup the algorithm (8-9) and the serial fraction of the code (cols 10-11). All of them show the average and standard deviation and all data are collected using 2 and 4 running machines. The results for neurons and test demonstrate that the quality of the solutions found when more machines are used are not worse the solutions found by Meta_CHC_RBF and SymbPar using one machine.

In order to measure the performance, speedup and serial fraction [9] have been reported. Even if a speedup value is small, result can be considered good if serial fraction remains constant for different values of the number of processors, since the loss of efficiency is due to the limited parallelism of the algorithm. On the other side, smooth increments of serial fraction indicates that the granularity of the parallel tasks is too fine [1].

Taking into account the running time and speedup, no improvement is obtained in the case of problems easily solved (see results for Postoperative, Glass and Flag problems). For simple problems, as the number of machines increases, (table 3, lines 1-3) the running time also increases due to the communications between machines. Parallelization in these cases makes no sense. In more complex problems (bottom of the table 3, Vehicle or German problems), speedup is obtained when increasing the number of machines. In these problems, speedup increases while serial fraction of the problems, (cols 9 and 10 of table 3) are constant. Thus, we have limitations due to the fraction of the code that is executed

Table 3. Results of SymbPar for 2 and 4 machines with test, number of neurons in RBF network (neurons), execution time speedup and serial fraction (f) for two and four machines. In complex problems (bottom), speedup is obtained when increasing the number of machines while serial fraction of the problems, (cols 9 and 10) is constant.

Database	Test(%)		Neurons		Running Time (min)		Speedup		Ser. F	
Machines	2	4	2	4	2	4	2	4	2	4
Postoperative	82.22 ± 6.9	82.22 ± 8.0	2.0 ± 1.5	1.9 ± 0.9	9.4 ± 0.5	10.0 ± 0.0	1.04	0.98	0.92	1.03
Glass	92.01 ± 2.1	92.08 ± 2.1	1.0 ± 0.0	1.1 ± 0.4	11.3 ± 0.5	11.5 ± 0.5	0.97	0.96	1.06	1.06
Flag	86.58 ± 15.5	87.04 ± 15.0	1.8 ± 1.4	1.6 ± 1.2	11.6 ± 0.4	12.0 ± 0.3	0.94	0.92	1.13	1.12
Haberman	82.09 ± 5.3	81.80 ± 5.1	6.4 ± 2.8	6.2 ± 2.6	15.5 ± 0.8	15.3 ± 0.8	1.06	1.08	0.89	0.9
Sonar	73.84 ± 7.4	72.10 ± 6.8	6.6 ± 4.4	7.7 ± 6.5	18.6 ± 1.8	17.6 ± 1.3	1.19	1.26	0.68	0.72
NewThyroid	98.17 ± 3.9	98.90 ± 2.0	5.9 ± 3.0	6.4 ± 3.8	17.7 ± 1.1	16.8 ± 0.8	1.13	1.18	0.77	0.80
Ionosphere	97.65 ± 2.8	96.98 ± 3.0	12.2 ± 6.3	13.7 ± 6.8	32.8 ± 4.4	29.0 ± 28.0	1.21	1.37	0.65	0.64
WDBC	94.91 ± 3.7	95.20 ± 3.5	6.5 ± 3.2	7.7 ± 7.5	35.3 ± 3.0	31.6 ± 2.8	1.17	1.31	0.71	0.68
Pima	80.39 ± 3.0	80.99 ± 2.6	12.0 ± 8.2	12.2 ± 7.0	38.7 ± 3.2	35.4 ± 2.7	1.26	1.37	0.59	0.64
German	73.47 ± 2.2	73.10 ± 3.4	9.2 ± 6.8	9.7 ± 8.7	42.0 ± 3.6	38.2 ± 3.3	1.20	1.32	0.67	0.68
Vehicle	93.85 ± 2.1	94.17 ± 2.3	41.1 ± 17.5	43.5 ± 20.5	97.7 ± 10.7	85.6 ± 8.7	1.22	1.39	0.64	0.63

in sequential way. For example, if we see line for Vehicle problem the speedup is 1.22 and 1.39 that is less we expect. But the serial fraction, is constant for these experiments (0, 64 for 2 machines and 0, 63 for 4 machines) that is, we can not get for speedup using this kind of parallelization. We need to reduce serial fraction value in order to get a better results for speedup.

5 Conclusions and Future Research

Comparison between Meta_CHC_RBF and Symbpar proposed in this paper, shows an improvement in running time. It allows to face complex problems in the future. In this paper the simplest parallelization method allows a parallel neural networks design. A random distribution of evaluations among the available machines has been used. Due to this fact, some machines might remain idle for some generations, and that might reduce the theoretical speedup. However, even with this limitation, by increasing the problem complexity, speedup increases and serial fraction is constant for several machines. This suggests that in more difficult problems, parallelization could improve results obtained. In future works, a fair distribution of assessments among available machines will be carried out, and more difficult problems will be faced.

References

1. Alba, E., Luque, G.: Evaluation of Parallel Metaheuristics. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 9–14. Springer, Heidelberg (2006)
2. Bethke, A.D.: Comparison of genetic algorithms and gradient-based optimizers on parallel processors: Efficiency of use of processing capacity. Tech. Rep., University of Michigan, Ann Arbor, Logic of Computers Group (1976)
3. Castillo, P.A., Merelo, J.J., Rivas, V., Romero, G., Prieto, A.: G-Prop: Global Optimization of Multilayer Perceptrons using GAs. Neurocomputing 35/1-4, 149–163 (2000)

4. Castillo, P.A., Arenas, M.G., Merelo, J.J., Rivas, V., Romero, G.: Optimisation of Multilayer Perceptrons Using a Distributed Evolutionary Algorithm with SOAP. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 676–685. Springer, Heidelberg (2002)
5. Cortez, P., Rocha, M., Neves, J.: A meta-genetic algorithm for time series forecasting. In: Torgo, L. (ed.) Proceedings of Workshop on Artificial Intelligence Techniques for Financial Time Series Analysis (AIFTSA 2001), 10th Portuguese Conference on Artificial Intelligence (EPIA 2001), Oporto, Portugal, pp. 21–31 (2001)
6. Grefenstette, J.: Optimization of control parameters for genetic algorithms. IEEE Trans. Syst. Man Cybern 16(1), 122–128 (1986)
7. Hernandez, A.I., Carrault, G., Mora, F., Bardou, A.: Model-based interpretation of cardiac beats by evolutionary algorithms: signal and model interaction. Artificial Intelligence in Medicine 26(3), 211–235 (2002)
8. Jelasity, M., Preub, M., Paechter, B.: A scalable and robust framework for distributed application. In Proc. on Evolutionary Computation, pp. 1540–1545 (2002).
9. Karp, A.H., Flatt, H.P.: Measuring parallel processor performance. Communications of the ACM 33(5), 539–543 (1990)
10. Laoutaris, N., Syntila, S., Stavrakakis, I.: Meta algorithms for Hierarchical Web Caches. In: IEEE International Conference on Performance, Computing, and Communications, pp. 445–452 (2004) ISBN: 0-7803-8396-6
11. Mercer, R.E., Sampson, J.R.: Adaptive search using a reproductive meta-plan. Kybernetes 7(3), 215–228 (1978)
12. Parras-Gutiérrez, E., Rivas, V.M., Merelo Juan, J., del Jesus, M.J.: Study of the robustness of a meta-algorithm for the estimation of parameters in Artificial Neural Networks design. In: HIS 2008: 8th International Conference on Hybrid Intelligent Systems, Barcelona, pp. 519–524. IEEE computer society, Los Alamitos (2008)
13. Rex, D.E., Shattuck, D.W., Woods, R.P., Narr, K.L., Luders, E., Rehm, K., Stolzner, S.E., Rottenberg, D.A., Toga, A.W.: A meta-algorithm for brain extraction in MRI. NeuroImage 23(2), 625–637 (2004)
14. Rivas, V.M., Merelo, J.J., Castillo, P.A., Arenas, M.G., Castellanos, J.G.: Evolving RBF neural networks for time-series forecasting with EvRBF. Information Sciences 165(3-4), 207–220 (2004)
15. Samsonovich, A.V., De Jong, K.A.: Pricing the 'free lunch' of meta-evolution. In: Beyer, H.-G., O'Reilly, U.-M. (eds.) GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 1355–1362. ACM Press, New York (2005)
16. Shahookar, K., Mazumder, P.: VLSI cell placement techniques. ACM Comput. Surv. 23(2), 143–220 (1991)
17. Tomassini, M.: Parallel and distributed evolutionary algorithms: A review. In: Miettinen, K., et al. (eds.) Evolutionary Algorithms in Engineering and Computer Science, pp. 113–133. J. Wiley and Sons, Chichester (1999)

A Genetic Algorithm for ANN Design, Training and Simplification

Daniel Rivero, Julian Dorado, Enrique Fernández-Blanco, and Alejandro Pazos

Department of Information Technologies and Communications
`{drivero,julian,efernandez,apazos}@udc.es`

Abstract. This paper proposes a new evolutionary method for generating ANNs. In this method, a simple real-number string is used to codify both architecture and weights of the networks. Therefore, a simple GA can be used to evolve ANNs. One of the most interesting features of the technique presented here is that the networks obtained have been optimised, and they have a low number of neurons and connections. This technique has been applied to solve one of the most used benchmark problems, and results show that this technique can obtain better results than other automatic ANN development techniques.

1 Introduction

Artificial Neural Networks (ANNs) are well-known systems which have already been used in many different applications [1]. They have been successfully applied in areas such as image processing, data mining, etc.

The development of ANNs is a process that can be roughly divided into two parts: development of the architecture and its corresponding training and validation. The development of the architecture refers to the process of determining how many neurons an ANN will have, in how many layers they will be structured, and how they will be interconnected. The training refers to calculating the weight values of this architecture's connections.

Given that the network architecture depends on the problem to be resolved, generally the design process of this architecture is the result of a manual process based on experience. In other words, the expert has to experiment with various different architectures until he finds one that provides good results after the training process. Therefore, the expert has to make several experiments with different architectures and train each one of them in order to be able to determine which one will be the best. This is a slow and manual process, even though some recent ANN creation techniques have been developed in a more or less automated way [2]. Some of the most recent techniques use Genetic Programming (GP) to develop ANNs [3] due to its higher capability of representation based on the use of trees as codification method. However, it remains very difficult to represent the extremely complex structure and high connectivity of ANNs by means of trees.

This paper proposes a novel approach in which a Genetic Algorithm (GA) is used to develop ANNs. Both architecture development and training are performed by the GA, so the expert does not have to do any effort to obtain the ANNs.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 391–398, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The model proposed in this paper has been applied to solve a well-known benchmark problem. This problem has already been solved with ANNs developed with other techniques, so it provides a basis to perform a comparison of this method with other existing ones.

2 State of the Art

2.1 Genetic Algorithms

A GA [4] is a search technique inspired in the world of Biology. More specifically, the Evolution Theory by Charles Darwin [5] is taken as basis for its working. GAs are used to solve optimization problems by copying the evolutionary behaviour of species. From an initial random population of solutions, this population is evolved by means of selection, mutation and crossover operators, inspired in natural evolution. By applying this set of operations, the population goes through an iterative process in which it reaches different states, each one is called generation. As a result of this process, the population is expected to reach a generation in which it contains a good solution to the problem. In GAs the solutions of the problem are codified as a string of bits or real numbers.

2.2 Development of ANNs with Evolutionary Computation

The development of ANNs is a topic that has been extensively dealt with very diverse techniques. The world of evolutionary algorithms is no exception, and proof of that is the great amount of works that have been published about the different techniques in this area, even with GAs or GP [2]. As a general rule, the field of ANNs generation using evolutionary algorithms is divided into three main fields: evolution of weights, architectures and learning rules.

First, the weight evolution starts from an ANN with an already determined topology. the problem to be solved is the training of the connection weights, attempting to minimize the network error. With the use of an evolutionary algorithm, the weights can be represented either as the concatenation of binary or real values [6].

Second, the evolution of architectures includes the generation of the topological structure. In order to use evolutionary algorithms to develop ANN architectures, it is needed to choose how to encode the genotype of a given network for it to be used by the genetic operators.

At the first option, direct encoding, there is a one-to-one correspondence between every one of the genes and their subsequent phenotypes. The most typical encoding method consists of a matrix that represents an architecture where every element reveals the presence or absence of connection between two nodes [7].

Inside the encoding schemes, GP has been used to develop both architecture and connection weights at the same time, either for feed-forward [3] or recurrent ANNs [4], with no limitations in their architecture. This new codification scheme also allows the obtaining of simple networks, with a low number of neurons and connections, and the results published are very promising.

Apart from direct encoding, there are some indirect encoding methods. In these methods, only some characteristics of the architecture are encoded in the chromosome. These methods have several types of representation. Firstly, the parametric representations represent the network as a group of parameters such as number of hidden layers, number of nodes for each layer, number of connections between two layers, etc [9]. Another non direct representation type is based on grammatical rules [8]. In this system, the network is represented by a group of rules, shaped as production rules that make a matrix that represents the network.

With regard to the evolution of the learning rule, there are several approaches [10], although most of them are only based on how learning can modify or guide the evolution and also on the relationship among the architecture and the connection weights.

3 Model

The model proposed in this paper allows the obtaining of ANNs with no limitation of architecture. Traditional ANNs are divided into several layers: one input layer, one or more hidden layers, and one output layer. Neurons are usually totally connected from one layer to the following, but not between neurons in the same layer.

However, the ANNs generated by this model do not have that limitation. In fact, any possible connectivity is allowed. The only restriction used in this paper is that no recurrent connections are allowed. With this restriction, the largest network that can be generated with N hidden neurons corresponds with a generalized multilayer perceptron (GMLP) [11]. In this kind of ANN, the ith node has connections from every jth node, with $j<i$, and i is not an input node. Fig. 1 shows the typical topology of a GMLP.

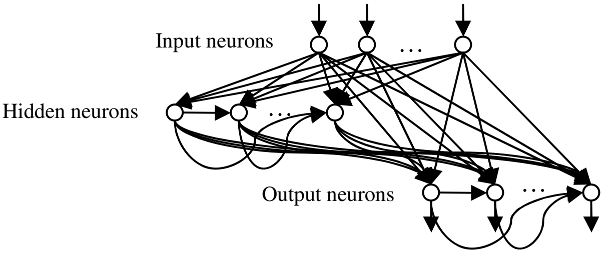


Fig. 1. Architecture of the GMLP

GMLP networks have a processing capacity similar to a MLP with a lower number of neurons and connections [12]. As these networks have a lower number of nodes and connections, they also show a better generalisation.

In order to evolve these kind of networks, it is necessary to define how a GA can codify them inside a string. This string represents a chromosome that codifies the ANN. This string can be divided into three different functional parts:

- Connections. A network with M inputs, N hidden neurons, and P outputs, has the number of connections necessary to connect the inputs to the outputs, inputs to the hidden nodes, hidden nodes to outputs and hidden nodes between them (not allowing recurrent connections). As the connectivity of the network is being evolved, in the chromosome each connection has associated a boolean value. A true value implies that the connection is present in the network, whereas a false value means that that connection has to be deleted. Next to that value, the chromosome contains the weight value of that connection.

- Bias. As happens with the weights of the connections, the biases of the neurons also have to be present in the chromosome. In this case, only N+P real values are necessary (bias for the hidden and output neurons).
- Neurons. Finally, in order to truly design the architecture of the networks, N boolean values are included in the chromosome. These values represent if a particular hidden neuron is present or not in the chromosome.

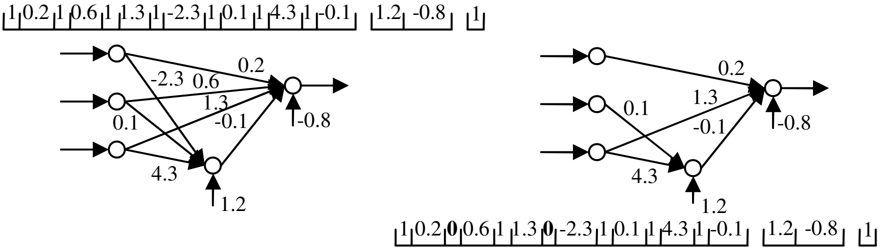


Fig. 2. Codification of two different ANNs

Fig. 2 shows two examples of two different chromosomes and the networks that they represent, given that they can only contain 1 hidden neuron. Both chromosomes are divided in their three components. The first chromosome uses all of the connections. The second chromosome is equal to the first except for two Boolean values associated to two connections. These two values have a 0 value in this second chromosome. This means that the network codified (and shown in the figure) does not have the corresponding connections.

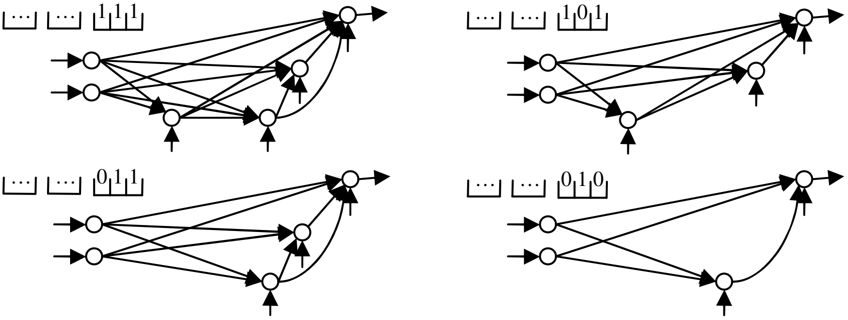


Fig. 3. Codification of four different ANNs

Fig. 3 shows other 4 chromosomes and their corresponding networks. In this case, the maximum number of hidden neurons is 4. In this figure, the "connection" and "bias" part of the chromosome are not shown, to simplify the figure, and it is supposed that all of the weights are used. Note that in three of the 4 networks some neurons are not used. Even if a connection is used in the "connection" part of the chromosome, if its neuron (whether input or output) is not used, it is deleted anyway.

These 2 figures show that, with this configuration, any possible feed-forward architecture can be evolved. The only limitation to the architectures developed by this model is in the number of hidden neurons that can appear in the networks. Therefore, this maximum number of hidden neurons has to be set large enough to allow the problem to be solved.

Once a chromosome has been decodified, an ANN is generated. However, that network may have some parts which can not be used. This can happen if a hidden neuron does not have any input our output connection. If a hidden neuron does not have input connections, its output will be always a constant. In this case, it does not contribute to the solution of the problem. On the other side, if a neuron does not have any output connections, its output is not used, and therefore that neuron is useless. In both cases, these neurons (and their corresponding input and output connections) are deleted from the network.

As a consequence of this deletion, there may be other neurons that become useless. Therefore, the network examination/neuron deletion becomes an iterative process which is performed until no useless neuron is found.

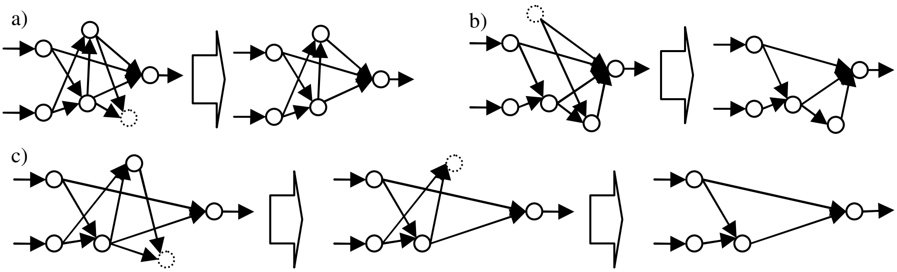


Fig. 4. Simplification of the networks

Fig. 4 shows three different examples of these situations. In the networks shown in this figure, all of the neurons that will be deleted have been drawn in dotted line. In network a) a neuron is found without output connections. In network b) a neuron is found with no input connections. In both cases, those neurons are deleted and the resulting networks do not have any other useless part.

In network c) a neuron is found with no output connections. That neuron is deleted. However, after this deletion the new network has another neuron that has to be deleted. This process goes on until the network does not have useless parts.

This simplification task is performed each time a genotype is evaluated and a network generated from it. Even a network undergoes a simplification process, its corresponding genotype is not modified to reflect those changes in the networks, because this would lead to losing many genetic material.

Once the tree has been evaluated, the genotype turns into phenotype. In other words, it is converted into an ANN with its weights already set (thus it does not need to be trained) and therefore it can be evaluated. The evolutionary process needs the assignation of a fitness value to every genotype. This value is the error of the evaluation of the network with the training pattern set.

Nevertheless, this error value considered as fitness value has been modified in order to induce the system to generate simple networks. The modification has been made by adding a penalization value multiplied by the number of connections of the network, because this number is the determining factor of the network's complexity. In such way, and given that the evolutionary system has been designed in order to minimize an error value, when adding a fitness value, a larger network would have a worse fitness value. Therefore, the existence of simple networks would be preferred as the penalization value that is added is proportional to the number of connections at the ANN. The calculus of the final fitness is as follows:

$$fitness = MSE + N * P \tag{1}$$

where MSE is the mean square error of the ANN within the group of training patterns, N is the number of connections of the network and P is the penalization value for such number of connections.

4 Results

This system has been used for the comparison with other methods, which results have already been published [2]. In order to do the comparison, a well-known benchmark problem has been used, taken from the UCI [13]. This problem is the breast cancer problem. This is a classification problem in which, from 9 different real-value attributes, the type on breast cancer has to be determined (malignant/benign). The data was taken from different patients in a Wisconsin Health Center.

The cross-validation method has been used for the comparison of different evolutionary methods-based techniques used in ANN generation [2]. The results of this work are the average accuracies obtained in the 10 test results generated by the five iterations of the 5-fold cross-validation (5x2cv) method. Such values are the basis for the comparison of the described technique with other well-known ones.

The algorithms compared with this technique are widely explained with detail in a previous work [2]. Such work shows the average times needed to achieve the results. Not having the same processor that was used, the computational effort needed for achieving the results can be estimated. This effort represents the number of times that the pattern file was evaluated. The computational effort for every technique can be measured using the population size, the number of generations, the number of BP epochs, etc. This calculation varies for every algorithm used.

The techniques with which this work is compared refer to the use of evolutionary algorithms to design neural networks. These techniques are connectivity matrix, pruning, parameter search and graph-rewriting grammar.

The results obtained with these 4 methods, in comparison with the method described in this work, can be seen in Table 1. In order to correctly compare the results, the computational effort needed by each of these techniques was calculated.

Table 1. Comparison of this technique with other

	Effort	Accuracy	Std.	Accuracy obtained here	Std. obtained here	Num. neurons obtained here	Num. connections obtained here
Matrix	92000	96.77 %	1.10	**96.79 %**	0.56	4.26	37.40
Pruning	4620	**96.31 %**	1.21	95.68 %	0.57	3.34	26.00
Parameters	100000	96.69 %	1.13	**96.79 %**	0.56	4.30	38.08
Grammar	300000	96.71 %	1.16	**96.79 %**	0.61	4.36	43.08

The system described here was then executed until that computational effort was reached, and then it was stopped. The accuracy value shown in that table with the technique described here corresponds to the accuracy obtained in that computational effort. Finally, that table also shows the average number of neurons and connections of the networks obtained in this method, in that computational effort.

As can be seen in this table, results obtained in this particular problem are higher in comparison with 3 of the 4 techniques. Only in one case the method proposed here achieved worse results than another technique. This is the case of the comparison with the pruning technique. The reason of this worse result is the low number of fitness function executions used by the pruning technique. However, as can be seen on the table, with more fitness function executions, the accuracy of the results obtained by the method described here are higher than those of the pruning method. In fact, the maximum test accuracy obtained was 96.84 %.

5 Conclusions

This paper presents a new technique for developing ANNs without any participation of the expert. To accomplish this, a GA is used to evolve different networks. The results obtained with one sample problem show that this technique can improve the accuracies obtained with other automatic ANNs generating techniques. However, even these results seem promising, more experiments have to be done in order to measure the real performance of the system.

An important feature of the technique proposed in this paper is that the ANNs generated undergo an optimization process. Due to having a penalisation to the number of connections of the network, the evolutionary process tries to evolve networks with a low number of connections, and therefore simple networks are found.

Acknowledgements

This work was partially supported by the Spanish Ministry of Science and Innovation (Ref TIN2006-13274) and the European Regional Development Funds (ERDF), grant (Ref. PIO52048 and RD07/0067/0005) funded by the Carlos III Health Institute, grant (Ref. PGDIT 07TMT011CT) and (Ref. PGDIT08SIN010105PR) from the General Directorate of Research, Development and Innovation of the Xunta de Galicia, grant (2007/127 and 2007/144) from the General Directorate of Scientific and Technologic

Promotion of the Galician University System of the Xunta de Galicia, and grants (AP2005-1415 and BES-2006-13535) from the Spanish Ministry of Education and Science.

The development of the particular experiments in this paper were made with thanks to the support of the "Centro de Supercomputación de Galicia (CESGA)".

The Cleveland heart disease database was available thanks to Robert Detrano, M.D., Ph.D., V.A. Medical Center, Long Beach and Cleveland Clinic Foundation.

References

1. Rabuñal, J.R., Dorado, J. (eds.): Artificial Neural Networks in Real-Life Applications. Idea Group Inc. (2005)
2. Cantú-Paz, E., Kamath, C.: An Empirical Comparison of Combinations of Evolutionary Algorithms and Neural Networks for Classification Problems. IEEE Transactions on systems, Man and Cybernetics Part B: Cybernetics 35, 915–927 (2005)
3. Rivero, D., Dorado, J., Rabuñal, J.R., Pazos, A.: Modifying genetic programming for artificial neural network development for data mining. Soft Computing - A Fusion of Foundations, Methodologies and Applications 13(3), 291–305 (2008)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
5. Darwin, C.: On the Origin of Species by means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life, 6th edn. Cambridge University Press, Cambridge (1864); originally published in 1859
6. Greenwood, G.W.: Training partially recurrent neural networks using evolutionary strategies. IEEE Trans. Speech Audio Processing 5, 192–194 (1997)
7. Alba, E., Aldana, J.F., Troya, J.M.: Fully automatic ANN design: A genetic approach. In: Mira, J., Cabestany, J., Prieto, A.G. (eds.) IWANN 1993. LNCS, vol. 686, pp. 399–404. Springer, Heidelberg (1993)
8. Kitano, H.: Designing neural networks using genetic algorithms with graph generation system. Complex Systems 4, 461–476 (1990)
9. Harp, S.A., Samad, T., Guha, A.: Toward the genetic synthesis of neural networks. In: Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications, pp. 360–369. Morgan Kaufmann, San Mateo (1989)
10. Turney, P., Whitley, D., Anderson, R.: Special issue on the baldwinian effect. Evolutionary Computation 4(3), 213–329 (1996)
11. Werbos, P.J.: The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting. Wiley, New York (1994)
12. Garcia-Pedrajas, N., Ortiz-Boyer, D., Hervas-Martinez, C.: Cooperative coevolution of generalized multi-layer perceptrons. Neurocomputing 56, 257–283 (2004)
13. Mertz, C.J., Murphy, P.M.: UCI repository of machine learning databases (2002), `http://www-old.ics.uci.edu/pub/machine-learning-databases`

Graph-Based Representations in Pattern Recognition and Computational Intelligence

R. Marfil[1], F. Escolano[2], and A. Bandera[1]

[1]Grupo ISIS, Dpto. Tecnología Electrónica, Universidad de Málaga (Spain)
`rebeca@uma.es, ajbandera@uma.es`
[2] Robot Vision Group, Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante (Spain)
`sco@dccia.ua.es`

Abstract. Graph theory, which used to be a purely academic discipline, is now increasingly becoming an essential part in different areas of research. This paper briefly present new perspectives in graph–based representations applied in emerging fields, such as computer vision and image processing, robotics, network analysis, web mining, chemistry, bioinformatics, sensor networks, biomedical engineering or evolutionary computation.

1 Introduction

The explosive growth of internet technology in the past decade and the recent proliferation of scientific data sets have generated a gigantic volume of information. These databases provide to the global community excellent opportunities for acquiring and disseminatting new data. This has provoked that todays pattern recognition tasks are less and less concerned with the analysis of a single pattern, as patterns are rather grouped in huge databases eventually disseminated over the internet. For instance, the Human Genome Project (HGP) has accumulated vast amounts of genomic data. When these repositories run very autonomously, the ways data are organized and stored are very diversified and bear little coordination. Furthermore, the patterns that need to be modeled are usually complex in nature [1]. This complex background for pattern analysis involves the use of many features and the analysis of their relationships, which is typically a graph problem as this representation, contrary to other representations such as the string–based one, allow to describe relational information in the patterns under consideration. Besides, the traditional feature–based representations cannot satisfactorily deal with data that is naturally divided into parts or where contextual information is essential for the classification process. For these reasons, the design of efficient graph–based algorithms will certainly be one of the major challenge of the next decade.

Graph–based representations constitute a typical way of representing patterns in the field of structural pattern recognition, which arise when these patterns are decomposed into parts and relationships between them. They constitute a useful representation because on the one hand the graph can encode complex

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 399–406, 2009.
© Springer-Verlag Berlin Heidelberg 2009

patterns in many research fields, such as computer vision, pattern recognition, robotic mapping or machine learning. On the other hand, theorems and algorithms from graph theory can be also employed to represent specific properties of the modeled system, such as deformations and forces, color or velocities and movements, as properties of the vertices or edges of the graph. Such representations and algorithms are quite natural and it is easy to find applications, such as image segmentation or pattern matching, where they are widely applied. Graph representations also pose unique problems in machine learning, since they are non–vectorial in nature and require new methodology to be developed if they are to be learned from image data.

This paper briefly reviews the application of the graph theory in different research fields: image processing and computer vision, mobile robotics, data mining, chemistry or bioinformatics. In the next Sections, the explanation of concepts which are related to these fields is combined with topics from the graph theory, such as graph matching, graph partitioning or graph clustering.

2 Graph-Based Image Processing and Computer Vision

Applications of graphs in image processing and computer vision include representations for cognitive vision inspired from human descriptions of videos [2], graph-based fingerprint representation encoding topology [3], human problem solving and the Traveling Salesman Problem [4], extraction of image primitives, such as line segments or salient regions, homotopic transformation for both combinatorial maps and weighted combinatorial maps, image smoothing [22], shape-from-shading [23] and efficient processing of huge amounts of structured sensory data for vision [5]. However, the specific image processing task where graphs have been widely used is in image segmentation. Segmentation can be defined as the process of decomposing an image into regions which are homogeneous according to some criteria. In this process, the internal properties of the region help to identify it, while its external properties (inclusion, adjacency,...) are used to group regions having a particular consistent meaning [6]. The segmentation algorithm must adapt the resulting regions to the contents of the image, being also interesting for the majority of applications that it also assures the spatial consistency or intra–connectivity of these regions. Graph–based image segmentation techniques generally represent the problem in terms of a graph where each vertex corresponds to a pixel in the image, and the edges connect pairs of neighboring pixels. A weight is associated with each edge based on some property of the pixels that it connects, such as their image intensities. The earliest graph–based methods use fixed thresholds and local measures in computing a segmentation [7]. However, it is not adequate to assume that regions have nearly constant or slowly varying intensities. Felzenszwalb and Huttenlocher [8] propose an adaptive criterion which determines that the intensity differences across the boundary of two regions are perceptually important if they are large relative to the intensity differences inside at least one of the regions. Other segmentation approaches are based on finding minimum cuts in the image graph. The cut criterion is designed in order to minimize the similarity between pixels that are being split [9]. Finally, in order to merge local and global information, the Global Probability of Boundary (gPb) is a recently proposed boundary detector which combines local information derived from low–level image features, such as brightness, color and texture, with global information obtained from spectral partitioning [10].

Segmentation task has been also widely accomplished by means of image pyramids [6]. Instead of performing image segmentation based on a single representation of the input image, a pyramid segmentation algorithm describes the contents of the image using multiple representations with decreasing resolution. Segmentation is then thought as the process of grouping visual information and the perceptual organization of the image is described by a hierarchy of partitions ordered by inclusion. The base of this hierarchy is the whole image, and each region represents an object at a certain scale of observation [11]. Data structures for encoding each hierarchy level range from arrays to simple graphs, dual graphs, combinatorial maps and generalized maps. Depending on the data structure and the reduction algorithm employed to build a hierarchy level from the level below, pyramids can be regular or irregular.

Irregular pyramid segmentation algorithms exhibit interesting properties with respect to segmentation algorithms based on a single representation. Thus, local operations can adapt the pyramidal hierarchy to the topology of the image, allowing the detection of global features of interest and representing them at low resolution levels. The *Ultrametric Contour Map* (UCM) approach [11] represents the geometric structure of the image by the soft boundary image associated to a family of nested segmentations. A threshold at level k on a UCM provides always a set of closed curves, which corresponds to the boundaries of the segmentation at this scale. Generic ultrametric distances for boundary extraction are defined by integrating local contour cues along the regions' boundaries and combining this information with intra-region attributes. Haxhimusa and Kropatsch [12] propose a hierarchical partitioning approach which uses a pairwise similarity function on a graph-based representation of the image. This function encapsulates the intuitive notion of contrast measuring the difference along the boundary of two regions relative to a measure of differences of region's internal differences. Two components are merged if there is a low-cost connection between them. The *hierarchy of partitions* approach allows to find region boundaries quickly in a 'stimulus-driven' way based on local differences in a specific feature, like as in preattentive vision.

3 Graphs in Mobile Robotics

The ability to encode structural patterns allows graph-based representations to be widely used for environment mapping. In fact, if we consider the two fundamental paradigms for mapping in mobile robotics (metric and topological), all approaches which implement the topological strategy typically represent the environment by means of a graph. Topological maps are usually more difficult to build than metric ones, but they do not suffer from the huge data load and time complexity associated to metric maps. Besides, topological maps are usually more compact, since their resolution is determined by the complexity of the environment. Thus, they allow fast path planning and provide more natural interfaces for human instructions. The principle of topological maps is to split the free-space of the real environment into a small number of regions. This encoding is usually conducted by means of simple graphs [13]. Vertices in such graphs correspond to different environment regions and edges indicate spatial relationships between them. However, simple graphs only takes into account adjacency relationships, being unable to distinguish from the graph an adjacency relation from an inclusion relation between two regions. Besides, if there is two non-connected boundaries which will allow a robot to cross from one region to another one, the simple graph only joins these nodes by one edge. These limitations can be raised if dual graphs are employed because their structure is adapted to the processed data and they correctly encode the topology in 2D [14].

On the other hand, in order to deal with large, complex environments, the internal representation acquired by the robot can be organized as a hierarchy of maps which represent the whole environment at different levels of abstraction. Typically, these hierarchical representations consists of two layers: a metric map and a higher-level topological map. The hybrid approach usually attaches a local metric map to the nodes of a graph-based environment representation, where arcs represent coordinates transformation between vertices. Thus, the complexity can be bounded within each local map. The problem is then to define what part of the mapped environment is associated to each topological vertex. Geometrical methods such as generalized Voronoi graphs [15] have been also used to segment the metric space representation. Recently, Zivkovic et al [13] and Blanco et al [16] have proposed to represent the base-level map as a graph, and to decompose it into nodes using efficient approximate solutions to the normalized graph cut criterion. This graph-partitioning method can be generalized for dividing a graph into a variable number of sub-graphs [16]. The main disadvantage of these approaches is that the whole base-level map must be built in advance. An alternative solution is to apply this partitioning algorithm periodically to the acquired local map [17]. In a recent work, Brunskill et al [18] have proposed a new algorithm to automatically decompose a map into submap segments using a spectral clustering approach. Finally, the problem of exploiting graph learning and matching for robot localization, has been tackled in [31].

4 Graph Matching for Object Recognition

In order to evaluate the similarity between patterns, the use of graph–based representations typically imply the computation of measures which are exponential in the number of involved vertices. Therefore, such computations are only feasible for rather small graphs only. The process of evaluating the structural similarity of graphs is commonly referred as graph matching, and it constitutes a challenging problem for large graphs. On the contrary, the matching problem can be solved in polynomial time when dealing with trees. Then, several authors have adopted the solution of reducing the graph representations into attributed trees through edit operations (removing, adding or replacing of a vertex or edge) and apply these solutions. Unfortunately, this reduction can discard a lot of information about the structure of the original graph in certain applications. Of course, the use of continuous, although approximate polynomial-time algorithms (Softassign [32] and Motzking-Strauss/Replicator Dynamics [33]) and their modifications and improvements (e.g. [34]) the basis of graph-matching has yielded good practical results. The key underlying idea is that transforming the problem into a continuous one it is possible to tackle the problem of finding the optimal matching or the maximal clique efficiently. Thus, as approximate solutions to the latter NP complete problems are now available in polynomial time, this line of research is still in progress. Two interesting challenges is this context are the proposal of new cost functions [35] (less-prone to plateaus and local maxima) and the enumeration of several cliques, because in real world the maximal clique does not necessarily implies the best solution.

On the other hand, one of the most flexible approaches for error-tolerant matching is based on the edit distance of graphs [41]. The idea of graph edit distance is to define the dissimilarity of graphs by the amount of distortion which is needed to transform one graph into the other. Based on a vertex and edge distortion model, the distance between two graphs is given by the minimum amount of distortion that is needed to transform one of the graph into the other one. The main drawback of graph edit distance is its computational complexity, although several approaches have been proposed to solve this problem. Other approaches are based on linear algebra techniques, exploiting the adjacency, incident or Laplacian matrices of a graph. The idea is to consider the spectral decomposition of the graph rather than the graph itself [41]. Finally, to match two non-attributed graphs, several approaches based on the maximum common subgraph or minimum common supergraph have been also proposed.

5 Graph Clustering, Embedding and Learning

In statistical pattern recognition, pattern samples are described by feature vectors and the recognition is based on the assumption that patterns which belong to the same class are located in a compact region of the feature space. As it has been aforementioned, graphs are widely used in the field of structural pattern recognition, as they provide the possibility to represent structural relationships. In order to combine the high representational abilities given by graphs with the large amount of clustering and classification techniques which use feature vectors, graphs can be embedded in a n-dimensional feature vector space [41]. In this regard, spectral-graph theory is being widely used to embed structural patterns. Initial remarkable approaches, like the MDS/PCA embedding of graph spectra [20] opened a way of embedding graphs onto a Riemannian manifold [24]. In this latter case, graph nodes are embedded and their coordinates are given by the eigenvectors of the Laplacian matrix, after applying a centered MDS to it.

Given the coordinates is possible to reformulate the graph-matching problem in terms of aligning pairs of manifolds. Actually, the problem is transformed into point-set matching with the Procustes algorithm. Another recent example of exploiting spectral-graph theory for embedding, and for clustering with graphs, is to exploit the concept of commmute time, which can be obtained from the Green function [21]. Other methods for graph embedding rely on exploiting similarities coming from matchings, for instance, from the building of tree prototypes (super-trees) and considering each node of the prototype as a dimension of the pattern space. Thus, in this latter approach, each tree has assigned a pattern and the set of patterns may be analyzed through PCA [19]. More recently, the idea behind the ISOMAP approach has been considered recently for embedding [25] and then clustering graphs.

Research on graph embedding followed by clustering (usually pairwise [27], has his counterpart in the problem of simultaneously learning the prototypes and finding the classes in the graph domain. Pioneering works in this context are either generative or metric. The generative approach has been successful when working with trees [26], although an EM-based approach for unsupervised learning prototypes from graphs is presented in [28]. However, in this latter case, the model is only first-order generative. On the other hand, the metric-based prototype-learning approach is represented by the concept of median graph introduced in [29]. However, due to the huge complexity of computing it, it has been recently proposed to exploit graph embedding [30].

6 Conclusion and Future Directions

Along this paper, we have described significant advances, though not an exhaustive enumeration, of graph-based methods in several research areas like computer vision, image processing, mobile robotics, object recognition and categorization, structural learning, and clustering. In all the named areas, it is quite clear that the improvement of the representational power (e.g. in the case of pyramids), the building of flexible and adaptive hierarchies of graphs (in the case of mobile robotics), the proposal of new approximate methods (in the case of object recognition) and also of generative models (in object categorization) and, finally, the formulation of new embeddings and graph descriptors, are challenging research topics for the future. Being more precise, we will cite here two examples which are transversal to most of the latter challenges. One of them is the definition and use of hypergraphs [37][39][38], and the other one is the quantification of graph complexity [40]. However, independently of the latter theoretical (but with a sound practical background) developments, and the emergence of new fields of application, like the recognition of 3D objects by exploiting Reeb-graphs, it is necessary to develop sound contests and benchmarking to some, or all, of the challenges listed above. This latter task implies the collaborative efforts of graph-theorists and experts in fields like computer vision, bioinformatics, chemistry and graphics, to name a few, in order to achieve really good databases of graphs, so that the performance proposed techniques can be compared properly.

References

1. Cheung, K., Shin, D.: A graph–based meta–data framework for interoperation between genome databases. In: First IEEE Int. Symposium Bioinformatics Biomedical Eng. (2000)
2. Ion, A., Haxhimusa, Y., Kropatsch, W.: A graph–based concept for spatiotemporal information in cognitive vision. In: M. Vento and L. Brun (Eds.), GbR 2005, 223–232, Springer Verlag (2005)
3. Ölz, W., Kropatsch, W.: Graph representation of fingerprint topology. In: Computer Vision Winter Workshop, 51-58 (2004)
4. Pizlo, Z., Stefanov, E., Saalweachter, J., Haxhimusa, Y., Kropatsch, W.: Traveling salesman problem: A foveating pyramid model. The Journal of Problem Solving 1(1), 83–101 (2006)
5. Kropatsch, W.: Benchmarking graph matching algorithm. Pattern Recognition Letters 24(8), 1051–1059 (2003)
6. Marfil, R., Molina-Tanco, L., Bandera, A., Rodríguez, J.A., Sandoval, F.: Pyramid segmentation algorithms revisited. Pattern Recognition 39(8), 1430–1451 (2006)
7. Zahn, C.: Graph-theoretic methods for detecting and describing gestalt clusters. IEEE Transactions on Computing 20, 68–86 (1971)
8. Felzenszwalb, P., Huttenlocher, D.: Image segmentation using local variation. In: IEEE Conf. on Computer Vision and Pattern Recognition, 98–104 (1998)
9. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Pattern Analysis Machine Intell. 22(8), 888–905 (2000)
10. Maire, M., Arbeláez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: Int. Conf. Computer Vision Pattern Recognition (2008)
11. Arbeláez, P., Cohen, L.: A metric approach to vector–valued image segmentation. Int. Journal of Computer Vision 69, 119–126 (2006)
12. Haxhimusa, Y., Glantz, R. and Kropatsch, W.G.: Constructing stochastic pyramids by MIDES - maximal independent directed edge set. In: E. Hancock and M. Vento (Eds.), GbR2003, LNCS 2726, 35–46, Springer Verlag (2003)
13. Zivkovic, Z., Bakker, B., Kröse, B.: Hierarchical map building and planning based on graph partitioning. In: Proc. of the IEEE Int. Conf. Robotics and Automation, 803–809 (2006)
14. Marfil, R., Bandera, A.: Graph abstraction preserving the topology: Application to environment mapping for mobile robotics. In: Computational Algebraic Topology within image context (2008)
15. Choset, H., Nagatani, K.: Topological simultaneous localisation and mapping: Towards exact localisation without explicit localisation. IEEE Trans. Robotics and Automation 17 (2), 125–137 (2001)
16. Blanco, J., González, J., Fernández-Madrigal, J.: Consistent observation grouping for generating metric–topological maps that improves robot localization. IEEE Int. Conf. Robotics Automation, 818–823 (2006)
17. Blanco, J., Fernández-Madrigal, J., González, J.: A new approach for large–scale localization and mapping: Hybrid metric–topological SLAM. IEEE Int. Conf. Robotics Automation, 2061–2067 (2007)
18. Brunskill, E., Kollar, T., Roy, N.: Topological mapping using spectral clustering and classification. IROS 2007: 3491–3496 (2007)
19. Torsello, A., Hancock,E.R: Graph Embedding using Tree Edit Union. Pattern Recognition, 40, 1393–1405 (2007)

20. Luo, B., Wilson,R.C., Hancock,E.R.: Spectral Embedding of Graphs. Pattern Recognition, 36, 2213–2223 (2003)
21. Qiu, H., Hancock, E.R: Clustering and Embedding using Commute Times. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 1873–1890 (2007)
22. Fan Zhang, F., Hancock, E.R.: Graph Spectral Image Smoothing using the Heat Kernel. Pattern Recognition 41(11): 3328–3342 (2008)
23. Robles-Kelly, A., Hancock, E.R.: A Graph Spectral Approach to Shape-from-shading. IEEE Transactions on Image Processing, 13, pp. 912–926 (2004)
24. Robles-Kelly, A., Hancock, E.R.: A Riemannian Approach to Graph Embedding. Pattern Recognition, 40, 1042–1056 (2007)
25. Xiao, B., Torsello, A., Hancock, E.R.: Isotree: Tree clustering via Metric Embedding. Neurocomputing 71(10-12): 2029–2036 (2008)
26. Torsello, A., Hancock, E.R.: Learning Shape-Classes using a Mixture of Tree-unions. IEEE Trans. Pattern Anal. Machine Intell., 28(6):954–967 (2006)
27. Torsello, A., Robles-Kelly, A., Hancock, E.R: Discovering Shape Classes using Tree Edit Distance and Pairwise Clustering. International Journal of Computer Vision, 72, 259–285 (2007)
28. Lozano, M.A., Escolano, F.: Protein Classification by Matching and Clustering Surface Graphs. Pattern Recognition 39(4), 539–551 (2006)
29. Jiang, X., Münger, A., Bunke, H.: On Median Graphs: Properties, Algorithms, and Applications. IEEE Trans. Pattern Anal. Mach. Intell. 23 (10) 1144–1151. (2001)
30. Ferrer, M., Valveny, E., Serratosa, F., Bunke, H.: Exact Median Graph Computation Via Graph Embedding. SSPR/SPR 2008: 15-24 (2008)
31. Escolano, F., Bonev, B., Suau, P., Aguilar, W., Frauel, Y., Sáez, J.M., Cazorla, M.: Contextual Visual Localization: Cascaded Submap Classification, Optimized Saliency Detection, and Fast View Matching. IROS 2007: 1715–1722 (2007)
32. Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. IEEE Trans. Pattern. Anal. Mach. Intell. 18 (4) 377–388 (1996)
33. Pelillo, M.: Replicator Equations, Maximal Cliques, and Graph Isomorphism. Neural Comput. 11 1933–1955 (1999)
34. Luo, B., Hancock E.R.: Structural Graph Matching using the EM Algorithm and Singular Value Decomposition. IEEE Trans. Pattern. Anal. Mach. Intell. 23 (10), 1120–1136 (2001)
35. Lozano, M.A., Escolano, F.: A Significant Improvement of Softassign with Diffusion Kernels. SPR+SSPR 2004, Lecture Notes in Computer Science, vol. 3138, 76–84 (2004)
36. Rota-Bulò, S., Torsello, A., Pelillo, M.: A Continuous-Based Approach for Partial Clique Enumeration. Gbr 2007, Lecture Notes in Computer Science, vol. 4538, 61–70 (2007)
37. Ren, P., Wilson, R.C, Hancock, E.R.: Spectral Embedding of Feature Hypergraphs. SSPR/SPR 2008: 308–317 (2008)
38. Rota-Bulò, S., Albarelli, A., Pelillo, M., Torsello, A.: A Hypergraph-based Approach to Affine Parameters Estimation, ICPR 2008: (2008)
39. Xia, S., Hancock, E.R.: 3D Object Recognition Using Hyper-Graphs and Ranked Local Invariant Features. SSPR/SPR 2008: 117–126 (2008)
40. Escolano, F., Hancock, E.R., Lozano, M.A.: Polytopal Graph Complexity, Matrix Permanents, and Embedding. SSPR/SPR 2008: 237–246 (2008)
41. Riesen, K., Neuhaus, M., Bunke, H.: Bipartite Graph Matching for Computing the Edit Distance of Graphs. Gbr 2007, Lecture Notes in Computer Science, vol. 4538, 1–12 (2007)

Kernelization of Softassign and Motzkin-Strauss Algorithms

M.A. Lozano and F. Escolano

Robot Vision Group, Departamento de Ciencia de la Computacin e IA,
Universidad de Alicante, Spain

Abstract. This paper reviews two continuous methods for graph matching: Softassign and Replicator Dynamics. These methods can be applied to non-attributed graphs, but considering only structural information results in a higher ambiguity in the possible matching solutions. In order to reduce this ambiguity, we propose to extract attributes from non-attributed graphs and embed them in the graph-matching cost function, to be used as a similarity measure between the nodes in the graphs. Then, we evaluate their performance within the reviewed graph-matching algorithms.

1 Graph Matching Methods

The first works on graph matching arose as an application to structural object recognition. In particular, Barrow and Popplestone in [BP1] and Fischler and Elschlager in [FE1] demonstrated the viability of using relational structures to tackle the problem of object recognition. From that moment, graph matching became an interesting topic for computer vision and pattern recognition. Therefore, until now many approaches have been developed in order to solve this problem.

The space of feasible solutions for graph matching is a discrete space, in which each possible solution consists of a set of matchings between the nodes of the input graphs. We need a cost function to evaluate these possible solutions, and a method to optimize such a function. We can distinguish two groups of optimization methods for graph matching: discrete and continuous methods.

Continuous methods rely on transforming the discrete search space into a continuous one and then exploiting optimization techniques to find a, typically approximate, solution. Within this kind of methods we find Softassign, which is one of the first continuous graph-matching algorithms [GR1]. We also find an alternative continuous formulation for graph-matching relying on the Motzkin-Straus theorem [P1].

1.1 Softassign

The Gold & Rangarajan [GR1] cost function is the standard quadratic formulation for graph-matching.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 407–414, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Given two graphs $G_X = (V_X, E_X)$, with nodes $a \in V_X$ and edges $(a,b) \in E_X$, and $G_Y = (V_Y, E_Y)$, with nodes $i \in V_Y$ and edges $(i,j) \in E_Y$, their adjacency matrices X and Y are defined by:

$$X_{ab} = \begin{cases} 1 & \text{if } (a,b) \in E_X \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad Y_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E_Y \\ 0 & \text{otherwise} \end{cases}$$

A feasible solution to the graph matching problem between G_X and G_Y is encoded by a matrix M of size $m \times n$, being $m = |V_X|$ and $n = |V_Y|$, with binary variables

$$M_{ai} = \begin{cases} 1 & \text{if } a \in V_X \text{ matches } i \in V_Y \\ 0 & \text{otherwise} \end{cases}$$

satisfying the constraints defined respectively over the rows and columns of M

$$\sum_{i=1}^{m+1} M_{ai} = 1, \forall a \ \text{ and } \ \sum_{a=1}^{n+1} M_{ai} = 1, \forall i \ , \tag{1}$$

where equality comes from introducing slack variables for registering outliers.

Following the Gold and Rangarajan [GR1] formulation we are interested in finding the feasible solution M that minimizes the following cost function,

$$F(M) = -\frac{1}{2} \sum_{a=1}^{m} \sum_{i=1}^{n} \sum_{b=1}^{m} \sum_{j=1}^{n} M_{ai} M_{bj} C_{aibj} \ , \tag{2}$$

where typically $C_{aibj} = X_{ab} Y_{ij}$, that is, when $a \in V_X$ matches $i \in V_Y$, it is desirable that nodes b adjacent to a (with $X_{ab} \neq 0$) and nodes j adjacent to i (with $Y_{ij} \neq 0$) also match, that is $M_{ai} = M_{bj} = 1$. This is the well known rectangle rule (in maximization terms we want to obtain as more rectangles as possible).

However, it has been reported that the performance of the algorithm decays significantly at mid and high levels of structural corruption, and also that such a decay can be attenuated by optimizing an alternative non-quadratic energy function. Regarding this, Finch proposes in [FWH1] a more elaborated version of this cost function including additional terms, and in [FWH2] proposes a non-quadratic version of its cost function.

In order to optimize the latter cost function, Gold and Rangarajan propose in [GR1] a relaxation method: Softassign algorithm. This algorithm performs a deterministic annealing process, in which the assignment variables are updated by

$$M_{ai} = \exp\left[-\frac{1}{T}\frac{\partial F}{\partial M_{ai}}\right] = \exp\left[\frac{1}{2T}\sum_{i=a}^{m} M_{bj} C^{K}_{aibj}\right] \ ,$$

where T is the temperature control parameter. Then, these assignments feed a Sinkhorn process [S1], which iteratively normalizes rows and columns in order to obtain a matrix M satisfying the constraints defined in Eq. 1. After this process, we obtain a doubly stochastic matrix, decrease T, and a new iteration begins. The final doubly stochastic matrix encodes the matchings between the nodes of G_X and G_Y. It would be possible that the variables in this matrix were not binary, but belonging to $[0, 1]$, and that some nodes were assigned to many nodes (ambiguous assignments). Therefore, this matrix is transformed into a permutation matrix by a proper clean-up process.

1.2 Motzkin-Strauss and Replicator Dynamics

Some cost functions for graph-matching, such as Gold and Rangarajan one, are related to the maximum clique formulation [P1]. Given two graphs $G' = (V', E')$ and $G'' = (V'', E'')$, the possible matchings between them can be represented by a non-directed graph $G = (V, E)$ known as association graph. Its node set V is defined as the possible matchings between the nodes of both graphs V' and V'':

$$V = V' \times V''$$

And its edges represent the compatibility between two possible matchings:

$$E = \{((i, h), (j, k)) \in V \times V : i \neq j, h \neq k, \text{and } (i, j) \in E' \Leftrightarrow (h, k) \in E''\}$$

Given a graph $G = (V, E)$, a clique is defined as a node set C whose nodes are all adjacent between them, that is, for all $i, j \in C$ there exists an edge $(i, j) \in E$. The problem of graph-matching can be posed as finding the maximum clique in the association graph, that is, the maximum set of compatible matches. Therefore, this evaluates how much structure is conserved in the matching.

In [P1], Pelillo proposes a continuous formulation for the maximum clique problem, based on the Motzkin-Straus theorem. This relaxation process uses the adjacency matrix from the association graph A. The domain of possible solutions is the space S_n, being n the number of nodes of the graph:

$$S_n = \{x \in \mathbb{R}^n : \forall i = 1 \ldots n \; x_i \geq 0 \wedge \sum_{i=1}^{n} x_i = 1\} \tag{3}$$

The possible solutions will be encoded with a vector $x \in S_n$. The elements x_i with a value higher than 0 will correspond with the nodes selected for the clique in this particular solution. Our aim is to find a vector $x \in S_n$ which maximizes the following quadratic form:

$$f(x) = x^T A x = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

The Motzkin-Straus theorem [MS1] establishes a relation between the global maximizers of the function f and the maximum clique in the graph given by the adjacency matrix A. This function can be maximized through a continuous optimization process: *Replicator Dynamics*.

First of all, the replicator process is initialized with the barycenter, in order not to favour any particular solution, that is:

$$\mathbf{x}(0) = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)^T$$

Then, this solution $\mathbf{x}$ is optimized by iteratively applying the following discrete-time differential equation [TJ1]:

$$x_i(t+1) = x_i(t)\frac{(A\mathbf{x}(t))_i}{\mathbf{x}(t)^T A\mathbf{x}(t)} \tag{4}$$

Provided that A is a non-negative real-valued symmetrical matrix, the function $\mathbf{x}(t)^T A\mathbf{x}(t)$ is strictly increasing with t along any non-stationary trajectory $\mathbf{x}(t)$ [LMO1], and it converges to a stationary point, that is, a point satisfying $x_i(t+1) = x_i(t)$ [BE1]. This point will correspond to the maximum clique of the association graph, and such a clique encodes the optimal matching between the input graphs.

2 Kernelizing Graph Matching

When the only information we have is the graph structure, we usually have a great ambiguity in the possible solutions to the problem, which is an intrinsic difficulty for obtaining the optimal solution, due to the fact that there could be many optimal solutions, and maybe some of these solutions are not what we are looking for. This problem is harder when obtaining a matching between a graph and a subgraph, due to the fact the this subgraph could fit many places of the graph. However, when we have attributed graphs, these attributes disambiguate the space of possible solutions and help to guide the search process. Our proposal consists in obtaining attributes from non-attributed graphs.

We address the key point of extracting good attributes for the nodes of non-attributed graphs by exploiting recent theoretical results in spectral graph theory [C1] : the definition of diffusion kernels on graphs [KL1] and their generalization to other families of kernels [SK1]. These latter works have transferred to the discrete domain of graphs the concept of a *kernel*, originally defined in the vector domain. When applied to graphs, kernels provide a similarity measure between the vertices of the same graph. In the case of diffusion kernels, defined by Kondor and Lafferty in [KL1], such a similarity can be seen as the sum of probabilities of all paths connecting the compared vertices. This kind of kernel is computed from the matrix exponentiation of the Laplacian of the adjacency matrix. As the Laplacian encodes information about the local structure of the graph, the global structure emerges in the kernel. These diffusion kernels on graphs are positive semi-definite matrices, with as many rows and columns as the number of vertices in the graph.

Table 1. Penalization Functions and Regularization Kernels

$r(\lambda)$	$K = r^{-1}(\mathcal{L})$	Name
$1 + \beta\lambda$	$(I + \beta\mathcal{L})^{-1}$	Regularized Laplacian
$e^{\beta\lambda}$	$e^{-\beta\mathcal{L}}$	Diffusion Process
$(a - \lambda)^{-p}$	$(aI - \mathcal{L})^{p}$	$p-$step Random Walk
$(\cos \lambda\pi/4)^{-1}$	$\cos \mathcal{L}\pi/4$	Inverse Cosine

Although understanding the role of the diffusion kernel in graphs is intuitive because these kernels are the discrete version of continuous Gaussian kernels, this is not the case for the regularization kernels recently proposed by Smola and Kondor [SK1]. These latter kernels are derived from studying the usefulness of the Laplacian of a graph (and its normalized version) as a smoothing operator. Considering smoothing operators from a spectral point of view, it results that a family of kernels emerges from considering different penalization functions, as it is shown in Fig. 1. It can be proved that the inverse of the so called *regularization matrix* for each element of the family yields a kernel (actually, the diffusion kernel belongs to this family).

Each row in a kernel matrix represents a node in the graph. In order to obtain an attribute for this node we can extract information from its corresponding row in the kernel matrix. We propose to use as attribute the entropy of this row, or only the diagonal value in the kernel matrix corresponding to this node. These attributes are not enough for obtaining a matching between all the nodes of two graphs, because although they are able to distinguish between correct and wrong matches in most of cases, there also exists a considerable amount of cases in which these attributes fail [L1]. Given that these attributes are only local information, when they fail the structure of the graph could be broken, because we are not considering the connections between the nodes in the graphs.

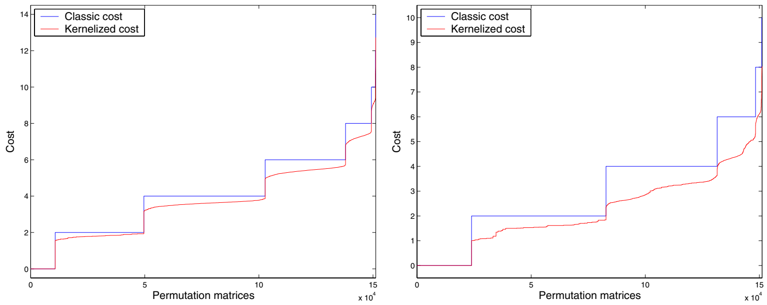
Therefore, these attributes must be used as support information within a graph matching algorithm which considers the structure of the input graphs. We will use the Softassign algorithm for graph matching, introducing the divergence between the node attributes in its cost function.

We can use the Gold and Rangarajan [GR1] formulation, inserting in their cost function the attributes obtained from kernels. A simple way of *kernelizing* this cost function, defined in Eq. 2, is to redefine C_{aibj} as

$$C^{K}_{aibj} = X_{ab}Y_{ij}P(match|\mathcal{K}^{X}_{a}, \mathcal{K}^{Y}_{i})P(match|\mathcal{K}^{X}_{b}, \mathcal{K}^{Y}_{j}) \ , \tag{5}$$

where the attributes $P(match|\mathcal{K}^{X}_{a}, \mathcal{K}^{Y}_{i})$ is the probability that nodes $a \in V_X$ and $b \in V_Y$ match, given their kernel attributes (it is also possible to use other kinds of attributes, such as degree or spectral information). This probability is obtained from an analysis of the distribution of divergences between kernel attribute values for both correct and wrong matches [L1].

The latter definition of C^{K}_{aibj} ensures that $C^{K}_{aibj} \leq C_{aibj}$, and the equality is only verified when nodes a and i have equal attributes, and the same for nodes b



(a) (b)

Fig. 1. Shape of the classic and the kernelized version of the cost function for the matching between a 10-node graph and the result of removing 4 nodes and a (a) 0% and (b) 25% of edges respectively from the same graph

and j. In practice, this weights the rectangles in such a way that rectangles with compatible attributes in their opposite vertices are preferred, and otherwise they are underweighted.

In Fig. 1 we compare the shape of the classic and the kernelized version of the cost function. The x-axis represents all the possible permutation matrices, ordered by its kernelized cost. We can see that the classic version has more ambiguous solutions (plateaus) than the kernelized one. We can see that as we remove edges, the ambiguity in the classic cost function is increased (plateaus becomes larger), whereas the kernelized version yields few ambiguous solutions, which helps the algorithm to achieve the optimal solution.

These attributes could be easily applied to other graph matching approaches. For instance, they can be applied to Replicator Dynamics. A straightforward way to kernelize this algorithm is to introduce these attributes as weights in the association graph edges:

$$A_{(i,h),(j,k)} = \begin{cases} P(match|\mathcal{A}_i^{G'}, \mathcal{A}_h^{G''})P(match|\mathcal{A}_j^{G'}, \mathcal{A}_k^{G''}) & \text{if } i \neq j, h \neq k, \text{and} \\ & (i,j) \in E' \Leftrightarrow (h,k) \in E'' \\ 0 & \text{otherwise} \end{cases}$$

Where A is the adjacency matrix of the association graph, E' is the edge set of the first input graph G', and E'' is the edge set of the second one G''.

3 Experimental Results

The first experiments to validate the effectiveness of kernels as attributes consist in applying the kernelized Softassign algorithm to randomly generated graphs. In these experiments we will use graphs of 50 nodes with edge densities of 10%

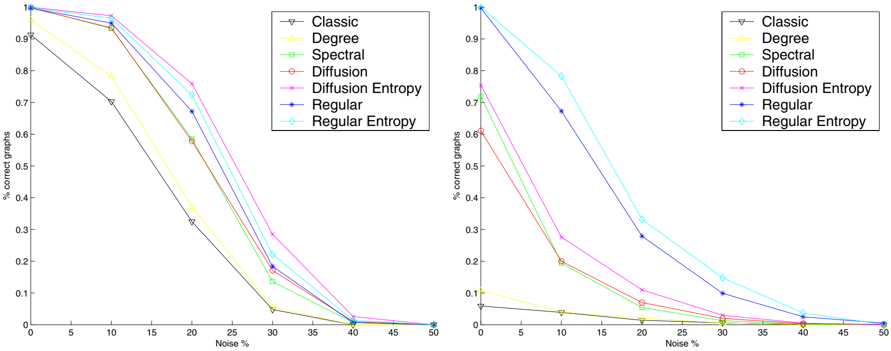


Fig. 2. Softassign decay for edge densities of 10% (left) and 50% (right)

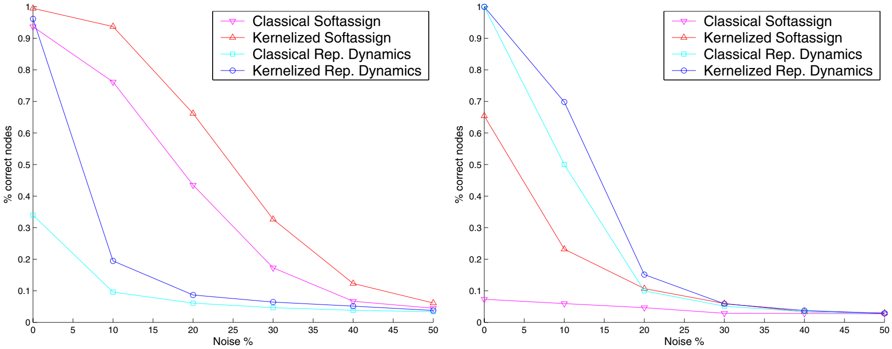


Fig. 3. Softassign and Replicator Dynamics decay for edge densities of 10% (left) and 50% (right)

and 50%. For each type of graphs, we will perform some tests increasing the edge noise level from 0% to 50%, with the different types of attributes. For each type of graphs and noise we have performed 1000 tests, and in Fig. 2 we show the average obtained results for each case. We can observe that the kernelized version outperforms the classical one, especially when regularized Laplacian kernel attributes are applied to high density graphs.

It is also interesting to test the attributes provided by kernels within other graph-matching approaches. In Fig. 3 it is shown a performance comparative of Softassign and Replicator Dynamics algorithms both with their classical and their kernelized versions. We can observe that the behaviour of Softassign and Replicator Dynamics depends on the edge density. Softassign yields better results when the edge density is low, whereas with higher edge densities Replicator Dynamics is preferable, but always their kernelized versions outperform the classical ones.

References

[BP1] Barrow, H., Popplestone, R.: Relational descriptions in picture processing. Machine Intelligence VI, 377–396 (1971)

[BE1] Baum, L.E., Eagon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bull. Amer. Math. Soc. 73, 360–363 (1967)

[C1] Chung, F.R.K.: Spectral Graph Theory. Conference Board of the Mathematical Sciences (CBMS). American Mathematical Society 92 (1997)

[CS1] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)

[FWH1] Finch, A.M., Wilson, R.C., Hancock, E.R.: Symbolic Graph Matching with the EM algorithm. Pattern Recognition 31(11), 1777–1790 (1998)

[FWH2] Finch, A.M., Wilson, R.C., Hancock, E.R.: An Energy Function and Continuous Edit Process for Graph Matching. Neural Computation 10(7) (1998)

[FE1] Fischler, M., Elschlager, R.: The representation and matching of pictorical structures. IEEE Transactions on Computers 22(1), 67–92 (1972)

[KL1] Kondor, R.I., Lafferty, J.: Diffusion Kernels on Graphs and other Discrete Input Spaces. In: Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), pp. 315–322 (2002)

[GR1] Gold, S., Rangarajan, A.: A Graduated Assignement Algorithm for Graph Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(4), 377–388 (1996)

[L1] Lozano, M.A.: Kernelized Graph Matching and Clustering. Ph.D. Thesis, University of Alicante, Spain (2008)

[LMO1] Lyubich, Y., Maistrowskii, G.D., Ol'khovskii, Y.G.: Selection-induced convergence to equilibrium in a single-locus autosomal population. Problems of Information Transmission 16, 66–75 (1980)

[MS1] Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turn. Canad. J. Math. 17, 533–540 (1965)

[MMRTS1] Mller, K.-R., Mika, S., Rshc, G., Tsuda, K., Schlkopf, B.: An Introduction to Kernel-based Learning Algorithms. IEEE Transactions on Neural Networks 12(2), 181–201 (2001)

[P1] Pelillo, M.: Replicator Equations, Maximal Cliques, and Graph Isomorphism. Neural Computation 11(8), 1933–1955 (1999)

[SS1] Schlkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)

[S1] Sinkhorn, R.: A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. Annals of Mathematical Statistics 35, 876–879 (1964)

[SK1] Smola, A., Kondor, R.I.: Kernels and Regularization on Graphs. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS, vol. 2777, pp. 144–158. Springer, Heidelberg (2003)

[TJ1] Taylor, P., Jonker, L.: Evolutionarily stable strategies and game dynamics. Math. Biosci. 40, 145–156 (1978)

Connectivity Forests for Homological Analysis of Digital Volumes*

Pedro Real

Dpto. Matematica Aplicada I, E.T.S.I. Informatica,
Universidad de Sevilla,
Avda. Reina Mercedes, s/n 41012 Sevilla (Spain)
`real@us.es`

Abstract. In this paper, we provide a graph-based representation of the homology (information related to the different "holes" the object has) of a binary digital volume. We analyze the digital volume AT-model representation [8] from this point of view and the cellular version of the AT-model [5] is precisely described here as three forests (connectivity forests), from which, for instance, we can straightforwardly determine representative curves of "tunnels" and "holes", classify cycles in the complex, computing higher (co)homology operations,... Depending of the order in which we gradually construct these trees, tools so important in Computer Vision and Digital Image Processing as Reeb graphs and topological skeletons appear as results of pruning these graphs.

Keywords: digital volume, cell complex, chain homotopy, graph, tree, forest, homology, cohomology.

1 Introduction

Finding a concise, yet geometrically and topologically faithful digital representation of a digital volume is at the core of several research themes in graphics. We propose a new strategy for the design of succinct and efficient volume approximations. In [5], given a binary 26-adjacency voxel-based digital volume V, the homological information (that related to n-dimensional holes: connected components, "tunnels" and cavities) is extracted from a linear map (called homology gradient vector field) acting on a polyhedral cell complex $P(V)$ homologically equivalent to V. Based on that description, we define here a three-level graph data structure for representing digital volumes. This connectivity encoding technique attempt to reduce the redundancy inherent in many popular representations of polyhedral or triangular meshes in 3D.

In this paper, we first develop an algebraic homological study for dealing with special chain homotopies appearing in the technique developed in [5] for determining homology gradient vector fields for a binary voxel-based digital volume.

---

* This work has been partially supported by "Computational Topology and Applied Mathematics" PAICYT research project FQM-296, "Andalusian research project PO6-TIC-02268 and Spanish MEC project MTM2006-03722.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 415–423, 2009.
© Springer-Verlag Berlin Heidelberg 2009

From this study, we derive a graph-based representation of homology gvfs and, finally, the problem of encoding the connectivity in 3D is solved here using a hierarchical "cover" forest describing the homology of a cell continuous analogous (cell AT-model) of the volume.

2 Chain Homotopies and Gradient Vector Fields

We work here with 3-dimensional cell complexes. The cells are typically polyhedra, polygons, line segments and points, in decreasing order of degrees. For convenience, we call a cell of degree p a *p-cell.* Since the objects considered in this paper are embedded in $\mathbf{R}^3$ then the homology groups vanish for dimensions greater than 3 and they are torsion–free for dimensions 0, 1 and 2. Throughout the paper, we consider that the ground ring is the finite field $\mathbf{F}_2 = \{0, 1\}$. Let K be a three-dimensional cell complex. A *q–chain* a is a formal sum of simplices of $K^{(q)}$. We denote $\sigma \in a$ if $\sigma \in K^{(q)}$ is a summand of a. The q–chains form a group with respect to the component–wise addition; this group is the *qth chain group* of K, denoted by $C_q(K)$. There is a chain group for every integer $q \geq 0$, but for a complex in $\mathbf{R}^3$, only the ones for $0 \leq q \leq 3$ may be non–trivial. The boundary map $\partial_q : C_q(K) \to C_{q-1}(K)$ applied to a q–cell σ gives us the collection of all its $(q-1)$–faces which is a $(q-1)$–chain. By linearity, the boundary operator ∂_q can be extended to q–chains. For the rest of the paper, we adopt the notation (K, ∂) for representing a 3D cell complex. In the concrete case of a simplicial complex, the boundary of a q-simplex defined in terms of vertices $\sigma = \langle v_0, \ldots, v_q\rangle$ is defined by: $\partial_q(\sigma) = \sum \langle v_0, \ldots, \hat{v}_i, \ldots, v_q\rangle$, where the hat means that vertex v_i is omitted. In our case, taking into account that the 3-cells of our cell complexes can automatically be subdivided into tetrahedra, its boundary map can directly be derived from that of the component tetrahedra. It is clear that $\partial_{q-1}\partial_q = 0$. A chain $a \in C_q$ is called a *q–cycle* if $\partial\_q\ (a) = 0$. If $a = \partial_{q+1}(a')$ for some $a' \in C_{q+1}$ then a is called a *q–boundary.* Define the *qth homology group* to be the quotient group of q–cycles and q–boundaries, denoted by $H_q(\mathcal{C})$. Let $\mathcal{C} = \{C_q, \partial\ _q\}$ and $\mathcal{C}' = \{C'_q, \partial\ '_q\}$ be two chain complexes. A *chain map* $f : \mathcal{C} \to \mathcal{C}'$ is a family of homomorphisms $\{f_q : C_q \to C'_q\}_{q \geq 0}$ such that $\partial\ '_q f_q = f_{q-1}\partial\ _q$.

The following definitions are necessary in order to classify gradient vector fields.

Definition 1. [3] *Let* (K, ∂) *be a finite cell complex. A linear map of chains* $\phi : C_*(K) \to C_{*+1}(K)$ *is a* combinatorial gradient vector field *(or, shortly, combinatorial gvf) over* K *if the following conditions hold:*

1. *For any cell* $a \in K_q$, $\phi(a)$ *is a* $q+1$*-cell* b.
2. $\phi^2 = 0$

If we remove the first condition, then ϕ will be called an *algebraic gradient vector field.* If ϕ is a combinatorial gvf which is only non-null for a unique cell $a \in K_q$ and satisfying the extra-condition $\phi\partial\phi = \phi$, then it is called a (combinatorial) *integral operator* [6]. An algebraic gvf satisfying the condition $\phi\partial\phi = \phi$ is called an *algebraic integral operator*. An algebraic gvf satisfying the conditions $\phi\partial\phi = \phi$ and $\partial\phi\partial = \partial$ will be called a *homology gvf* [5]. A gvf is called *strongly-nilpotent* if it satisfies the following property: given any $u \in K_q$, and being $\phi(u) = \sum_{i=1}^{r} v_i$, then $\phi(v_i) = 0$, $\forall i$. We say that a linear map $f : C_*(K) \to C_*(K)$ is strongly null over an algebraic gradient vector field ϕ if given any $u \in K_q$, and being $\phi(u) = \sum_{i=1}^{r} v_i$, then $f(v_i) = 0$, $\forall i$.

Using homological algebra arguments, it is possible to deduce that a homology gvf ϕ over K determines a strong algebraic relationship connecting $C(K)$ and its homology vector space $H(K)$. Let us define a *chain contraction* $(f, g, \phi) : (C, \partial) => (C', \partial')$ between two chain complexes as a triple of linear maps such that $f : C_* \to C'_*$, $g : C'_* \to C_*$ and $\phi : C_* \to C_{*+1}$ and they satisfy the following conditions: (a) $id_C - gf = \partial\phi + \phi\partial; (b) f\, g = id_{C'}$; (c) $f\, \phi = 0$; (d) $\phi\, g = 0$; (e) $\phi\, \phi = 0$.

Given a chain contraction (f, g, ϕ), it is an elementary homological algebra result that Ker ϕ = Img + Imϕ and Imϕ is acyclic (i.e, it has null homology).

Proposition 1. *Let (K, ∂) be a finite cell complex. A homology gvf $\phi : C_*(K) \to C_{*+1}(K)$ over K give raise to a chain contraction $(f, g, \phi) = (\pi, incl, \phi)$ from $C(K)$ onto a chain subcomplex of it isomorphic to the homology of K. Reciprocally, given a chain contraction (f, g, ϕ) from $C(K)$ to its homology $H(K)$, then ϕ is a homology gvf.*

Let $incl$: Im$\pi \to C(K)$ be the inclusion map. Let $\pi = id_{\mathcal{C}(\mathcal{K})} - \partial\phi - \phi\partial$. This chain map describe for each cell a representative cycle of the homology class associated to this cell and satisfies that $\pi^2 = \pi$. If Im$\pi = \{x \in C(K),$ such that $x = \phi(y)$ for some $y\}$ and Ker$\pi = \{x \in C(K)$ such that $\phi(x) = 0\}$, then $\mathcal{C}(\mathcal{K}) =$ Im$\pi \oplus$ Kerπ). Let $f : C(K) \to$ Im(π) be the corestriction of π to Im(π) (that is, $\pi : C(K) \to$ Im(π)) and $g :$ Im$(\pi) \to C(K)$ be the inclusion. Let $\tilde{\partial}$ be the boundary operator of Im(π). We now prove that $\tilde{\partial} = 0$. Taking into account that $id_{C(K)} + gf = \phi\partial + \partial\phi$, $\partial\partial = 0$ and $\partial\phi\partial = \partial$, we then obtain $\partial - \partial gf = \partial$. Therefore, $\partial gf = g\tilde{\partial} f = 0$. Since f is onto and g is one-to-one, we deduce that $\tilde{\partial} = 0$. That means that the so-called *Morse complex associated to* ϕ, $M_{\phi,\partial} =$ Imπ is a graded vector space with null boundary operator isomorphic to the homology $H(K)$. ◇

The following proposition can be seen as an elementary boundary-perturbation result.

Proposition 2. *Let (K, ∂) be a finite cell complex and $\phi : C_*(X) \to C_{*+1}(X)$ be a homology gvf over K. If we consider a new boundary map $\partial + \delta$ for $C_*(X)$ (that is,$\delta : C_*(K) \to C_{*-1}(K)$ is a linear map satisfying $(\partial + \delta)^2 = 0$) such that $\phi\delta\phi = 0$, then a new chain contraction $(f_\delta, g, \phi) = (\pi + \phi\delta + \delta\phi, incl, \phi)$ from $C_*(K)$ onto the Morse complex $M_{\phi,\partial+\delta} =$ Im$(\pi + \phi\delta + \delta\phi)$ (having in general, a non-null boundary map $\delta + \partial\phi\delta + \delta\phi\partial + \delta\phi\delta$) can be established.*

It is clear that

Proposition 3. *In the conditions of Proposition 2 and if $\delta\phi = 0$, then the boundary map of $\mathcal{M}_{\phi,\partial+\delta}$ is $\delta + \partial\phi\delta$.*

Naming $\pi' = \pi + \phi\delta + \delta\phi$ and being in general $\phi\delta\phi \neq 0$, then we have that $id_{C(K)} + gf = \phi d + d\phi$ and, for an element $x = \pi'(y)$, such that $\pi'(y) \neq (\pi')^2(y)$ then

$$x + \pi'(x) = (\partial + \delta)\phi\pi'(y) + \phi\pi'(\partial + \delta)(y).$$

In fact, an algebraic integral operator (non necessarily a homology gvf) ϕ over K always determines a chain contraction from $C(K)$ to Imπ (Morse complex having in general a non-null boundary). If ϕ is an algebraic gvf which does not satisfy in general the condition $\phi d\phi = \phi$, then we have $id_{C(K)} + gf = \phi d + d\phi$ and, for an element $x = \pi(y)$, such that $\pi(y) \neq \pi^2(y)$ then

$$x + fg(x) = \partial\phi\pi(y) + \phi\pi\partial(y).$$

Now, we can easily prove the following gvf-perturbation result:

Proposition 4. *Let (K, ∂) be a finite cell complex and $\phi : C_*(K) \to C_{*+1}(K)$ be an algebraic gvf over K. If we consider a new algebraic gvf $\phi + \psi$ (that is, $\psi : C_*(K) \to C_{*+1}(K)$ is a linear satisfying that $(\phi + \psi)^2 = 0$) such that $(\phi + \psi) + (\phi + \psi)\partial(\phi + \psi) = 0$, then a new chain contraction $(f_\psi, g, \phi + \psi) = (\pi + \psi\partial + \partial\psi, incl, \phi)$ from $C_*(K)$ onto the Morse complex $\mathcal{M}_{\phi+\psi,\partial} = \mathrm{Im}(\pi + \psi\partial + \partial\psi)$ (having in general, a non-null boundary map $\partial + \partial(\phi + \psi)\partial$) can be established.*

It is time to analyze an incremental technique for getting gvfs, using the previous elementary perturbation steps. In this way, the incremental technique gives us a combinatorial gvf. Given a cell complex (K, ∂), the ordered set of cells $\mathcal{K} = \langle c_1, \ldots, c_m \rangle$ is a filter if all the faces of the cell c_j belong to the subset $\langle c_1, \ldots, c_{j-1} \rangle$. It is possible to "filter" K by first considering all the 0-cells in a certain order, then an order on all the 1-cells, and so on.

Algorithm 1. *Let (K, ∂) be a finite cell complex with filter $\mathcal{K}_m = \langle c_0, \ldots, c_m \rangle$. We represent the cell complex K up to filter level i by $\mathcal{K}_i = \langle c_0, \ldots, c_i \rangle$, with boundary map ∂_i. Let $\mathcal{M}_i$ be a chain complex associated to $\mathcal{K}_i$.*

```
$\mathcal{M}_0 := \{c_0\}$, $\phi_0(c_0) := 0$.
For $i = 1$ to $m$ do
    $\mathcal{M}_i := \mathcal{M}_{i-1} \cup \{c_i\}$, $\phi_i(c_i) := 0$,
    If $(\partial_i + \partial_{i-1}\phi_{i-1}\partial_i)(c_i) = 0$, then
       For $j = 0$ to $i - 1$ do,
            $\phi_i(c_j) := \phi_{i-1}(c_j)$.
    If $(\partial_i + \partial_{i-1}\phi_{i-1}\partial_i)(c_i)$ is
    a sum of a kind $\sum_{j=1}^{r} u_j \neq 0$ ($u_i \in \mathcal{M}_{i-1}$), then:
       Let us choose one $c^i_r$ from all the summands $u_k = f_i(c^i_k)$
       $= (1 + \phi_{i-1}\partial + \partial\phi_{i-1})(c^i_k)$ such that $\phi_{i-1}(c^i_r) = 0$
       and $c^i_r \notin$ Im$\phi_{i-1}$ and define $\phi_i(c^i_r) := c_i$
       and $\phi_i := \phi_{i-1}$ for the rest of elements of $\mathcal{M}_{i-1}$.

OUTPUT: a combinatorial gradient vector field $\phi_m$ for $K$.
```

The previous algorithm can be considered as an application of discrete Morse techniques [3]. The following figure shows in a pictorial way the result of the algorithm of a concrete cell complex. The gvf ϕ_m does not satisfy in general that $\phi_m \partial_m \phi_m = \phi_m$ and it is necessary to "correct" this fact in order to get a homology gvf.

Proposition 5. *Let* (K, ∂) *a finite cell complex and* $\phi : C_*(K) \to C_{*+1}(K)$ *a gvf such that* $\phi\partial\phi \neq \phi$*. Then, starting from* ϕ *it is possible to construct a chain contraction connecting* $C(K)$ *and* $Im\pi^n$ *(Morse cellular complex associated to the gvf* ϕ*), where* $\pi = 1 + \partial\phi + \phi\partial$*.*

Proof. If $\phi\partial\phi \neq \phi$ for a gvf ϕ over (K, ∂) , the translation of Forman's work [3] to the chain homotopy language is as follows:

$$1 + \pi = \phi\partial + \partial\phi$$

$$\pi + \pi^2 = \pi\phi\partial + \partial\pi\phi$$

$$\cdots\cdots$$

$$\pi^{n-1} + \pi^n = \pi^{n-1}\phi\partial + \partial\pi^{n-1}\phi$$

$$\pi^n + \pi^{n+1} = \pi^n\phi\partial + \partial\pi^n\phi$$

Let us suppose that $\pi^n = \pi^{n+1}$, that is, $\pi^n(1 + \pi) = 0$. This property, called stabilization by Forman, can be satisfied, for example if $(1 + \partial\phi)^n = 0$. Then, we get that

$$1 + \pi^n = (\sum \pi^k)\phi\partial + \partial(\sum \pi^k)\phi,$$

and it is not difficult to prove that $(\pi^n, incl, (\sum_{k=0}^{n} \pi^k)\phi)$ is a chain contraction connecting $C(K)$ and $\mathrm{Im}\pi^n$ (Morse cellular complex associated to the gvf ϕ).

3 Graph Representation of a Homology Gradient Vector Field

Using Discrete Morse Theory and its pictorial language, combinatorial gvfs can be described in terms of directed graphs on the cell complex. For example, let us take an integral operator ϕ such that $\phi(a) = c$, $a \in K_0$ and being a and b the vertices of the 1-cell c. It is clear that ϕ can be represented by a directed tree consisting in the edge c together with its vertices, such that the arrow on c goes out from vertex a. Of course, the previous properties of a homology gvf $\phi_i : C_i(K) \to C_{i+1}(K)$ $(i = 0, 1, 2)$ help us to suitably express all the ϕ_i in terms of graphs.

Proposition 6. *If* $\phi : C(K) \to C(K)$ *is a homology gvf for a cell complex* (K, ∂) *and we denote by* $H^{\partial}(K)$ *and* $H^{\phi}(K)$ *the homology groups of* K *taking respectively* ∂ *and* ϕ *as boundary maps on* K *(both satisfy the* 2*-nilpotency condition). Then,* $H^{\partial}(K)$ *and* $H^{\phi}(K)$ *are isomorph. The maps* $h : H^{\partial}(K) \to H^{\phi}(K)$ *defined by* $h([c]^{\partial}) = [c + \partial\phi(c)]^{\phi}$ *and* $k : H^{\phi}(K) \to H^{\partial}(K)$ *defined by* $h([c]^{\phi}) = [c + \phi\partial(c)]^{\phi}$ *specify this isomorphism.*

The following homology computation algorithm is given in[5]. It describes each incremental step as a composition of chain contractions and provides a easy way to prove that the final chain homotopy ϕ_m is strongly nilpotent.

Given a cell complex (K, ∂), the ordered set of cells $\mathcal{K} = \langle c_1, \ldots, c_m \rangle$ is a filter if c_i is a face of c_j for $i < j$. It is possible to "filter" K by first considering all the 0-cells in a certain order, then an order on all the 1-cells, and so on.

Algorithm 2. *Let (K, ∂) be a finite cell complex with filter $\mathcal{K}_m = \langle c_0, \ldots, c_m \rangle$. We represent the cell complex K up to filter level i by $\mathcal{K}_i = \langle c_0, \ldots, c_i \rangle$, with boundary map ∂_i. Let $\mathcal{H}_i$ the homology chain complex (with zero boundary map) associated to $\mathcal{K}_i$.*

```
$\mathcal{H}_0 := \{c_0\}$, $\phi_0(c_0) := 0$, $\pi_0(c_0) := c_0$.
For $i = 1$ to $m$ do
    $\pi_i(c_i) = \overline{c}_i = c_i + \phi_{i-1}\partial_i(c_i)$,
    $\mathcal{H}_i := \mathcal{H}_{i-1} \cup \{\overline{c}_i\}$, $\phi_i(c_i) := 0$,
    If $(\partial_i + \partial_{i-1}\phi_{i-1}\partial_i)(c_i) = 0$, then
       For $j = 0$ to $i - 1$ do,
           $\phi_i(c_j) := \phi_{i-1}(c_j)$.
    If $\pi_{i-1}\partial_i(c_i)$ is
    a sum of a kind $\sum_{j=1}^{r} \pi_{i-1}(e_{s_j}) = \sum_{j=1}^{r} u_j \neq 0$ ($u_i \in \mathcal{H}_{i-1}$), then:
       Let us choose a summand $u_k$ and define $\tilde{\phi}(u_k) := \overline{c}_i$
       and zero for the rest of elements of $\mathcal{H}_{i-1}$.
       then For $j = 0$ to $i - 1$ do,
                 $\phi_i(c_j) = (\phi_{i-1} + \tilde{\phi}\pi_{i-1})(c_j)$,
                 $\pi_i(c_j) = [1_{\mathcal{K}_i} - \phi_i\partial_i + \partial_i\phi_i](c_j)$
    $\mathcal{H}_i := \mathcal{H}_i \setminus \{u_k, \overline{c}_i\}$
OUTPUT: a homology gradient vector field $\phi_m$
       and the set of homology generators $\mathcal{H}_m$ for $K$.
```

The following result that can be easily proved using induction on i is the key for determining the graph nature of a homology gradient vector field

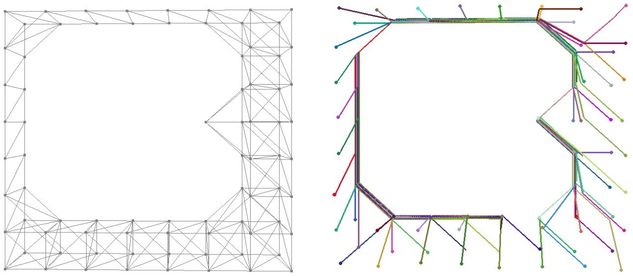


Fig. 1. Zoom of the polyhedral cell complex associated to a digital volume and its associated vertex tree

Proposition 7. *In the previous algorithm, the homology gradient vector field* ϕ_m *is strongly nilpotent and the map* $\pi_m = 1 + \partial\phi_m + \phi_m\partial$ *is strongly null over* ϕ_m.

Proposition 8. *Let* (K, ∂) *a finite cell complex and* ϕ *a strongly nilpotent homology gradient vector field on* K *with map* $\pi = 1 + \partial\phi + \phi\partial$ *being strongly null over* ϕ. *Then, the following properties hold:*

1. $\phi(u)$ *can be represented as a path from* u *to a fixed* $u_0 = (1 + \phi d + d\phi)(u)$.
2. *If* $\phi(u) = uv + \tilde{\phi}(u)$, *then* $\phi(v) = \tilde{\phi}(u)$.
3. *The graph of consisting of all the paths* $\phi(u)$, $u \in K_0$ *is a forest (as many trees as connected components the object has).*
4. *If* $\partial\phi(c - c') = 0$, *then* $\phi(c) = \phi(c')$, $\forall\, c, c' \in K_i, \;\; i = 0, 1, 2$.

In fact, these properties and the fact that Img has null homology guarantee that Imϕ in each level (levels 0 determined by the set $\phi(C_0(K))$, level 1 by $\phi(C_1(K))$ and level 2 by $\phi(C_2(K))$) can be represented as a kind of "cover" forest $(F(\phi)_0, F(\phi)_1, F(\phi)_2)$ for the cell complex K. From level 0 to 1, the 1-cells appearing as summands in $\phi(C_0(K))$ are the edges of the graph $F(\phi)_0$ and all the 0-cells are the vertices. In fact, $F(\phi)_0$ is a spanning forest of the graph defined by the vertices and edges of the cell complex K and it is called *vertex forest*. From level 1 to 2, the rest of 1-cells of K not involved in $F(\phi)_0$ are the vertices of the graph $F(\phi)_1$ (in fact, each edge is represented by its barycenter) and the 2-cells which are summands in $\phi(C_1(K))$ generates the edges of $F(\phi)_1$ (in fact, they join the "vertices" of the graph $F(\phi)_1$). The critical 1-cells (that is, the 1-cells u of K such that $\phi(u) = 0$)are the vertices of $F(\phi)_1$ which are isolated. If there is a 2-cell $v = \phi(u)$, for some 1-cell u such that the rest of 1-cells of its boundary belong to the set of edges of $F(\phi)_0$ or to the set of critical 1-cells, then v is represented in $F(\phi)_1$ by an edge connecting the vertex u with the barycenter of v.

From level 2 to 3, the rest of 2-cells of K not involved in $F(\phi)_1$ are the vertices of the graph $F(\phi)_2$ (each 2-cell is represented by its barycenter) and all the 3-cells which are summands in $\phi(C_2(K))$ generate the edges connecting them (if two 2-cells share a 3-cell, then these "vertices" are connected by an edge). The critical 2-cells (that is the 2-cells u of K such that $\phi(u) = 0$)are the vertices of $F(\phi)_2$ which are isolated. If there is a 3-cell $v = \phi(u)$, for some 2-cell u such that the rest of 2-cells of its boundary belong to the set of edges of $F(\phi)_1$ or to the set of critical 2-cells, then v is represented in $F(\phi)_2$ by an edge connecting the vertex u with the barycenter of v.

4 Trees Encoding the Connectivity of Digital Volumes

In [5], given a binary 26-adjacency voxel-based digital volume V, the homological information (that related to n-dimensional holes: connected components, "tunnels" and cavities) is extracted from a linear map (called homology gradient vector field) acting on a polyhedral cell complex $P(V)$ homologically equivalent

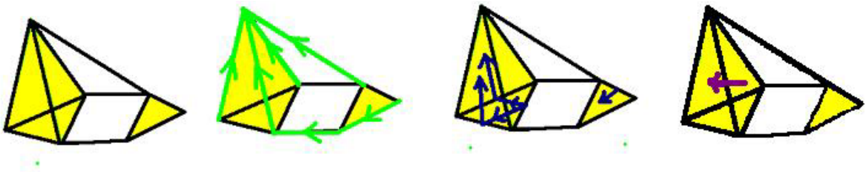


Fig. 2. A cell complex K and its corresponding forests $F(\phi)_0$, $F(\phi)_1$ and $F(\phi)_2$

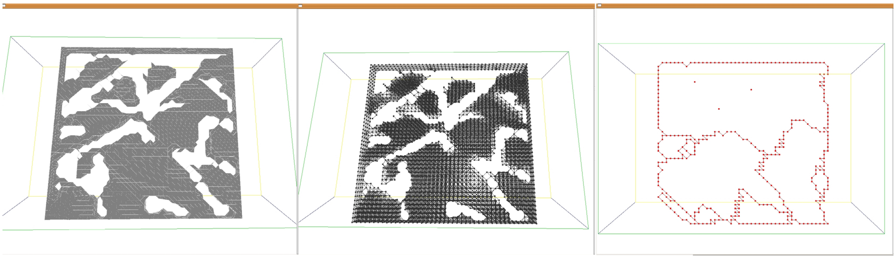


Fig. 3. Topological skeleton of a trabecular bone

to V. As we have seen in the previous sections, this map can be represented with no loss of information by three forests. This technique can be considered as an appropriate generalization of the spanning tree technique (arising from the graph theory) to three dimensions [13].

For a 3D binary digital image V, the choice of the filter and the choice of the pair of cells in each step of the Algorithm 2 allows us to see Reeb graphs, topological skeletons or other topological tools of V as results of pruning in the connectivity forests determining the homology gradient vector field.

References

1. Damiand, G., Peltier, S., Fuchs, L., Lienhardt, P.: Topological Map: An Efficient Tool to Compute Incrementally Topological Features on 3D Images. In: Reulke, R., Eckardt, U., Flach, B., Knauer, U., Polthier, K. (eds.) IWCIA 2006. LNCS, vol. 4040, pp. 1–15. Springer, Heidelberg (2006)
2. Delfinado, C.J.A., Edelsbrunner, H.: An Incremental Algorithm for Betti Numbers of Simplicial Complexes on the 3–Sphere. Comput. Aided Geom. Design 12, 771–784 (1995)
3. Forman, R.: A Discrete Morse Theory for Cell Complexes. In: Yau, S.T. (ed.) Geometry, Topology & Physics for Raoul Bott. International Press (1995)
4. Gonzalez-Diaz, R., Medrano, B., Real, P., Sanchez-Pelaez, J.: Algebraic Topological Analysis of Time-sequence of Digital Images. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) CASC 2005. LNCS, vol. 3718, pp. 208–219. Springer, Heidelberg (2005)
5. Molina-Abril, H., Real, P.: Advanced homological information of digital volumen via cell complexes. In: da Vitora Lobo, N., et al. (eds.) SSPR&SPR 2008. LNCS, vol. 5342, pp. 361–371. Springer, Heidelberg (2008)

6. Gonzalez-Diaz, R., Jimenez, M.J., Medrano, B., Molina-Abril, H., Real, P.: Integral Operators for Computing Homology Generators at Any Dimension. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 356–363. Springer, Heidelberg (2008)
7. Gonzalez-Diaz, R., Jiménez, M.J., Medrano, B., Real, P.: Chain Homotopies for Object Topological Representations. Discrete and Applied Mathematics 157, 490–499 (2009)
8. Gonzalez-Diaz, R., Real, P.: On the Cohomology of $3D$ Digital Images. Discrete Applied Math. 147, 245–263 (2005)
9. Hatcher, A.: Algebraic Topology. Cambridge University Press, Canbridge (2001)
10. Munkres, J.R.: Elements of Algebraic Topology. Addison-Wesley Co., Reading (1984)
11. Peltier, S., Alayrangues, S., Fuchs, L., Lachaud, J.O.: Computation of homology groups and generators. Computers & Graphics 30(1), 62–69 (2006)
12. Peltier, S., Ion, A., Haxhimusa, Y., Kropatsch, W.G., Damiand, G.: Computing Homology Group Generators of Images Using Irregular Graph Pyramids. In: Escolano, F., Vento, M. (eds.) GbRPR 2007. LNCS, vol. 4538, pp. 283–294. Springer, Heidelberg (2007)
13. Suuriniemi, S., Tarhasaari, T., Kettunen, L.: Generalization of the spanning-tree technique. IEEE Transactions on Magnetics 38(2), 525–528 (2002)
14. Taubin, G., Rossignag, J.: Geometric compression through topological surgery. ACM Transactions on Graphics 17(2), 84–115 (1998)

Energy-Based Perceptual Segmentation Using an Irregular Pyramid

R. Marfil and F. Sandoval

Grupo ISIS, Dpto. Tecnología Electrónica
University of Málaga, Campus de Teatinos 29071-Málaga (Spain)
rebeca@uma.es

Abstract. This paper implements a fast bottom-up approach for perceptual grouping. The proposal consists of two main stages: firstly, it detects the homogeneous blobs of the input image using a color-based distance and then, it hierarchically merges these blobs according to a set of energy functions. Both stages are performed over the Bounded Irregular Pyramid. The performance of the proposed algorithm has been quantitatively evaluated with respect to ground–truth segmentation data.

1 Introduction

Perceptual grouping can be defined as the process which allows to organize low–level image features into higher level relational structures. Handling such high–level features instead of image pixels offers several advantages such as the reduction of computational complexity of further processes like scene understanding. It also provides an intermediate level of description (shape, spatial relationships) for data, which is more suitable for object recognition tasks [1].

The simplest approach for perceptual grouping consists in grouping pixels into higher level structures based on low–level descriptors such as color or texture. However, these approaches cannot deal with natural, real objects as these are usually non–homogeneous patterns composed by different low–level descriptors. To solve this problem, the Gestalt principles can be applied to image perceptual segmentation [1]. As taking into account these principles to group pixels into higher level structures is computationally complex, perceptual segmentation approaches typically combine a pre–segmentation stage with a subsequent perceptual grouping stage. Basically, the pre–segmentation stage conducts the low–level definition of segmentation as a process of grouping pixels into homogeneous clusters, meanwhile the perceptual grouping stage performs a domain–independent grouping which is mainly based on properties like the proximity, similarity, closure or continuity. Although the final regions obtained by these approaches do not always correspond to the natural image objects, they provides a mid-level segmentation which is more coherent with the human-based image decomposition.

This paper presents a hierarchical perceptual segmentation approach which is divided into these two aforementioned stages. The pre–segmentation stage

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 424–431, 2009.
© Springer-Verlag Berlin Heidelberg 2009

groups the image pixels into a set of homogeneous regions or blobs whose spatial distribution is physically representative of the image content. The size of this set is commonly very much less than the original number of pixels. Thus, these blobs constitute an efficient image representation that replace the pixel–based image representation. Besides, they preserve the image geometric structure as each significant feature contain at least one region. Our approach accomplishes this pre-segmentation stage by means of an irregular pyramid: the Bounded Irregular Pyramid (BIP) [2,3]. The second stage groups the set of blobs into a smaller set of regions taking into account a set of features derived from the Gestalt theory. Specifically, this grouping is based on five properties: proximity, similarity, closure, continuity and symmetry. From these properties, Luo and Guo [4] proposed to calculate several energy functions for any region or group of regions. In order to find the lowest energy groupings, these functions have been used in this paper to measure the cost of a grouping. However, contrary to the original work of Luo and Guo [4], our approach can simultaneously merge in one grouping any number of regions instead to group only pair–wise regions. For this purpose, the BIP structure is used. The perceptual grouping is performed by adding new levels to the pre–segmentation pyramid. In these new levels, the decimation process will use the energy functions to determine if two nodes must be grouped. The rest of this paper is organized as follows: Sections 2 and 3 present the proposed segmentation approach. Experimental results revealing the efficacy of the method are described in Section 4. The paper concludes along with discussions and future work in Section 5.

2 Proposed Irregular Pyramid

The data structure of the Bounded Irregular Pyramid (BIP) is a mixture of regular and irregular data structures: a 2x2/4 regular structure and a simple graph. This data structure generates an irregular configuration which is described as a graph hierarchy. In this hierarchy, there are two types of nodes: nodes belonging to the 2x2/4 structure, named *regular nodes* and *irregular nodes* or nodes belonging to the irregular structure. In general, a node $\mathbf{x}$ is neighbor of other node $\mathbf{x}'$ if their reduction windows (the set of nodes with are linked to them in the level below) $w_{\mathbf{x}}$ and $w_{\mathbf{x}'}$ are connected. Two reduction windows are connected if there are at least two nodes at level l-1, $\mathbf{y} \in w_{\mathbf{x}}$ and $\mathbf{y}' \in w_{\mathbf{x}'}$, which are neighbors. Two nodes $\mathbf{x}_1$ and $\mathbf{x}_2$ which are neighbors at level l are connected by an intra-level arc $e = (\mathbf{x}_1, \mathbf{x}_2) \in E_l$.

The BIP arose as a mixture of regular and irregular structures whose goal is to obtain accurate results at a low computational cost. However, experimental results have shown that, although computationally efficient, the BIP presents a shift-variance value higher than the rest of the most important irregular approaches [2,3]. In this paper, the original BIP decimation strategy has been modified to increase the degree of mixture between the regular and irregular parts of the BIP data structure. This modification significantly reduces the shift-variance problem without an increase of the computational cost.

Let $G_l = (N_l, E_l)$ be a graph where N_l stands for the set nodes and E_l for the set of intra-level arcs. Let $\varepsilon_l^{\mathbf{xy}}$ be equal to 1 if $(\mathbf{x}, \mathbf{y}) \in E_l$ and equal to 0 otherwise. Let $\xi_{\mathbf{x}}$ be the neighborhood of the node $\mathbf{x}$ defined as $\{\mathbf{y} \in N_l : \varepsilon_l^{\mathbf{xy}}\}$. It can be noted that a given node $\mathbf{x}$ is not a member of its neighborhood, which can be composed by regular and irregular nodes. Each node $\mathbf{x}$ has associated a $v_{\mathbf{x}}$ value. Besides, each regular node has associated a boolean value $h_{\mathbf{x}}$: the homogeneity [3].Only regular nodes which have $h_{\mathbf{x}}$ equal to 1 are considered to be part of the regular structure. At the base level of the hierarchy G_0, all nodes are regular, and they have $h_{\mathbf{x}}$ equal to 1. The proposed decimation process transforms the graph G_l in G_{l+1} such that the image is divided into a set of homogeneous blobs. This aim is achieved using a pairwise comparison function, $g(v_{\mathbf{x_1}}, v_{\mathbf{x_2}})$, which is true if the $v_{\mathbf{x}_1}$ and $v_{\mathbf{x}_2}$ values associated to the $\mathbf{x}_1$ and $\mathbf{x}_2$ nodes are similar according to some criteria and false otherwise.

The proposed decimation algorithm runs two consecutive steps to obtain the set of nodes N_{l+1}. The first step generates the set of regular nodes of G_{l+1} from the regular nodes at G_l and the second one determines the set of irregular nodes at level l+1. Contrary to previously proposed approaches [2,3], this second process employs a union-find process which is simultaneously conducted over the set of regular and irregular nodes of G_l which do not present a parent in the upper level $l + 1$. The decimation process consists of the following steps:

1. Regular decimation process. The $h_{\mathbf{x}}$ value of a regular node $\mathbf{x}$ at level l+1 is set to 1 if

$$\{\bigcap_{\forall \mathbf{y}_j, \mathbf{y}_k \in \{\mathbf{y}_i\}} g(v_{\mathbf{y}_j}, v_{\mathbf{y}_k})\} \cap \{\bigcap_{\mathbf{y}_j \in \{\mathbf{y}_i\}} h_{\mathbf{y}_j}\} \tag{1}$$

   Besides, at this step, inter-level arcs among regular nodes at levels l and l+1 are established. If $\mathbf{x}$ is an homogeneous regular node at level l+1 ($h_{\mathbf{x}}$==1), then the set of four nodes immediately underneath $\{\mathbf{y_i}\}$ are linked to $\mathbf{x}$.
2. Irregular decimation process. Each irregular or regular node $\mathbf{x} \in N_l$ without parent at level l+1 chooses the closest neighbor $\mathbf{y}$ according to the $v_{\mathbf{x}}$ value. Besides, this node $\mathbf{y}$ must be similar to $\mathbf{x}$. That is, the node $\mathbf{y}$ must satisfy

$$\{||v_{\mathbf{x}} - v_{\mathbf{y}}|| = \min(||v_{\mathbf{x}} - v_{\mathbf{z}}|| : \mathbf{z} \in \xi_{\mathbf{x}})\} \cap \{g(v_{\mathbf{x}}, v_{\mathbf{y}})\} \tag{2}$$

   If this condition is not satisfy by any node, then a new node $\mathbf{x}'$ is generated at level l+1. This node will be the parent node of $\mathbf{x}$. Besides, it will constitute a root node and its receptive field at base level will be an homogeneous set of pixels according to the specific criteria. On the other hand, if $\mathbf{y}$ exists and it has a parent $\mathbf{z}$ at level l+1, then $\mathbf{x}$ is also linked to $\mathbf{z}$. If $\mathbf{y}$ exists but it does not have a parent at level l+1, a new irregular node $\mathbf{z}'$ is generated at level l+1. In this case, the nodes $\mathbf{x}$ and $\mathbf{y}$ are linked to $\mathbf{z}'$.

   This process is sequentially performed and, when it finishes, each node of G_l is linked to its parent node in G_{l+1}. That is, a partition of N_l is defined. It must be noted that this process constitutes an implementation of the union-find strategy.

3. Definition of intra-level arcs. The set of edges E_{l+1} is obtained by defining the neighborhood relationships between the nodes N_{l+1}. Two nodes at level $l+1$ are neighbors if their reduction windows are connected at level l.

When the decimation scheme proposed in [2,3] is used, regular and irregular nodes of level l cannot be linked to the same parent node at level $l+1$. This provokes that the image partition varies when the base of the pyramid is shifted slightly or rotated. In this case, the decimation process has been simplified, allowing that regular and irregular nodes will be grouped. Both decimation schemes have been compared using the Shift Variance test proposed by Prewer and Kitchen [2] in a color-based segmentation framework. Average values obtained from 30 color images from the Waterloo and Coil 100 databases are equal to 73.8 for the decimation process proposed in [2,3] and 43.7 for the new decimation process.

3 Segmentation Approach

3.1 Pre–segmentation Stage

The pre–segmentation stage groups the image pixels into a set of photometric homogeneous regions (blobs) whose spatial distribution is physically representative of the image content. This stage is accomplished by means of the irregular pyramid described in Section 2, taking into account the CIELab color of the image pixels. Let $\mathcal{I} \subset \Re^2$ be the domain of definition of the input image and $l \in \Re$ a level of the hierarchy, the pre–segmentation assigns a partition $\mathcal{P}_l$ to the couple $(\mathcal{I}, l)$ which is defined by the receptive fields associated to the nodes at level l. In this process, the effective set of pyramid levels is restricted to the interval $[0, l_m]$. At $l = 0$, each pixel of the input image is an individual blob of the partition $\mathcal{P}_0$. If $l \geq l_m$, then the partition $\mathcal{P}_l$ is equal to $\mathcal{P}_{l_m}$. That is, the structure hierarchy stops growing at a certain level l_m when it is no longer possible to link together any more nodes because they are not similar. Besides, the following property is satisfied [11]:

$$l \leq l' \Rightarrow \mathcal{P}_l \subseteq \mathcal{P}_{l'} \Leftrightarrow \forall p \in \mathcal{P}_l, \exists p' \in \mathcal{P}_{l'} : p \subseteq p' \tag{3}$$

Equation (3) determines that partitions at different levels are nested and they impose a hierarchical structure to the set of image blobs $\mathcal{B} = \{b \subset \mathcal{I} \mid \exists l : b \in \mathcal{P}_l\}$. This hierarchy can be naturally combined with homogeneity criteria. That is, if a consistent homogeneity criteria is defined over the set of partitions as a boolean function $\mathcal{H}$ over all its elements, the following property is verified [10]

$$\forall (p \in \mathcal{P}_l, p' \in \mathcal{P}_{l'}) \;\; p \subset p' \Rightarrow (\mathcal{H}(p') \Rightarrow \mathcal{H}(p)) \tag{4}$$

This property states that the elements that constitute a homogeneous blob are also homogeneous. In our approach, the aim is to find partitions $\mathcal{P}_l = \{p_l^i\}_{i=1}^n$ such that these components will define blobs of uniform color at the hierarchy base level. Each component p_l^i is defined by the receptive field associated to a

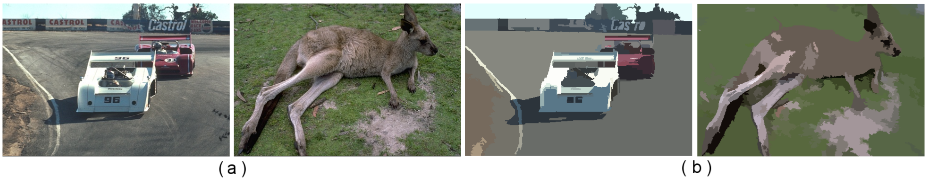


Fig. 1. Pre-segmentation stage: a) Original images; and b) colorized pre-segmentation images

node $\mathbf{x}_i^{(l)}$ at level l. To find these partitions, we use the pairwise comparison of neighboring nodes: $g(v_{\mathbf{x}_\mathbf{i}^{(1)}}, v_{\mathbf{x}_\mathbf{j}^{(1)}})$ is true if the color difference between $\mathbf{x}_i^{(l)}$ and $\mathbf{x}_j^{(l)}$ is under a given threshold U_v, and false otherwise. Hence, this is a boolean comparison function for pairs of partitions. In order to measure the color difference between two nodes, the $v_\mathbf{x}$ value of each node stores the mean color of its receptive field and the Euclidean distance is employed. That is, the distance between two nodes $\mathbf{x}_i^{(l)}$ and $\mathbf{x}_j^{(l)}$ at the pre–segmentation stage is defined as

$$\psi^\beta = \sqrt{(L_{\mathbf{x}_i^{(l)}} - L_{\mathbf{x}_j^{(l)}})^2 + \beta(a_{\mathbf{x}_i^{(l)}} - a_{\mathbf{x}_j^{(l)}})^2 + \beta(b_{\mathbf{x}_i^{(l)}} - b_{\mathbf{x}_j^{(l)}})^2} \tag{5}$$

where β is a parameter which allows to weight lightness and chrominance (grey scale images correspond to β equal to 0 and the usual CIELab space to β equal to 1). The final pre–segmentation image is provided by the partition $\mathcal{P}_{l_m}$, which will be associated to the nodes at level l_m. It must be noted that all nodes $\mathbf{x}_i^{(l_m)} \in N_{l_m}$ are root nodes. Fig. 1 shows two colorized pre–segmentation images.

3.2 Perceptual Grouping Stage

After the local similarity pre–segmentation stage, grouping regions aims at simplifying the content of the obtained partition. For managing this grouping, the irregular structure is used: the roots of the pre–segmented blobs at level l_m constitute the first level of the perceptual grouping multiresolution output. Successive levels can be built using the decimation scheme described in Section 2. Let $\mathcal{P}_{l_m}$ be the image partition provided by the pre–segmentation stage and $l > l_m \in \Re$ a level of the hierarchy, this second stage assigns a partition $\mathcal{Q}_l$ to the couple $(\mathcal{P}_{l_m}, l)$, satisfying that $\mathcal{Q}_{l_m}$ is equal to $\mathcal{P}_{l_m}$ and that

$$\exists l_n \in \Re^+ \quad : \quad \mathcal{Q}_l = \mathcal{Q}_{l_n}, \quad \forall l \geq l_n \tag{6}$$

The family of obtained partitions $\mathcal{Q}_l$ also satisfies the conditions expressed by Eqs. (3) and (4). However, in order to achieve the perceptual grouping process, a similarity function must be defined.

In the Luo and Guo's proposal [4], a set of energy functions was used to characterize desired single–region properties and pairwise region properties. The single-region properties include region area, region convexity, region compactness, and color variances in one region. The pairwise properties include the color mean differences between two regions, the edge strength along the shared boundary, the color variance of the cross–boundary area, and the contour continuity between two regions.

With the aim of finding the lowest energy groupings, Huart and Bertolino [12] propose to employ these energies to measure the cost of any region or group of regions. In a similar way, we have defined a pairwise comparison function to evaluate if two nodes can be grouped. Two energies are defined:

$$E_{fusion} = E_{area} + E_{compactness} + E_{varianceL} + E_{varianceA} + E_{varianceB} \quad (7)$$

$$\begin{gathered} E_{region} = E_{colorMeanDiffL} + E_{colorMeanDiffA} + E_{colorMeanDiffB} + \\ E_{BoundaryVarianceL} + E_{BoundaryVarianceA} + E_{BoundaryVarianceB} + E_{edge} \end{gathered} \quad (8)$$

where the energy functions have been taken from [4]. E_{fusion} and E_{region} are used to measure the cost of the fusion operation and the energy of a region resulting from the fusion, respectively. If $E_{fusion} + E_{region}$ is less than a given threshold U_c, the comparison function is true and the grouping is accepted. Otherwise, the function is false. Fig. 2 shows the final segmentation images associated to the images at Fig. 1a.

4 Experimental Results

In order to evaluate the performance of the proposed color image segmentation approach, the BSDB has been employed[1] [6]. In this dataset, the methodology for evaluating the performance of segmentation techniques is based in the comparison of machine detected boundaries with respect to human-marked boundaries using the *Precision-Recall framework* [5]. This technique considers two quality measures: precision and recall. The *precision* (P) is defined as the fraction of boundary detections that are true positives rather than false positives. Thus, it quantifies the amount of noise in the output of the boundaries detector approach. On the other hand, the *recall* (R) is defined by the fraction of true positives that are detected rather than missed. Then, it quantifies the amount of ground truth detected. Measuring these descriptors over a set of images for different thresholds of the approach provides a parametric Precision-Recall curve. The current public version of the data set is divided in a training set of 200 images and a test set of 100 images. In order to ensure the integrity of the evaluation, only the images and segmentation results from the training set can be accessed during the optimization phase. Fig. 2 shows the partitions on the higher level of the hierarchy for several images. The optimal training parameters have been chosen. It can be noted that the proposed approach is able to group perceptually important regions in spite of the large intensity variability presented on several areas of the input images. The pre–segmentation stage provides an over-segmentation

---

[1] http://www.cs.berkeley.edu/projects/vision/grouping/segbench/

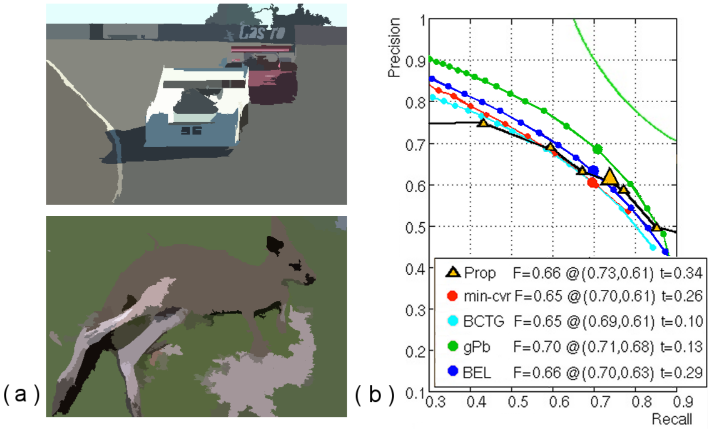


Fig. 2. a) Segmentation images associated to input images at Fig. 1a, and b)Comparison of the proposed approach with other methods on color images using the BSDB protocol (see text)

of the image which overcomes the problem of noisy pixels [10], although bigger details are preserved in the final segmentation results.

The F–measure is defined as the harmonic mean of the presicion and recall quality measures. The F-measure associated to the individual results ranged from bad to significantly good values. Thus, the F-measure value of all images in Fig. 2 is over 0.8. On the contrary, the main problems of the proposed approach are due to its inability to deal with textured regions which are defined at high natural scales. In any case, the maximal F–measure obtained from the whole test set is 0.66 (see Fig. 2). This value is equal than the F–measure obtained by the BEL [7] and greater than values provided by other methods like the BCTG [5] or the min–cover [9]. However, it must be noted that the obtained value is less than values provided by the gPb [8]. The *Global Probability of Boundary* (gPb) is a recently proposed boundary detector which combines local information derived from low–level image features, such as brightness, color and texture, with global information obtained from spectral partitioning. The main disadvantage of this method is its associated high computational costs [8]. On the contrary, the pre–segmentation stage of the proposed approach runs very fast. Thus, using an Intel Core Duo T7100 PC with 1Gb DDR2 of memory, the processing times associated to the pre-segmentation stage are typically less than 250 ms to process any image on the test set. Although the processing times provided by the perceptual grouping stage are greater, the average processing time of the segmentation approach is less than 2 seconds per image for the whole test set.

5 Conclusions and Future Work

This paper has presented a generic approach which combines an accurate segmentation process that takes into account the color information of the image, with a grouping process that merges blobs to produce regions that are perceptually relevant. The obtained result is a stack of partitions which presents a very few regions at the high level.

Future work will be focused on studying the interactions of the different energies to determine the relevance of each of them according to the local neighborhood. We also aim at employing a texture descriptor at the pre–segmentation stage and at processing videos to add the temporal information to this process. It could provide additional merging information or be used to query the stack of partitions.

Acknowledgments

This work has been partially granted by the Spanish Junta de Andalucía, under projects P07-TIC-03106 and P06-TIC-02123.

References

1. Zlatoff, N., Tellez, B., Baskurt, A.: Combining local belief from low-level primitives for perceptual grouping. Pattern Recognition 41, 1215–1229 (2008)
2. Marfil, R., Molina-Tanco, L., Bandera, A., Rodríguez, J.A., Sandoval, F.: Pyramid segmentation algorithms revisited. Pattern Recognition 39(8), 1430–1451 (2006)
3. Marfil, R., Molina-Tanco, L., Bandera, A., Sandoval, F.: The construction of bounded irregular pyramids with a union-find decimation process. In: Escolano, F., Vento, M. (eds.) GbRPR 2007. LNCS, vol. 4538, pp. 307–318. Springer, Heidelberg (2007)
4. Luo, J., Guo, C.: Perceptual grouping of segmented regions in color images. Pattern Recognition, 2781–2792 (2003)
5. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using brightness, color, and texture cues. IEEE Trans. on Pattern Analysis Machine Intell. 26(1), 1–20 (2004)
6. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. Int. Conf. Computer Vision (2001)
7. Dollár, P., Tu, Z., Belongie, S.: Supervised learning of edges and object boundaries. In: Proc. Int. Conf. Computer Vision Pattern Recognition (2006)
8. Maire, M., Arbeláez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: Int. Conf. Computer Vision Pattern Recognition (2008)
9. Felzenszwalb, P., McAllester, D.: A min-cover approach for finding salient curves. In: POCV 2006 (2006)
10. Haxhimusa, Y., Glantz, R., Kropatsch, W.G.: Constructing stochastic pyramids by MIDES - maximal independent directed edge set. In: Hancock, E.R., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, pp. 35–46. Springer, Heidelberg (2003)
11. Arbeláez, P., Cohen, L.: A metric approach to vector-valued image segmentation. Int. Journal of Computer Vision 69, 119–126 (2006)
12. Huart, J., Bertolino, P.: Similarity-based and perception-based image segmentation. In: IEEE Int. Conf. on Image Processing (2005)

Hierarchical Graphs for Data Clustering

E.J. Palomo[1], J.M. Ortiz-de-Lazcano-Lobato[1], Domingo López-Rodríguez[2], and R.M. Luque[1]

[1] Department of Computer Science
E.T.S.I.Informatica, University of Malaga
Campus Teatinos s/n, 29071 – Malaga, Spain
{ejpalomo,jmortiz,rmluque}@lcc.uma.es
[2] Department of Applied Mathematics
E.T.S.I.Informatica, University of Malaga
Campus Teatinos s/n, 29071 – Malaga, Spain
dlopez@ctima.uma.es

Abstract. The self-organizing map (SOM) has been used in multiple areas and constitutes an excellent tool for data mining. However, SOM has two main drawbacks: the static architecture and the lack of representation of hierarchical relations among input data. The growing hierarchical SOM (GHSOM) was proposed in order to face these difficulties. The network architecture is adapted during the learning process and provides an intuitive representation of the hierarchical relations of the data. Some limitations of this model are the static topology of the maps (2-D grids) and the big amount of neurons created without necessity. A growing hierarchical self-organizing graph (GHSOG) based on the GHSOM is presented. The maps are graphs instead of 2-D rectangular grids, where the neurons are considered the vertices, and each edge of the graph represents a neighborhood relation between neurons. This new approach provides greater plasticity and a more flexible architecture, where the neurons arrangement is not restricted to a fixed topology, achieving a more faithfully data representation. The proposed neural model has been used to build an Intrusion Detection Systems (IDS), where experimental results confirm its good performance.

Keywords: Data clustering, self-organization, graph-based representation, hierarchical clustering, intrusion detection systems.

1 Introduction

Data clustering is an unsupervised learning method to find an optimal partitioning of the data set. This partitioning organizes the input data into clusters according to a similarity measure, where data belonging to one cluster are more similar than data belonging to different clusters. Generally speaking, unsupervised learning methods are specially useful when we have unlabeled input data, that is, the groups in which data are classified are unknown. The purpose of these methods is to discover groups in a data set based on similarity, where input data are usually represented as feature vectors of high dimensionality.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 432–439, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The self-organizing map (SOM) is a widely used unsupervised neural network for clustering high-dimensional input data and mapping these data into a two-dimensional representation space [1]. However, it has some drawbacks. To begin with, the number and arrangement of neurons (network architecture) of the SOM is static and has to be established in advance. This task can be difficult because it is needed a prior study of the problem domain, specially when we have vectors with many features. Also, inherent hierarchical relations among input data are not represented on the map and can make difficult the interpretation of the data.

The growing hierarchical SOM (GHSOM) tries to face these problems derived from SOM. The GHSOM has a hierarchical architecture arranged in layers, where each layer is composed of different growing SOMs expanded from neurons of the upper layer maps and the number of neurons of each map are adaptively determined [2]. This way, the architecture of the GHSOM is established during the unsupervised learning process according to the input patterns. However, this neural model keep showing some difficulties related to its static topology. In fact, each map is initiated with 2x2 neurons, forcing the map to be a 2-D rectangular grid. The problem of this topology is that we need to preserve the rectangular grid and when the map grows, not only a neuron but a row or a column of neurons has to be inserted. As a result, a lot of neurons are inserted without necessity, being the amount of neurons far from optimal, especially when we have to insert neurons in very large maps. Moreover, there are wide spectrum application domains where a rectangular grid is not the most suitable topology to represent the input data. In order to make easy the identification of relations among input data and mirror its inherent structure, the map should capture the own topology of the data as faithfully as possible.

A graph-based representation space provides the enough flexibility to reflect different representations of the same domain (data and relations) and is easy to understand. Most of the related works which use SOMs with a graph-based representation, treat each neuron as a graph since they use this model for representing input data [3,4]. The goal is to cluster these graphs. Here the goal is different; we use graphs to represent the topology of a SOM. Thus, the representation space can totally adapt to the data structure.

In this paper, by using this graph-based representation in hierarchical self-organization, we propose a novel artificial neural network architecture based on the GHSOM called the growing hierarchical self-organization graph (GHSOG). This new model has the advantages of the GHSOM model and also provides a better and easy data representation thanks to a more flexible and adaptive topology and a lower number of neurons used in the representation.

In order to check the performance of the GHSOG, an Intrusion Detection System (IDS) has been implemented using this new model. An IDS monitors network traffic to detect an attack or intrusion in a network environment. Some anomaly detection systems using data mining techniques such as clustering, support vector machines (SVM) and neural network systems have been proposed [5,6]. Many self-organizing models have been used to implement an IDS, however they have many difficulties detecting a wide variety of attacks with low false positive rates [7].

The remainder of this paper is organized as follows. In Section 2, we first present a description of our new GHSOG model. In Section 3, an IDS is implemented with the GHSOG model, where data preparation and some experimental results are presented. Some remarks conclude this paper in Section 4.

2 Growing Hierarchical Self-Organizing Graph

The Growing Hierarchical Self-Organizing Graph is a neural network based on the GHSOM, where only one neuron is added when a self-organizing map should be expanded. This way, the topology that the tree nodes present is a graph instead of a grid. Initially, a segment composed of two neurons and a connection edge is used to represent a cluster. If more neurons are needed to capture the cluster knowledge accurately, then these neurons are added one by one, ensuring that each new neuron forms a triangle tile in the graph. See Figure 1.

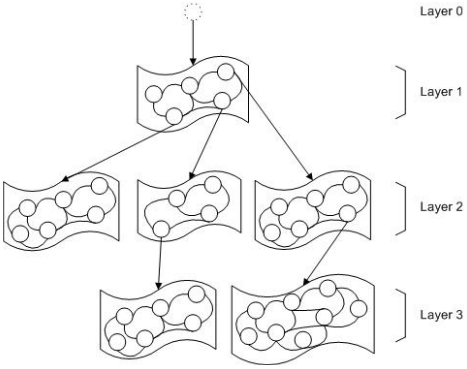


Fig. 1. A sample of the GHSOG model

Let $X = x_1, x_2, \ldots, x_L$ the training data set consisting of L samples, the starting point of the training algorithm is the computation of the *quantization error* of the whole training data set

$$qe_0 = \sum_{x_j \in X} \|w_0 - x_j\| \tag{1}$$

with w_0 the global mean. Using the quantization error instead of the mean quantization error gives the method the capacity of assigning more neurons in the map for more densely populated regions of the input space. The first self-organizing map is created at layer 1, i.e., at depth level 1 in the tree. This map consists of two neurons which are linked by a unique connection.

Then for each local graph, i.e. each self-organizing graph, a two phase learning process begins. The first phase carries out the training (and growing if required) of the local self-organizing graphs; the second one tries to expand that neurons which required a more thorough clustering of the data they represent. For that purpose a new graph in the next tree layer is created.

In the *local training and growing phase* each local graph is trained following the learning process proposed by Kononen in [1] except for the neighborhood function. In the proposed network it is considered that the neighbors of a neuron are all the neurons which are connected directly in the graph.

It should be noticed that the samples used to train each local graph are those belonging to the cluster corresponding to the father neuron in the tree, i.e., if f is the father neuron then the training data set for the child graph is $C_f \subseteq X$ where C_f is the subset of vectors of the input data that is mapped onto unit f.

After training the local graph, the learning algorithm checks if the graph must grow.

$$\frac{1}{|U_f|} \sum_{i \in U_f} qe_i < \tau_1 \cdot qe_f \tag{2}$$

where $qe_i = \sum_{x_j \in C_f} \|w_i - x_j\|$ is the absolute quantification error and U_f the neurons in the hierarchical tree which are part of the local graph, expanded from neuron f. The expression compares the mean quantization error of the graph with the quantization error of the corresponding neuron f in the upper layer. The parameter τ_1 allow the user to control the depth/shallowness of the obtained hierarchical structure. If the criterion is not satisfied then a new neuron is added and this phase starts again. This iterative process finishes when the expression holds true.

When the mean quantization error of the graph is too high and does not fulfill the stopping criterion, a new neuron is added to the graph in order to reduce that error. Therefore, we look for the neuron of the graph which contributes most to the quantization error. This neuron is named the *error neuron*

$$e = \arg\max_i \sum_{x_j \in C_i} \|x_j - w_i\| \tag{3}$$

A second neuron is needed to keep the triangle tile based structure of the local graphs. The selected neuron will be the most dissimilar neighboring neuron d of the error neuron e

$$d = \arg\max_i (w_e - w_i), w_i \in N_e \tag{4}$$

with N_e the set of neighbours neurons of the error neuron e. Finally the location of the new neuron in the input space, which coincides with the weight vector of the neuron is

$$w_{new} = \frac{qe_e \cdot w_e + qe_d \cdot w_d}{qe_e + qe_d} \tag{5}$$

Note that the most dissimilar neuron d of the error neuron e is the farthest of its neighbors and, therefore, most of the time the new neurons are placed near the error neuron, which is the region in the input space where the quantization error is greater and help is needed.

Once the first iterative phase has finished, the *neurons expansion phase* starts. In this step for each neuron of the graph we compute the checking

$$qe_i = \tau_2 \cdot qe_0 \tag{6}$$

where τ_2 is a user parameter which controls the desired quality for the data representation. If the expression holds true for all the neurons the training of the current local graph finishes and the process continues with another local graph (in the same layer of the tree or in deeper layers). Otherwise for each neuron which does not satisfies the criterion a new graph is created in the next layer of the tree. The new graph is a graph consisting of two neurons, whose weigth vectors are the average between the father vector and the first and second more similar neighbor vectors, respectively. If the father has only one neighbor, the second expanded neuron is a copy of the father.

3 Experimental Results

In order to prove the performance of the new GHSOG model, an IDS has been implemented, which detects a wide variety of attacks in a military network environment. GHSOG training and testing have been done with the KDD Cup 1999 benchmark data set created by MIT Lincoln laboratory. The purpose of this benchmark was to build a network intrusion detector capable of distinguishing between intrusions or attacks, and normal connections. To make easy the training and testing, the 10% training data set and the 10% testing data set are provided, which contain 494,021 and 311,029 connection records, respectively. In the training data set there are 22 attack types in addition to normal records whereas in the testing data set 15 new attack types are also present.

3.1 Data Preparation

In the KDD 1999 data set, each connection record has 41 features preprocessed from monitoring network packets. Among these features, 3 are qualitative: the protocol type, which can take the values TCP, UDP or ICMP; the service type, with 66 different values and the flag, with 11 possible values.

Most of the related works map these data into consecutive quantitative values so as to compute the similarities of the connections. However, since qualitative data have no an order associated, the use of the Euclidean distance is not appropriate for these types of data. To solve this problem, each qualitative feature has been replaced with a binary vector composed by as many binary features as different possible values that feature can take. Thus, the distance among qualitative values is always the same allowing the use of the Euclidean distance or other standard metric. This way, the number of features has been increased from 41 to 118.

3.2 Results

For our experiments, two data subsets have been randomly selected from the 10% training data set: S1 with 100,000 connection records and S2 with 169,000 connection records. The 22 attack types in addition to normal records are contained in both data subsets, and the distribution of each connection type reflects the own distribution of the 10% training data set. In order to train two neural networks with the two data subsets, 0.1 and 0.03 have been chosen as values for parameters τ_1 and τ_2, respectively. The resulting architecture of the first neural networks trained with S1 is shown in Fig. 1. The second one has 1 layer more, but with just a graph in that layer.

Most of the related works are only interested in knowing whether the input pattern is an anomaly or a normal connection record. Here, our interest is also to know whether one connection record type is identified as from its own type. We name this identification rate. Detection rate is the percentage of detected attacks and false positive rate is the percentage of normal connections records detected as attacks. The training results for S1 and S2 are given in Table 1. We can see that the detection rate is around the 96% for both data sets, but the false positive rate is higher for the first data subset.

Table 1. Training results for S1 and S2 data subsets

Training Set	Detection (%)	False Positive (%)	Identification (%)
S1	96.75	3.81	94.64
S2	96.44	1.42	95.40

Table 2. Testing results for GHSOGs trained with S1 and S2 data subsets

Training Set	Detection (%)	False Positive (%)	Identification (%)
S1	90.15	2.09	90.68
S2	91.48	4.51	90.57

After training, the two trained GHSOGs have been tested with the 10% KDD 1999 testing data set, which contains 311,029 connection records and 15 new attack types as mentioned above. Testing results are shown in Table 2. During the testing, we achieve similar detection rates for both GHSOGs but the false positive rate is higher in the second GHSOG. Note that the identification rate is higher than the detection rate in the first GHSOG. Although it can seem strange, this is due to the fact that normal connections are taken into account in addition to attack connections to compute the identification rate.

There are many related works that have used self-organizing maps to implement an IDS. A hierarchical Kohonen net (K-Map) was proposed in [8], which consists of three static SOMs arranged in layers. They achieved 99.63% detection rate after testing, but taking into account several limitations. Although they

Table 3. Testing results for differents IDSs based on self-organization

	Detected(%)	False Positive(%)	Neurons
GHSOG	90.68	2.09	35
K-Map	99.63	0.34	144
SOM	97.31	0.042	400
SOM (DoS)	99.81	0.1	28800

used a data subset of 169,000 connection records with 22 attack types from the KDD Cup 1999 data set as we used, during the testing they used just 3 attack types, whereas we used 38 attack types, where 15 attack types were unknown. They also used a combination of 20 features that had to be established in advance and 48 neurons in each layer. In [9], a SOM trained on all the 41 features was chosen in order to compare results. A 97.31% detection rate was achieved but using 400 neurons. An emergent SOM (ESOM) for the Intrusion Detection process was proposed in [10]. They achieved detection rates between 98.3% and 99.81% and false positives between 2.9% and 0.1%. However, they just detected denial of service attacks (DoS), they used a pre-selected subset of 9 features and a number of neurons between 160x180 and 180x200. In addition, their best result (99.81% detection rate and 0.1% false positive rate) was just trained and tested with one DoS attack type, the smurf attack. These different results are summarized in Table 3. Note that our detection rates are lower than the rest of the proposed IDSs. Nevertheless, we provide a more simple architecture with less neuron, which was automatically generated during the training, and where the data hierarchical structure is easier to understand.

4 Conclusions

In this paper, a growing hierarchical self-organizing map (GHSOG) has been proposed. This novel neural network faces the limitations of the SOM related to its static architecture and the lack of representation of hierarchical relations among input data. The GHSOM model also faces these problems of the SOM, but keep maintaining some limitations: the static topology of the maps (2-D grids) and the big amount of neurons created without necessity. The proposed GHSOG has a topology based on graphs instead of 2-D rectangular grids, where the neurons are considered the vertices, and each edge of the graph represents a neighborhood relation between neurons. This graph-based representation provides greater plasticity and a more flexible and adaptive architecture, where the neurons arrangement is not restricted to a fixed topology, achieving a more faithfully data representation.

The performance of the GHSOG has been tested by implementing an Intrusion Detection System (IDS). To train and test the neural network, the KDD Cup 1999 benchmark data set has been used. During the training, two different data subsets has been selected from the 10% training KDD data set, with 100,000 and 169,000 connection records, respectively. The two generated GHSOG architectures were tested with the 10% training KDD data set, which contains 311,029 connection records. After testing, we achieved a 90.15% detection rate and a false positive rate of 2.09% as shown in Table 2. Comparison results are also provided. Although testing results cannot seem as good as in other related works, these just are limited to specific subsets of features and/or attacks, which were established in advanced and the number of neurons are higher than we used. Furthermore, our architecture was generated during the training adapting to input data, mirroring their hierarchical relations and achieving a more faithfully data representation.

Acknowledgements

This work is partially supported by Spanish Ministry of Science and Innovation under contract TIN-07362, project name Self-Organizing Systems for Internet.

References

1. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological cybernetics 43(1), 59–69 (1982)
2. Rauber, A., Merkl, D., Dittenbach, M.: The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. IEEE Transactions on Neural Networks 13(6), 1331–1341 (2002)
3. Phuc, D., Hung, M.: Using som based graph clustering for extracting main ideas from documents. In: IEEE International Conference on Research, Innovation and Vision for the Future, pp. 209–214 (2008)
4. Hussin, M., Farra, M., El-Sonbaty, Y.: Extending the growing hierarchal som for clustering documents in graphs domain. In: IEEE International Joint Conference on Neural Networks, pp. 4028–4035 (2008)
5. Maxion, R., Tan, K.: Anomaly detection in embedded systems. IEEE Transactions on Computers 51(2), 108–120 (2002)
6. Tan, K., Maxion, R.: Determining the operational limits of an anomaly-based intrusion detector. IEEE Journal on Selected Areas in Communications 21(1), 96–110 (2003)
7. Ying, H., Feng, T.J., Cao, J.K., Ding, X.Q., Zhou, Y.H.: Research on some problems in the kohonen som algorithm. In: International Conference on Machine Learning and Cybernetics, vol. 3, pp. 1279–1282 (2002)
8. Sarasamma, S., Zhu, Q., Huff, J.: Hierarchical kohonenen net for anomaly detection in network security. IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 35(2), 302–312 (2005)
9. DeLooze, L., AF DeLooze, L.: Attack characterization and intrusion detection using an ensemble of self-organizing maps. In: 7th Annual IEEE Information Assurance Workshop, pp. 108–115 (2006)
10. Mitrokotsa, A., Douligeris, C.: Detecting denial of service attacks using emergent self-organizing maps. In: 5th IEEE International Symposium on Signal Processing and Information Technology, pp. 375–380 (2005)

Real Adaboost Ensembles with Emphasized Subsampling[*]

Sergio Muñoz-Romero, Jerónimo Arenas-García, and Vanessa Gómez-Verdejo

Department of Signal Theory and Communications,
Universidad Carlos III de Madrid, 28911 Leganés-Madrid, Spain
{smunoz,jarenas,vanessa}@tsc.uc3m.es
http://www.tsc.uc3m.es

Abstract. Multi-Net systems in general, and the Real Adaboost algorithm in particular, offer a very interesting way of designing very powerful classifiers. However, one inconvenient of this schemes is the large computational burden required for their construction. In this paper, we propose a new Boosting scheme which incorporates subsampling mechanisms to speed up the training of base learners and, therefore, the setup of the ensemble network. Furthermore, subsampling the training data provides additional diversity among the constituent learners, according to the some principles exploited by Bagging approaches. Experimental results show that our method is in fact able to improve both Boosting and Bagging schemes in terms of recognition rates, while allowing significant training time reductions.

1 Introduction

Nowadays, Artificial Neural Networks (ANN) have become a very important tool for Machine Learning. However, when a difficult task has to be solved a strong and usually very complex network has to be employed to obtain a good solution. To overcome this limitation, Multi-Net (M-N) systems have emerged under the idea of combining a set of "weak" learners to build a classifier of improved performance.

Although many methods to build of M-N systems can be found in the literature [1,2], two of them have received special attention because of their simplicity and good performance: Bagging [3] and Boosting [4]. Both methods build an NN ensemble by combining a set of learners that present different solutions, i.e., diversity. Bagging schemes enforce this diversity by training each learner with a different subset (generated by a subsampling procedure) of the complete training data, while Boosting methods and, in particular, the Adaboost algorithm [5] sequentially trains the learners, by emphasizing samples according to the error of previous networks. Although both methods follow a similar methodology, the emphasis mechanism employed by Adaboost has made it one of the most powerful M-N systems.

---

[*] This work was supported in part by the MEC Pjt. TEC2008-02473/TEC and the CM Pjt. S-0505/TIC/0223.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 440–447, 2009.
© Springer-Verlag Berlin Heidelberg 2009

In [6] we presented a method to alleviate the computational burden during the operation of the classifier. Here we pay attention to the training of the ensemble, and propose a new method which achieves significant reductions in training time, while preserving the good recognition performance of Adaboost. Our novel method, that we call Subsampled Boosting (SB), follows the principles of both Bagging and Adaboost approaches, in the sense that Real Adaboost (RA) emphasis is used to guide subsampling of the training data.

In the following section, we will briefly review the two reference methods: Bagging and Adaboost. Next, in Section 3, we will describe how the Bagging and Adaboost schemes are combined to derive the Subsampled Boosting method. In Section 4, we will analyze the advantages of our proposal in terms of computational savings and accuracy. To conclude, in Section 5 conclusions and future research lines will be presented.

2 Review of Existing M-N Methods

In this section, we give a brief description of the probably two most popular algorithms for committee machines creation: Bagging and Adaboost. Among the different versions that can be found in the Adaboost literature, here we describe the Real Adaboost approach presented by Schapire and Singer [7].

2.1 Bagging

Bagging methods rely on the linear combination of several base learners, where each of these learners has been trained with a different data subset. To create these subsets, the Bagging method employs bootstrap sampling, i.e., it randomly selects a subset of samples from the original training set.

Let us consider a binary classification problem defined by the following training data set $S = \left\{\left(\mathbf{x}^{(l)}, d^{(l)}\right) \in \chi \times \{-1, 1\}, l = 1, \ldots, L\right\}$, where $\mathbf{x}^{(l)}$ is the l-th training pattern and $d^{(l)}$ its corresponding label. Then, to build a Bagging ensemble with T learners, a different bootstrap training set, χ_t^B, has to be created for each learner. Base learners are then trained to minimize a particular cost function, such as the mean square error

$$C_t = \sum_{l=1}^{L} [d^{(l)} - o_t(\mathbf{x}^{(l)})]^2, \tag{1}$$

where $o_t(\mathbf{x}) : \chi \to [-1, 1]$ is the output function t-th learner.

The final output of the ensemble for sample $\mathbf{x}$ is obtained by averaging the learners outputs, i.e.

$$f_{bag}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} o_t(\mathbf{x}). \tag{2}$$

2.2 Real Adaboost

The RA algorithm implements also a linear combination of weak learners, but now the base networks are trained sequentially forcing at each round that the new learner pays more attention to patterns that have been misclassified by the previous networks.

To be more specific at each round $t = 1, \ldots, T$ a new learner, $o_t(\mathbf{x}) : \chi \rightarrow [-1, 1]$, is trained to minimize a weighted mean square error,

$$C_t = \sum_{l=1}^{L} D_t(\mathbf{x}^{(l)})[d^{(l)} - o_t(\mathbf{x}^{(l)})]^2, \tag{3}$$

where $D_t(\mathbf{x}^{(l)})$ is the emphasis function that indicates the importance that the t-th learner should assign to pattern $\mathbf{x}^{(l)}$. At first, this function is uniformly distributed ($D_t(\mathbf{x}^{(l)}) = 1/L, l = 1, \ldots, L$), and in the following rounds it is updated as

$$D_{t+1}(\mathbf{x}^{(l)}) = \frac{D_t(\mathbf{x}^{(l)}) \exp\left(-\alpha_t o_t(\mathbf{x}^{(l)}) d^{(l)}\right)}{Z_t} \tag{4}$$

where α_t is the weight that the ensemble assigns to the t-th learner, and Z_t is a normalization factor that guarantees $\sum_{l=1}^{L} D_t(\mathbf{x}^{(l)}) = 1$. The output weight associated to each learner is calculated at each round as

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 + r_t}{1 - r_t}\right) \tag{5}$$

where $r_t = \sum_{l=1}^{L} D_t(\mathbf{x}^{(l)}) o_t(\mathbf{x}^{(l)}) d^{(l)}$.

Once all learners have been trained and the output weights have been calculated, the final ensemble output is obtained according to

$$f_{RA}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t o_t(\mathbf{x}) \tag{6}$$

3 Subsampled Boosting

Although RA generally performs better than Bagging approaches, it also incurs in a significantly larger computational burden since all training patterns are used to train every base classifier. In this section we propose a modified version of RA that by using bootstrap strategies similar to those of Bagging, can significantly accelerate the training phase, while preserving RA's excellent recognition capabilities. The new method, to which we will refer as Subsampled Boosting (SB), can therefore induce diversity by simultaneously using the emphasis and subsampling mechanisms of RA and Bagging, respectively.

Just like the RA algorithm, SB works sequentially, but now each base learner is trained using an emphasized subset of the training data, i.e.,

$$C_t = \sum_{l \in S_t^B} D_t(\mathbf{x}^{(l)})[d^{(l)} - o(\mathbf{x}^{(l)})]^2 \tag{7}$$

where S_t^B is the subset of patterns used by the t-th learner. By reducing the number of training samples at each round, SB gets a clear computational advantage during the learners training.

To generate the data subsets S_t^B our method uses a modified version of bootstrap sampling. Since the emphasis function indicates on which patterns we have to focus, this information is also used to generate the different data subsets $\left\{S_t^B\right\}_{t=1}^T$, adjusting the probability of selecting each sample according to the current value of the emphasis function. This procedure seems similar to the "roulette wheels" used in the individuals selection from genetic methods [8].

The update of the emphasis function and computation of the learners output weights is carried out according to (4) and (5), using all the available training patterns. In other words, the subsampling mechanism only reduces the number of samples which are used to train individual learners, but after each learner has been trained, all patterns are used to compute the corresponding output weight and to readjust the emphasis.

Finally, we would like to remark that both the emphasis and subsampling mechanisms are used by SB as two complementary ways of creating diversity among the ensemble networks. As we will show in the following section, in some cases, this provides significant accuracy improvements with respect to RA and Bagging approaches.

4 Experiments

To test the efficiency of the SB classifier we are going compare its performance with both Bagging and RA schemes. Since SB and Bagging employ a subsampling procedure, we will test their performances using three different subsampling rates: 80%, 60%, and 40%.

These experiments will be carried out over six binary problems: the synthetic dataset *Ripley* (*Ri*) [9] and five real problems taken from [10]: *Abalone* (*Ab*), *Hand* (*Ha*), *Image* (*Im*), *Phoneme* (*Ph*), and *Spam* (*Sp*). Table 1 shows the main features of these problems: number of dimensions (dim) and number of training and test samples (#Train and #Test) in each class ($\mathrm{n}_1/\mathrm{n}_{-1}$). When the problem had no predefined train/test partitions, ten random partitions have been created with 60%/40% for training and test, averaging the results over all partitions.

The three ensembles under study, Bagging, RA, and SB, employ Multi Layer Perceptrons (MLPs) with one hidden layer as base learners. To train each MLP, a back-propagation algorithm is used, adjusting the learning step to $\mu = 0.01$, both for the hidden and output layers.

To achieve an optimal ensemble behavior, an appropriate number of hidden neurons (M) has to be fixed. Despite each ensemble can need a different M value, to carry out a fair comparison in computational terms, we have preferred to fix the same M for all algorithms (Bagging, RA, and SB) and subsampling rates (80%, 60% or 40%). Then, this parameter has been selected by, first applying 5 fold cross-validation over a single MLP trained with different number of data

Table 1. Summary of the six benchmark problems

	M	dim	# Train (n_1/n_{-1})	# Test (n_1/n_{-1})
Abalone (*Ab*)	6	8	2507 (1238/1269)	1670 (843/827)
Hand (*Ha*)	82	62	3823 (1923/1900)	1797 (906/891)
Image (*Im*)	54	18	1848 (821/1027)	462 (169/293)
Phoneme (*Ph*)	68	5	3243 (952/2291)	2161 (634/1527)
Ripley (*Ri*)	49	2	250 (100/150)	1000 (500/500)
Spam (*Sp*)	24	57	2761 (1673/1088)	1840 (1115/725)

Table 2. Averaged classification error (CE) and number of base learners (T_{av}) achieved by Bagging (Bg), RealAdaboost (RA) and Subsampled Boosting (SB). Bagging and Subsampled Boosting results are given for subsampling rates of 40%, 60% and 80%

	Bg_{40}	Bg_{60}	Bg_{80}	RA		SB_{40}		SB_{60}		SB_{80}	
	CE(%)	CE(%)	CE(%)	CE(%)	T_{av}	CE(%)	T_{av}	CE(%)	T_{av}	CE(%)	T_{av}
Ab	19,48 ±0,18	19,30 ±0,19	19,23 ±0,19	19,30 ±0,20	31,96 ±2,48	19,38 ±0,41	26,44 ±4,86	19,23 ±0,33	27,62 ±4,26	**19,19** ±0,37	27,70 ±4,46
Ha	3,89 ±0,25	3,19 ±0,18	2,84 ±0,19	3,48 ±0,37	29,64 ±8,38	2,17 ±0,15	50,00 ±0,00	2,18 ±0,14	50,00 ±0,00	**2,15** ±0,13	50,00 ±0,00
Im	3,68 ±0,32	2,71 ±0,23	2,47 ±0,20	2,41 ±0,38	26,28 ±5,51	2,27 ±0,35	33,52 ±4,26	**2,16** ±0,25	26,80 ±3,84	2,29 ±0,32	25,92 ±3,68
Ph	14,42 ±0,49	13,92 ±0,56	**13,76** ±0,55	13,89 ±0,56	27,68 ±2,37	14,23 ±0,56	22,82 ±3,88	14,10 ±0,55	24,76 ±3,90	14,07 ±0,63	23,92 ±3,45
Ri	10,97 ±0,36	11,04 ±0,22	10,85 ±0,17	9,85 ±0,13	31,02 ±5,05	10,88 ±0,48	16,60 ±4,10	10,17 ±0,49	20,14 ±3,84	**9,77** ±0,36	21,66 ±4,03
Sp	6,16 ±0,50	6,09 ±0,52	5,96 ±0,54	5,84 ±0,59	32,52 ±6,76	**5,63** ±0,27	25,75 ±2,10	5,80 ±0,24	22,12 ±1,89	5,88 ±0,25	20,50 ±1,74

percentages (100%, 80%, 60%, and 40%); after this, the lowest M among the four candidates is selected as the number of hidden neurons (see Table 1).

A last parameter to be adjusted is the number of learners for each ensemble (T). To select it correctly, in the SB and RA algorithms we have followed the criterion given in [11], where the ensemble growth is stopped when the ratio between the average α_t in the last 10 rounds and the sum of all α_t is lower than a stop value (C_{stop}), i.e.,

$$\frac{\sum_{t'=T-9}^{T} \alpha_{t'}}{10 \sum_{t'=1}^{T} \alpha'_t} < C_{\text{stop}} \tag{8}$$

where C_{stop} has been experimentally fixed to 0.01. In problem *Ha*, a very large number of learners is needed to get SB convergence, so we have directly fixed T to 50. The number of learners for Bagging schemes has to be selected a priori. So, for this architecture we will use $T = 30$, which we have checked is enough to get the most of Bagging networks. Furthermore this value is usually similar to the number of weak learners used by RA and SB.

Table 3. Time elapsed in seconds when ensembles with the average number of learners are evaluated. Computational savings with regard to RA are given inside parentheses.

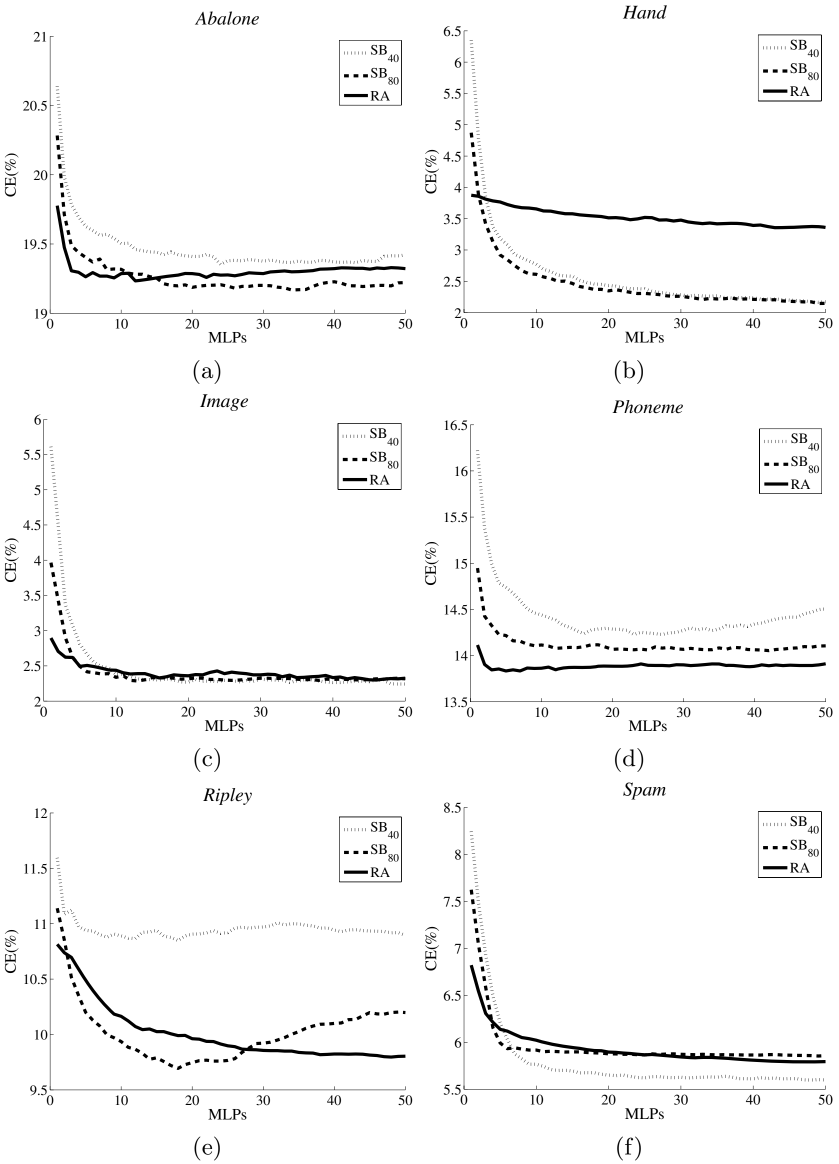
	Bg$_{40}$	Bg$_{60}$	Bg$_{80}$	RA	SB$_{40}$	SB$_{60}$	SB$_{80}$
Ab	140, 81 (−60, 88%)	208, 61 (−42, 04%)	278, 22 (−22, 70%)	359, 93 (0, 00%)	120, 73 (−66, 46%)	195, 40 (−45, 71%)	261, 92 (−27, 23%)
Ha	1030, 77 (−59, 07%)	1537, 03 (−38, 97%)	2036, 52 (−19, 13%)	2518, 35 (0, 00%)	1713, 11 (−31, 98%)	2570, 08 (2, 05%)	3468, 26 (37, 72%)
Im	209, 46 (−51, 65%)	306, 97 (−29, 15%)	406, 74 (−6, 12%)	433, 24 (0, 00%)	229, 32 (−47, 07%)	268, 44 (−38, 04%)	344, 30 (−20, 53%)
Ph	3205, 20 (−51, 84%)	4895, 20 (−26, 44%)	6499, 61 (−2, 34%)	6655, 04 (0, 00%)	2277, 12 (−65, 78%)	4025, 41 (−39, 51%)	5123, 88 (−23, 01%)
Ri	21, 76 (−58, 75%)	32, 63 (−38, 12%)	43, 44 (−17, 63%)	52, 74 (0, 00%)	12, 42 (−76, 45%)	20, 39 (−61, 34%)	31, 59 (−40, 09%)
Sp	2894, 31 (−59, 14%)	4401, 53 (−37, 86%)	5261, 94 (−25, 71%)	7082, 86 (0, 00%)	2468, 74 (−65, 14%)	3237, 60 (−54, 29%)	3523, 30 (−50, 26%)

Table 2 summarizes the classification errors (CE) and the average number of learners (T_{av}) achieved by each method in the different problems. These results have been averaged over 50 independent runs, then, their mean values and their standard deviations are shown. The best result for each problem is indicated in boldface. Besides, the computational time to train the different ensembles and the savings provided by our proposal are shown in Table 3.

When comparing SB and Bagging schemes, it is clear that the former shows better performance in terms of CE in most cases, independently of the subsampling rate applied to the training data. For some problems, such as *Ha*, *Im* and *Sp*, this performance can be quite significant. The only problem where Bagging outperforms SB is *Ph*, but in this case the difference is rather small. It is also important to remark that our method usually requires fewer learners than Bagging to achieve these results, thus leading to a faster training of the ensemble, as well as to faster classification of test data.

With respect to RA, using training subsets of 40% of the training data does not result in serious performance degradation, and allows a very important reduction of training times. In fact, we can observe a worse performance only for the *Ri* and *Ph* dataset, while SB excels again in the *Ha* dataset. When increasing the percentage of data used to train the learners (see results under columns SB$_{60}$ and SB$_{80}$ in Table 2 and Table 3), the recognition capabilities of our method improve slightly, but the computational burden is also increased (time savings between 20% and 60% are achieved). In any case, sampling rates of 60% or 80% can be considered as conservative selections to minimize possible performance degradations with respect to standard RA.

To complete our discussion, Fig. 1 shows Real Adaboost and Subsampled Boosting performance as a function of the number of base learners. As we could expect, SB performs poorly during the first iterations, given that the subsampling process negatively affects the performance of the first MLPs incorporated



(a) (b)

(c) (d)

(e) (f)

Fig. 1. Classification error (CE) evolution according to the number of learners for all algorithms for the problems *Abalone* (a), *Hand* (b), *Image* (c), *Phoneme* (d), *Ripley* (e) and *Spam* (f)

to the ensemble. However, after only a few iterations ($T > 10$), SB becomes as competitive as standard RA and, in many cases better than it.

5 Conclusions

In this paper we have proposed an efficient method to alleviate the computational effort demanded by Adaboost algorithm during its training phase. This method, that we call, Subsampled Boosting, incorporates the subsampling procedure of Bagging methods in the Adaboost algorithm. In this way, the learners training time is significantly reduced. Furthermore, we think combining two different procedures to induce diversity among base learners can allow our proposal to take the most of Bagging and RA methods and to achieve reductions of the classification errors.

Experimental results not only corroborate the expected training time savings, but also show accuracy improvement with regard to Adaboost and Bagging schemes. An extension of this proposal to achieve computational savings not only during the training time but also during the operational phase is a promising research line which we are currently working on.

References

1. Kuncheva, L.I.: Combining Pattern Classifiers. Methods and Algorithms. Wiley, New Jersey (2004)
2. Sharkey, A.J.C. (ed.): Combining Artificial Neural Nets. Ensemble and Modular Multi-Net Systems. Perspectives in Neural Computing. Springer, Heidelberg (1999)
3. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
4. Schapire, R.E.: The strength of weak learnability. In: 30th Annual Symposium on Foundations of Computer, pp. 28–33 (1989)
5. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to Boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
6. Arenas-García, J., Gómez-Verdejo, V., Muñoz-Romero, S., Ortega-Moral, M., Figueiras-Vidal, A.R.: Fast Classification with Neural Networks via Confidence Rating. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 622–629. Springer, Heidelberg (2005)
7. Schapire, R.E., Singer, Y.: Improved Boosting algorithms using confidence-rated predictions. Machine Learning 37(3), 297–336 (1999)
8. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
9. Ripley, B.D.: Neural networks and related methods for classification (with discussion). Journal of the Royal Statistical Society 56(3), 409–456 (1994)
10. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
11. Gómez-Verdejo, V., Arenas-García, J., Figueiras-Vidal, A.R.: A dynamically adjusted mixed emphasis method for building Boosting ensembles. IEEE Transactions on Neural Networks 19(1), 3–17 (2008)

Using the Negentropy Increment to Determine the Number of Clusters

Luis F. Lago-Fernández[1] and Fernando Corbacho[2]

[1] Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid, Spain
[2] Cognodata Consulting, Calle Caracas 23, 28010 Madrid, Spain

Abstract. We introduce a new validity index for crisp clustering that is based on the average normality of the clusters. A normal cluster is optimal in the sense of maximum uncertainty, or minimum structure, and so performing further partitions on it will not reveal additional substructures. To characterize the normality of a cluster we use the negentropy, a standard measure of distance to normality which evaluates the difference between the cluster's entropy and the entropy of a normal distribution with the same covariance matrix. Although the definition of the negentropy involves the differential entropy, we show that it is possible to avoid its explicit computation by considering only negentropy increments with respect to the initial data distribution. The resulting *negentropy increment* validity index only requires the computation of determinants of covariance matrices. We have applied the index to randomly generated problems, and show that it provides better results than other indices for the assessment of the number of clusters.

Keywords: Crisp clustering, Cluster validation, Negentropy.

1 Introduction

The goal of cluster analysis [1] is the automatic partition of a data set into a finite number of natural structures, or clusters, so that the elements inside a cluster are similar while those belonging to different clusters are not. Clustering algorithms are usually divided into crisp and fuzzy. In crisp clustering, each data point is uniquely assigned to a single cluster. On the contrary, fuzzy clustering allows each point to belong to any of the clusters with a certain degree of membership. A common problem to both approaches is the lack of a general framework to measure the validity of the outcomes of a particular clustering method, and in particular to assess the correct number of clusters. This is the subject of cluster validation [2], whose objective is to provide a quality measure, or validity index, that allows to evaluate the results obtained by a clustering algorithm.

In this paper we propose a new validity index for crisp clustering that is based on the normality of the clusters. The normal or Gaussian distribution maximizes the entropy for a fixed covariance matrix [3]. This means that a normally distributed cluster has the lowest possible structure, and so it can

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 448–455, 2009.
© Springer-Verlag Berlin Heidelberg 2009

not be further partitioned into additional substructures. This principle suggests to select the partition for which the clusters are on average more Gaussian. The normality of a probability distribution can be measured by means of the negentropy, which is defined by subtracting the entropy of the distribution from the entropy of a Gaussian with the same covariance matrix. The negentropy is a frequently used measure of distance to normality [4], and has been widely used to measure normality in the context of projection pursuit and independent component analysis [5].

The lower the negentropy of the cluster, the more Gaussian it is. So, according to the previous principle, cluster validation should be accomplished by selecting the partition whose clusters have on average lower negentropy. The most involved technical issue regarding the use of the negentropy to measure the normality of a partition is related to the computation of the differential entropy of the clusters. However, we show that, by subtracting the negentropy of the initial data distribution (no partition), we obtain a validity index that: (i) measures the increment in normality that is gained with the partition; and (ii) avoids the explicit computation of any differential entropies. We call this index the *negentropy increment* associated to the partition.

To test the negentropy increment as a cluster validity index, we use it as the fitness function of a genetic algorithm that searches the partitions space. We have applied this method to a set of randomly generated problems that consist of Gaussian mixtures in 2D. We show that the proposed negentropy based validity index is able to assess the correct number of clusters with more reliability than other indices proposed in the literature for the validation of crisp clustering, such as the Davies-Bouldin [6], the Dunn [7,8], or the PBM [9] indices.

2 The Negentropy Increment of a Partition

Given a random variable $\mathbf{x}$ with probability density function $f(\mathbf{x})$, the negentropy of $\mathbf{x}$ is defined as:

$$J(\mathbf{x}) = \hat{H}(\mathbf{x}) - H(\mathbf{x}) \tag{1}$$

where $H(\mathbf{x})$ is the differential entropy of $\mathbf{x}$ and $\hat{H}(\mathbf{x})$ is the differential entropy of a normal distribution with the same covariance matrix. Due to the maximum entropy property of the normal distribution, the negentropy is always equal to or greater than 0, with equality holding if and only if the distribution is Gaussian.

Given a crisp partition of the data space into a set of k non overlapping regions $\{\Omega_1, \Omega_2, ..., \Omega_k\}$, we compute its average negentropy, which we denote $\bar{J}(\mathbf{x})$, as:

$$\bar{J}(\mathbf{x}) = \sum_{i=1}^{k} p_i J_i(\mathbf{x}) \tag{2}$$

where p_i is the a-priori probability of $\mathbf{x}$ falling into the region Ω_i, and $J_i(\mathbf{x})$ is the negentropy of $\mathbf{x}$ in the region Ω_i. The lower $\bar{J}(\mathbf{x})$, the better (more Gaussian on average) the partition is. Note that we can add any constant to $\bar{J}(\mathbf{x})$ without affecting this result so, instead of Eq. 2, we will consider the index:

$$\varDelta J = \bar{J}(\mathbf{x}) - J_0(\mathbf{x}) \tag{3}$$

where $J_0(\mathbf{x})$ is the negentropy of $\mathbf{x}$ when no partition is performed, which is a constant for each problem. We call this index the *negentropy increment* after the partition. It measures the change in negentropy when we make a partition on the data set. If $\varDelta J < 0$ then we are gaining normality (losing negentropy) with the partition, while if $\varDelta J > 0$ we are losing normality (gaining negentropy). Given two different partitions of a data set, we will select the one for which $\varDelta J$ is lower.

The main advantage of computing the difference of negentropies in Eq. 3 is that, after some manipulations, the differential entropies cancel out and the negentropy increment can be written as:

$$\varDelta J = \frac{1}{2}\sum_{i=1}^{k} p_i \log |\varSigma_i| - \frac{1}{2}\log |\varSigma_0| - \sum_{i=1}^{k} p_i \log p_i \tag{4}$$

where $\varSigma_i$ is the covariance matrix of $\mathbf{x}$ in the region $\varOmega_i$ and $\varSigma_0$ is the covariance matrix of the complete data set $\mathbf{x}$. Note that to evaluate this final expression we only need to compute the determinants of the covariance matrices for each region. This index can be applied as a general tool to validate the outcome of any crisp clustering algorithm, and also to compare solutions provided by different algorithms for a single problem.

3 Evaluation of the New Validity Index

To evaluate the negentropy increment as a validity index for crisp clustering, we test its ability to assess the correct number of clusters in a set of 500 randomly generated problems in 2 dimensions. For every problem, the number of clusters, n, is between 1 and 5, each one consisting of 200 points randomly extracted from a normal distribution whose parameters are also randomly selected. The set of points (x, y) belonging to a particular cluster is generated as follows. First, two random numbers, μ_x and μ_y, are extracted from a uniform distribution in the interval $(0, 10)$, and other two, σ_x and σ_y, are extracted from a uniform distribution in the interval $(0, 1)$. Then x is extracted from the normal distribution $N(\mu_x, \sigma_x)$ and y is extracted from $N(\mu_y, \sigma_y)$. Finally a rotation by an angle θ, with center at (μ_x, μ_y), is applied to the points (x, y). The angle θ is randomly selected from a uniform distribution in $(0, 2\pi)$. There are 100 data sets for each value of n. Fig. 1 shows some of the data sets for $n = 3$, $n = 4$ and $n = 5$. Note that, as the number of clusters increases, the overlap among them is higher in general.

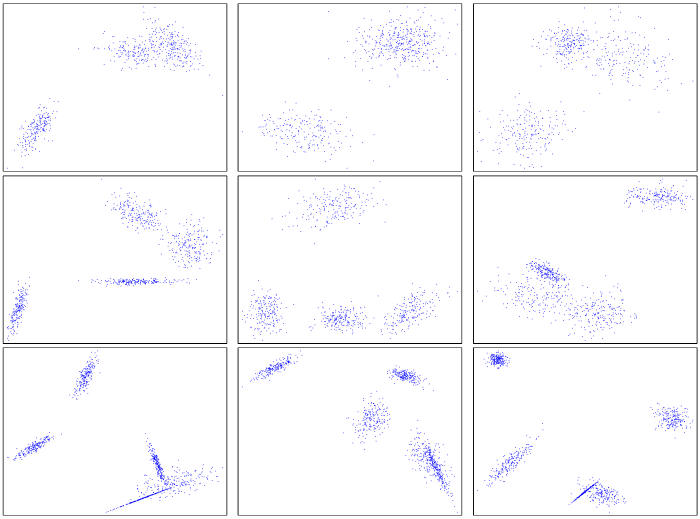


Fig. 1. Some examples of the data sets used to test the cluster validity measure. The top row shows some sets with tree clusters ($n = 3$). The middle row shows sets consisting of four clusters ($n = 4$). The bottom row shows data sets with five clusters ($n = 5$).

For a given problem, a genetic algorithm is used to search for the partition that minimizes the negentropy increment. The kind of partitions searched by the algorithm consist of convex non overlapping regions delimited by linear separators. Given the number k of partition regions, the partition is defined as a 2-dimensional Voronoi diagram around the k centers $\{\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_k}\}$. The region Ω_i consists of all the points that are closer to $\mathbf{p_i}$ than to any other center. In the genetic algorithm implementation we codify each region center as a set of 10×2 bits (10 bits per coordinate), and so the full partition can be coded as a binary string of length $10 \times 2 \times k$. We use the PGAPack genetic algorithm library [10] with the default mutation and crossover operators. In all the trials performed the population size is set to 500 individuals, each one representing a different partition. After random initialization, the algorithm is run for 250 iterations, and the best (lowest ΔJ) partition at the end is used as the solution for a particular run. We consider k ranging between 1 and 9, and we make 20 different runs for each k. The final number of clusters is selected as the minimum k for which ΔJ lies within a 95% of the absolute minimum found throughout the 9×20 trials which are run for the problem.

In order to compare the ΔJ validity index with other approaches to the validation of crisp clustering, we have performed the same kind of experiments using the Davies-Bouldin (DB) index [6], the Dunn index [7,8] and the PBM index [9]. The DB and the Dunn indices were found among the best in a study that compared 23 validity indices on 12 data sets that consisted of bivariate Gaussian mixtures [11]. On the other hand, the PBM index was shown to outperform the other two in [9].

4 Results

The results are evaluated by comparing the actual number of clusters in the problem with the number of regions in the best partition provided by the algorithm for a given validity index. In Fig. 2 we show the selected partitions for all the data sets shown in Fig. 1, using the ΔJ index. Note that the algorithm performs quite well even in cases where two clusters are partially overlapping, predicting the correct number of clusters in 7 out of the 9 sets.

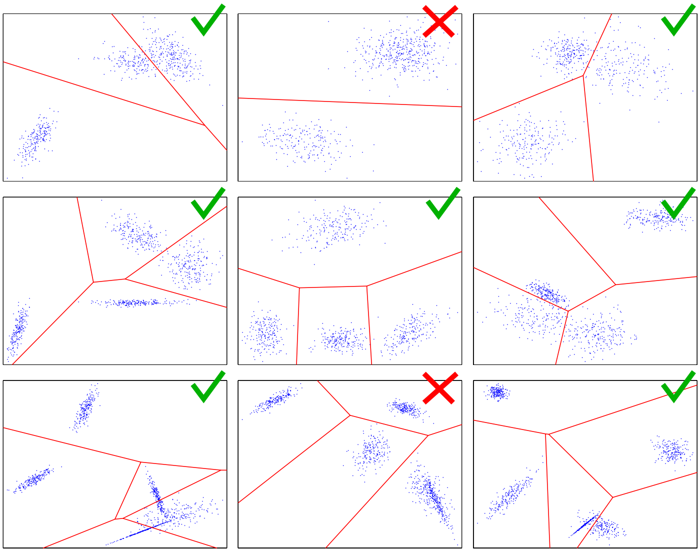


Fig. 2. Selected partitions, according to ΔJ, for all the data sets shown in Fig. 1. Partitions marked with a tick are correct. Those marked with a cross are wrong. The partitions identify the correct clusters in 7 of the 9 sets.

In Fig. 3 we show the average number of regions, k_{opt}, versus the actual number of clusters, n, obtained with each of the 4 validity indices considered. Each point in the figure is an average over the 100 problems with the same number of clusters. Note that:

1. The ΔJ index is the only one which provides the correct partition for data sets with one single cluster.
2. For data sets with two clusters all the methods, except the PBM, produce a satisfactory solution.
3. For data sets with 3, 4 and 5 clusters all the methods tend to underestimate the number of clusters, presumably due to the higher overlap. The PBM and the ΔJ indices behave similarly and beat the others for these problems.
4. The ΔJ index is the only one that shows a good performance for the full range of n.

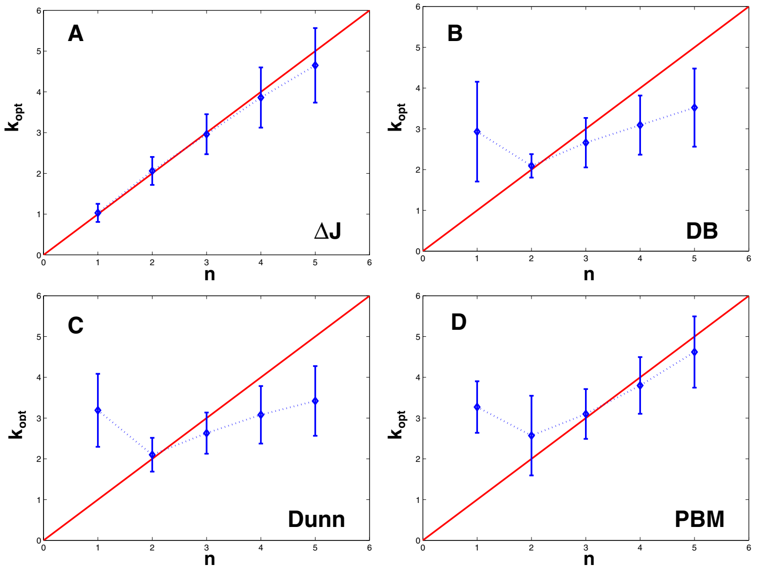


Fig. 3. Average number of regions versus number of clusters in the data set for all the four validity indices considered. A. ΔJ index. B. DB index. C. Dunn index. D. PBM index.

To quantify these observations we have computed, for each index and each number of clusters n, the percentage of data sets for which the algorithm predicts the correct number of clusters ($k_{opt} = n$). The results are summarized in table 1. The ΔJ index provides the best results for all the cases except $n = 2$, for which the Dunn index obtains the highest score. Note, however, that in this case the ΔJ and the DB indices also reach very high values. The high accuracy obtained by these three indices for $n = 2$ is also observed in Fig. 3, and is presumably due

Table 1. Percentage of data sets for which the number of clusters is correctly predicted by the algorithm, using the 4 different validity indices. Each value is calculated using 100 data sets. Values in bold are the best results for a given n.

n	ΔJ	DB	Dunn	PBM
1	**98%**	-	-	-
2	91%	91%	**93%**	66%
3	**87%**	60%	61%	74%
4	**66%**	31%	29%	57%
5	**47%**	14%	12%	45%

to the low overlap. Also note that the ΔJ index finds the correct solution for the 98% of the problems with one single cluster, while the other indices always produce partitions into two or more regions.

5 Discussion

In this paper we have introduced a new crisp cluster validity index that is based on the average normality of the clusters. The normality of a cluster is measured by means of its negentropy. To avoid the explicit calculation of differential entropies in the computation of the index, we subtract the negentropy of the original data distribution. For a crisp partition, the final form of the validity index only requires the computation of the determinants of the covariance matrices of the data for each partition region. Evaluation of the index using a randomly generated set of problems shows that it is able to assess the correct number of clusters with more reliability than other frequently used crisp cluster validity measures, such as the Dunn, the Davies-Bouldin and the PBM indices.

The idea of using normality as a measure of a cluster's quality underlies the Gaussian Mixture Model, where many studies address the problem of cluster validation, in particular the assessment of the number of clusters [12,13,14,15,16,17]. In general these approaches use validity indices that combine the log-likelihood with some measure that penalizes the model complexity. Unfortunately, due to their probabilistic nature, these validity indices can not be easily applied to crisp partitions. The use of the negentropy for cluster validation has been explored in some recent works. For example, Geva et al. [18] used a cluster validity index based on the negentropy calculated along single dimensions; and Ciaramella et al. [19] calculate the negentropy along Fisher's projection to determine whether two clusters must be merged. Both of these works limit the estimation of the negentropy to one single dimension. Instead, the proposed approach provides a full characterization of the clusters' normality through the negentropy increment.

Although here we have restricted our analysis to crisp clustering, it is possible in principle to extend the results to fuzzy clustering. In this case, a new term measuring the uncertainty in the cluster given the data appears in the expression of ΔJ. Work in progress deals with the extension of the negentropy increment validity index in this direction, as well as with the evaluation of the index with standard benchmark problems in arbitrary dimensions.

Acknowledgments. This work has been partially supported with funds from MEC BFU2006-07902/BFI, MEC PHB2007-0013 and CAM S-SEM-0255-2006.

References

1. Everitt, B., Landau, S., Leese, M.: Cluster Analysis. Hodder Arnold, London (2001)
2. Gordon, A.D.: Cluster Validation. In: Hayashi, C., Ohsumi, N., Yajima, K., Tanaka, Y., Bock, H.H., Baba, Y. (eds.) Data Science, Classification and Related Methods, pp. 22–39. Springer, New York (1998)

3. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley, New York (1991)
4. Comon, P.: Independent Component Analysis, a New Concept? Signal Processing 36(3), 287–314 (1994)
5. Hyvärinen, A.: New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit. Technical Report A47, Dept. of Computer Science and Engineering and Laboratory of Computer and Information Science, Helsinki Univ. of Technology (1997)
6. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Trans. Pattern Analysis and Machine Intelligence 1(4), 224–227 (1979)
7. Bezdek, J.C., Pal, R.N.: Some New Indexes of Cluster Validity. IEEE Trans. Systems, Man and Cybernetics B 28(3), 301–315 (1998)
8. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-Separated Clusters. J. Cybernetics 3(3), 32–57 (1973)
9. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.: Validity Index for Crisp and Fuzzy Clusters. Pattern Recognition 37(3), 487–501 (2004)
10. Levine, D.: PGAPack Parallel Genetic Algorithm Library, `http://www-fp.mcs.anl.gov/CCST/research/reports_pre1998/comp_bio/stalk/pgapack.html`
11. Bezdek, J.C., Li, W.Q., Attikiouzel, Y., Windham, M.: A Geometric Approach to Cluster Validity for Normal Mixtures. Soft Computing 1, 166–179 (1997)
12. Biernacki, C., Celeux, G., Govaert, G.: An Improvement of the NEC Criterion for Assessing the Number of Clusters in a Mixture Model. Pattern Recognition Letters 20(3), 267–272 (1999)
13. Bozdogan, H.: Choosing the Number of Component Clusters in the Mixture-Model Using a New Information Complexity Criterion of the Inverse-Fisher Information Matrix. In: Opitz, O., Lausen, B., Klar, R. (eds.) Data Analysis and Knowledge Organization, pp. 40–54. Springer, Heidelberg (1993)
14. Figueiredo, M.A.T., Jain, A.K.: Unsupervised Learning of Finite Mixture Models. IEEE Trans. Pattern Analysis and Machine Intelligence 24(3), 381–396 (2002)
15. Rasmussen, C.: The Infinite Gaussian Mixture Model. In: Solla, S., Leen, T., Müller, K.-R. (eds.) Advances in Neural Information Processing Systems, vol. 12, pp. 554–560. MIT Press, Cambridge (2000)
16. Neal, R.M.: Markov Chain Sampling Methods for Dirichlet Process Mixture Models. J. Computational and Graphical Statistics 9(2), 249–265 (2000)
17. Richardson, S., Green, P.: On Bayesian Analysis of Mixtures with Unknown Number of Components. J. Royal Statistical Soc. B 59, 731–792 (1997)
18. Geva, A.B., Steinberg, Y., Bruckmair, S., Nahum, G.: A Comparison of Cluster Validity Criteria for a Mixture of Normal Distributed Data. Pattern Recognition Letters 21(6-7), 511–529 (2000)
19. Ciaramella, A., Longo, G., Staiano, A., Tagliaferri, R.: NEC: A Hierarchical Agglomerative Clustering Based on Fisher and Negentropy Information. In: Apolloni, B., Marinaro, M., Nicosia, G., Tagliaferri, R. (eds.) WIRN 2005 and NAIS 2005. LNCS, vol. 3931, pp. 49–56. Springer, Heidelberg (2006)

A Wrapper Method for Feature Selection in Multiple Classes Datasets[⋆]

Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, and Rosa M. Calvo-Estévez

University of A Coruña, Department of Computer Science, 15071 A Coruña, Spain
nsanchez@udc.es, ciamparo@udc.es, infmts00@ucv.udc.es

Abstract. Feature selection algorithms should remove irrelevant and redundant features while maintaining or even improving performance, and thus contributing to enhance generalization in learning models. Feature selection methods can be mainly grouped into filters and wrappers. Most of the models built can deal more or less adequately with binary problems, but often under perform on multi-class tasks. In this article, a new wrapper method, called IAFN-FS (Incremental ANOVA and Functional Networks-Feature Selection) is described in its version for dealing with multiclass problems. In order to carry out the multiclass approach, two different alternatives were tried: (a) treating directly the multiclass problem, (b) dividing the original multiclass problem in several binary problems. In order to evaluate the performance of both approaches, a comparative study using several benchmark datasets, our two methods and other wrappers based in classical algorithms, such as C4.5 and Naive-Bayes, was carried out.

1 Introduction

Nowadays, many computer applications (data mining, pattern recognition and machine learning methods, for example), dealing with very large data sets are common. Many real data are affected by the existence of irrelevant or redundant attributes, that can influence learning algorithms in performance, learning time, number of samples necessary and cost. For these reasons, feature selection methods are attracting attention, because they center in obtaining a subset of features that adequately describe the problem at hand without degrading (and most of the times, improving) performance.

There are two main models dealing with feature selection: filter methods and wrapper methods [1]. Wrapper models optimize a predictor as part of the selection process, while filter models rely on the general characteristics of the training data to select the best features with independence of any predictor. Although filters are computationally less expensive, wrapper models give usually better results. Wrapper methods use a learning algorithm and a search technique to score subset of features according to their predictive power to determine the best values (regarding certain criteria) of some parameters.

[⋆] This work has been funded in part by Project TIN2006-02402 of the Ministerio de Educación y Ciencia, Spain (partially supported by the European Union ERDF).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 456–463, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Although feature selection can deal with classification and regression problems, the majority of applications work with classification problems [1]. On the other hand, there has been increasing progress in the development of binary classifiers, while it is more difficult to find methods that can deal directly with multiclass problems, as very few research has been done in this aspect [2,3]. In these studies, the multiclass approaches are shown to suffer from the so-called accumulative effect, which becomes more visible with the growing number of classes and results in removing relevant and unredundant features. The main difficulties to be taken into account in multiclass algorithms are the following:

- the data set presents one or several classes that contain a considerable higher number of samples than the data of the other classes. This is the problem that appears in several well-known data sets, such as KDDCup 99 or Zoo for example, and in this case most algorithms tend to classify the new samples in the majority classes.
- It is complicated to determine which features are appropriate for each class, because feature selection results in a set of attributes that could represent only the majority classes [4].

In this paper a new wrapper method for multiclass feature selection is presented. The method is a extension of a previous binary wrapper, called AFN-FS [5], that is based on functional decomposition and sensibility analysis. Two different approaches were tried: the first one treats the multiclass problem directly, while the second divides the original multiclass problem in several binary problems. In order to evaluate the performance of both approaches, several benchmark datasets were processed using our two methods and other wrappers based in classical algorithms, such as C4.5 and Naive-Bayes.

2 The Wrapper Method

The AFN-FS is a wrapper method that employs the AFN as induction algorithm. The AFN (ANOVA and Functional Networks) is based on a combination of the ANOVA (ANalysis Of VAriance) decomposition and functional networks [6] in order to estimate a function f in terms of n variables, $f(x_1, x_2, \ldots, x_n)$, by approximating its functional components. According to Sobol[7], any square integrable function, $f(x_1, x_2, \ldots, x_n)$, can always be written as the sum of the 2^n orthogonal summands:

$$f(x_1, \ldots, x_n) = f_0 + \sum_{\nu=1}^{2^n-1} f_\nu(\mathbf{x}_\nu), \tag{1}$$

where ν represents each possible subset formed with the n input variables and f_0 is a constant that corresponds to the function with no arguments.

The AFN method approximates each functional component $f_\nu(\mathbf{x}_\nu)$ in (1) as:

$$f_\nu(\mathbf{x}_\nu) = \sum_{j=1}^{k_\nu} c_{\nu j} p_{\nu j}(\mathbf{x}_\nu), \tag{2}$$

where $c_{\nu j}$ are parameters to be estimated and p_ν is a set of orthonormalized basis functions. There exists several alternatives to choose those functions [8].In our case, we have used Legendre polynomials, form tensor products with them and select a subset of them.

The $c_{\nu j}$ parameters are learnt by solving an optimization problem:

$$Minimize \quad J = \sum_{s=1}^{m} \epsilon_s^2 = \sum_{s=1}^{m} (y_s - \hat{y}_s)^2, \tag{3}$$

being m the number of available samples, y_s the desired output for the sample s and $\hat{y}_s$ the estimated output obtained by:

$$\hat{y}_s = \hat{f}(x_{s1}, \ldots, x_{sn}) = f_0 + \sum_{\nu=1}^{2^n-1} \sum_{j=1}^{k_\nu} c_{\nu j} p_{\nu j}(\mathbf{x}_{s\nu}). \tag{4}$$

Once the $c_{\nu j}$ parameters are learnt, two sets of indices are directly derived, GSI (Global Sensitivity Index) and TSI (Total Sensitivity Index). GSI reflects the relevance, in terms of variance, of each functional component in (1) and TSI reflects the relevance of each feature. The expressions for both indices are the following:

$$GSI_\nu = \sum_{j=1}^{k_\nu} c_{\nu j}^2 \quad \nu = 1, 2, \ldots, 2^n - 1; \quad TSI_i = \sum_{\nu=1}^{2^n-1} GSI_\nu \quad x_i \in \nu; i = 1, \ldots, n.$$

The use of those indices allows us to determine which features are relevant, individually or in combination with others, in the problem at hand. Then a wrapper method, called AFN-FS, was developed based on this information provided by the AFN induction algorithm. The characteristics of this wrapper method are the following:

- **Initial state.** The complete set of features.
- **State Space.** As well as in other wrappers, the state space is formed by all the possible feature subsets, derived from the initial feature set.
- **Stop criteria.** The algorithm stops when accuracy diminishes, that is, when eliminating more features decrements the performance of the induction algorithm.
- **Search strategy.** It follows a backward search strategy discarding features according to the sensitivity indices of the AFN method. This elimination process has two advantages: first, it can discard several features at each step; and second, it allows for backtracking, and so, when a subset of features worsens the results obtained by the previous one, some previously eliminated features can be included in the new subset for re-evaluation.
- **Evaluation function.** As it is a wrapper method, the induction algorithm is part of the evaluation function. This last is the accuracy reached by the induction algorithm with a specific number of features. However, it is not the accuracy achieved by one execution of the induction algorithm, but the mean accuracy obtained from a five-fold cross-validation [1].

The AFN-FS was modified to increase its scope of application, limited to the exponential complexity of function (4), by using an incremental approximation.

So, instead of estimating the desired output using the equation in (4), initially the desired output is estimated excluding the interactions between the features, and including only the univariate components:

$$\hat{y}_s = f_0 + \sum_{i=1}^{n} \sum_{j=1}^{k_i} c_{ij} p_j(\mathbf{x}_{si}), \tag{5}$$

where s is a specific sample, n the number of initial features and k_i the set of functions used to estimate the univariate component i. Once some features are discarded, the complexity of the approximation may be increased by adding more components (first, bivariate components, later, trivariate components and so on). The same process continues, discarding as many features as possible, but without decrementing the mean accuracy of the approximation.

The method developed was called IAFN-FS (Incremental AFN-Feature Selection), and has several interesting characteristics, such as:

- Feature selection is based on the sensitivity analysis the AFN algorithm carries out over the functional components of the approximate function. So, it can be said that the selection is *informed*, instead of the *blind* selection of most methods, that searches the space of all possible feature subsets, and selects the one obtaining the best performance by the induction algorithm.
- The sensitivity analysis allows for discarding several features in just one step. This is an interesting advantage, because the number of features subsets to be evaluated diminishes drastically, and so it does the number of executions needed for the induction algorithm, thus obtaining a considerable reduction in computational time.
- The IAFN-FS method takes into account not only the individual features, but also the possible interactions among them.
- The user can interpret the results obtained, because the relevance of each feature selected or rejected is given in terms of variance. This is a very interesting advantage for several fields, such as medical and biomedical, in which the acquisition of a data type can be very difficult, expensive or aggressive for the patient, or in real time applications, such as intrusion detection systems.

3 Multiclass Approaches

There are two main approaches for the classification problems involving more than two classes. One transforms the multiclass problem in several binary problems, while the other deals directly with the multiclass problem. This last strategy can present overtraining for those classes that are the majority in the dataset, or those classes that are easily separable [9]. Besides, some authors [10] defend that as the number of classes increases, it is less probable to obtain a good classification with a given feature set. However, the first strategy has also its drawbacks, such as how to integrate the information that comes from each of the binary classifiers, or the fact that there exists enough representation of a specific class in the training set generated from the original one to train the binary classifier.

The changes made in the basic algorithm described in section 2 for these two approaches are the following:

- In the multiclass approach, the output is coded using a distributed representation; that is, in a set with L classes, the output y_s is formed by L elements, and so $y_{sl} = 1$ if sample s belongs to class l and $y_{sl} = 0$, if it does not belong. Then, for a multiclass classification problem, the optimization problem is changed to:

  $$J = -\sum_{s} \sum_{l=1}^{L} \{y_{sl} \ln \hat{y}_{sl} + (1 - y_{sl}) \ln(1 - \hat{y}_{sl})\},$$

  where y_{sl} is the desired output and $\hat{y}_{sl}$ its estimation, indicating the probability of sample s belonging to class l. It has been considered that the sample belongs to the class with the highest $\hat{y}_{sl}$ value.
- The second approach transforms the multiclass problem in several binary problems. The possible different schemes for accomplishing this task can be generalized by the Error-Correcting Output-Codes (ECOC) method [11]. In this method, instead of providing each classifier with a set of inputs and its corresponding output, the output is transformed by using a matrix $M_{k \times c}$ which columns are the number of classes (c) and its rows the number of classifiers (k). The most usual schemes are the following [12]:
  - *one-versus-rest*: it transforms a problem with c classes in c binary problems, i.e., it uses one classifier for each class ($k = c$),
  - *one-versus-one*: it generates a classifier for each pair of class, therefore, there are $k = \frac{c(c-1)}{2}$ classifiers.

  In our case, the approach *one-versus-rest* has been chosen, since it is the most employed in the bibliography [4,13,14]. So, a classifier for each class is generated, confronting a specific class with the rest of the classes. Thus, if we are considering class l, the output of a sample y_{sl} could have the values $y_{sl} = 1$ if $y_s = l$ and $y_{sl} = 0$, in other cases. Finally, the optimum final feature subset is obtained as the union of all the subsets obtained for each of the classes. Analogously, the precision of the model is calculated as the average of the partial precisions for each class.

  Another problem of this approach is the need for enough number of samples in the training set. This aspect is critical in those cases in which there is a great difference between the number of samples of one class and the number of samples of others. In those cases, the capacity of generalization of the learning algorithm can be seriously affected, as it could end up ignoring the samples of the minority class, better than trying to learn it. In order to mitigate this problem, there exists different alternatives that consist on downsizing the majority class (*undersampling*), upsizing the minority class (*oversampling*) or altering the relative costs of misclassifying the small and the large classes [15]. For the sake of simplicity, an *undersampling* technique was employed in this work taking into account that we are dealing with a l *versus rest* approximation, so the minority class can be in both: the class l or one of the classes included in *rest*.

4 Experimental Results

There are several studies using different wrapper methods over several data sets. However, different validation techniques are used in each study, thus making it practically impossible to establish a fair comparison among the methods being evaluated. In our case, a comparative study with several classical algorithms such as C 4.5 and Naive-Bayes (available in Weka [16]), combined with a best-first strategy has been accomplished over several benchmark data sets (shown in Table 1). A 5-fold cross-validation was used for evaluating each candidate feature subset during the search process such it is done in [1], while a 10-fold cross-validation was employed to evaluate the overall performance of the wrappers.

Table 2 shows the results obtained using the multiclassifier strategy, while in Table 3 it can be seen the results obtained for multiple binary classifiers. In both, Accur. stands for accuracy and Feat. for the number of features selected. Moreover, bold font indicates the best results for each data set (mean accuracy and smallest number of features).

Concerning the multiclass (MC) approach, the accuracy obtained by our algorithm is higher than the obtained by the other methods in Iris and Vehicle data sets, and comparable in the remaining data sets. Therefore, it gets the best accuracy results in average and it achieves the best position in the ranking (last rows in Table 2). However, our method tends to select more features than the others.

Table 1. Description of the data sets. CV means cross-validation

Set	Training	Test	Features	Classes
Iris	150	CV	4	3
Zoo	101	CV	16	7
Wine	178	CV	13	3
Soybean-large	307	376	35	19
Vehicle	846	CV	18	4
Segment	2310	CV	19	7

Table 2. Results obtained for the multiclass classifiers (MC) approach

Data set	Naive-Bayes				C4.5				Prop. Alg.	
	NB-backward		NB-forward		C4.5-backward		C4.5-forward			
	Accur.	Feat.	Accur.	Feat.	Accur.	Feat.	Accur.	Feat.	Accur.	Feat.
Iris	94.67	1.7	94.67	1.7	94.67	**1.4**	94.00	**1.4**	**96.67**	2.1
Zoo	95.00	8.0	**96.00**	5.7	91.00	6.5	89.00	**5.2**	95.00	10.6
Wine	**97.78**	8.5	96.67	8.1	92.12	**4.0**	90.46	4.1	96.63	9.6
Soybean	92.23	17.0	**92.57**	14.0	91.55	**12.0**	91.55	13.0	91.89	24.0
Vehicle	56.63	**9.4**	57.23	9.5	71.29	13.4	71.53	10.9	**75.18**	15.2
Segment	90.13	**7.4**	88.74	7.7	**96.71**	10.9	96.62	7.7	91.47	14.1
Media	87.74	8.67	87.65	7.78	89.56	8.03	88.86	**7.05**	**91.14**	12.6
Ranking	2.67	2.83	2.5	2.5	3.00	2.33	3.83	**1.83**	**2.17**	5.00

Table 3. Results obtained for the multiple binary classifiers (MBC) approach

Data set	Naive-Bayes				C4.5				Prop.Alg.	
	NB-backward		NB-forward		C4.5-backward		C4.5-forward			
	Accur.	Feat.	Accur.	Feat.	Accur.	Feat.	Accur.	Feat.	Accur.	Feat.
Iris	96.67	2.7	96.67	2.7	95.11	2.4	95.11	2.3	**97.78**	**2.1**
Zoo	95.90	4.9	**96.17**	6.8	95.17	4.4	95.17	**4.0**	94.77	5.1
Wine	**97.58**	11.0	97.17	9.5	94.56	**6.4**	94.35	6.9	96.46	11.0
Soybean	**96.49**	32.0	95.90	21.0	94.57	**17.0**	96.01	**17.0**	96.44	23.0
Vehicle	76.37	**14.5**	76.46	14.6	83.98	16.5	83.75	14.6	**85.73**	14.9
Segment	92.38	16.2	94.19	**12.9**	**98.18**	14.7	98.06	13.7	96.38	14.6
Media	92.56	13.55	92.76	11.25	93.60	10.23	93.74	**9.75**	**94.59**	11.78
Ranking	2.67	3.67	2.67	3.00	3.17	2.67	3.00	**1.67**	**2.50**	3.33

Regarding the multiple binary classifiers approach (MBC), our method obtains again the best results for Iris and Vehicle data sets, while behaves similarly to the other methods for the remaining data sets. Analogously, it obtains the best accuracy on average and it gets the first position in the ranking, but using a higher number of features, although the difference is not so notorious in this case.

In general, it can also be seen that any MBC wrapper outperforms the results of its analogous MC wrapper, but increasing the number of features selected. With the aim of checking if there are statistically significant differences between both approximations, an ANOVA test, under the normality hypothesis, or a Kruskal-Wallis test, in the contrary case, have been performed. Both tests with a significance level of 0.05 showed that MBC wrappers are significantly different in accuracy for several data sets such as Vehicle and Segment.

5 Conclusions and Future Work

In this paper a new wrapper approach for selecting features in multiclass classification problems has been proposed. Two alternatives were explored: a direct multiclass approach and a multiple binary classifiers approach. In general the latter obtains better results in accuracy, although it has the drawback of selecting a higher number of features. The explanation for the binary approach obtaining better results can be that in this case, the algorithm selects the best features for each class, and then proceeds to merge all of them to treat the multiclass problem. So, the number of selected features is higher (it is the sum of the features for each class), while the features for each class are the most appropriate for it.

The comparative study clearly shows that MCB approaches lead to better performance results than MC ones. Then, in order to improve the MCB results, a more sophisticated and more computationally expensive method would be used in future work: the *one versus one* schema. Moreover, different techniques should be tested to mitigate the unbalance problem, including oversampling techniques such as the SMOTE algorithm [17]. When compared with other methods, the proposed wrapper obtains better results on average but using more number of features. Besides, it requires a computationally expensive backward strategy, thus in future work, a forward strategy would be employed using as starting set of features those provided by a filter or ensemble of filters.

References

1. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence, Special issue on relevance 97(1-2), 273–324 (1997)
2. Chiblovskii, B., Lecerf, L.: Scalable feature selection for multiclass problems. In: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2008, pp. 227–240 (2008)
3. Bruzzone, L., Serpico, S.: A technique for feature selection in multiclass problems. Int. J. of Remote Sensing 21(3), 549–563 (2000)
4. Bosin, A., Dessí, N., Pes, B.: Capturing heuristic and intelligents methods for improvinf micro-array data classification. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 790–799. Springer, Heidelberg (2007)
5. Sánchez-Maroño, N., Alonso-Betanzos, A.: Feature selection based on sensitivity analysis. In: Borrajo, D., Castillo, L., Corchado, J.M. (eds.) CAEPIA 2007. LNCS, vol. 4788, pp. 239–248. Springer, Heidelberg (2007)
6. Castillo, E., Cobo, A., Gutiérrez, M., Pruneda, E.: Functional Networks with Applications. Kluwer Academic Publishers, Dordrecht (1998)
7. Sobol, I.M.: Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Mathematics and Computers in Simulation 55, 271–280 (2001)
8. Castillo, E., Sánchez-Maroño, N., Alonso-Betanzos, A., Castillo, M.: Functional network topology learning and sensitivity analysis based on anova decomposition. Neural Computation 19(1) (2007)
9. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research, 1289–1305 (2003)
10. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature extraction. Foundations and applications. Springer, Heidelberg (2006)
11. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output code. Journal of Artificial Intelligence Resarch 2, 263–285 (1995)
12. Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., Hall, M.: Multiclass alternating decision trees. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS, vol. 2430, p. 161. Springer, Heidelberg (2002)
13. Friedman, J.: Additive logistic regresion: a statistical view of boosting. The Annals of Statistic 28(2), 337–374 (2000)
14. Oza, N., Tumer, K.: Input decimation ensembles: decorrelation through dimensionality reduction. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, p. 238. Springer, Heidelberg (2001)
15. Jaopkowicz, N., Stephen, S.: The class imbalance problem: A system study. Intelligent Data Analysis 6(5) (2002)
16. Witten, H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)
17. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)

New Challenges in the Application of Non-classical Formal Languages to Linguistics

Gemma Bel-Enguix[*] and M. Dolores Jiménez-López

Research Group on Mathematical Linguistics,
Rovira i Virgili University
Av. Catalunya, 35, 43002 Tarragona, Spain
Phone: + 34 977559542; Fax: + 34 977 559597
{gemma.bel,mariadolores.jimenez}@urv.cat

Abstract. We provide a short overview on the application of non-classical formal languages to linguistics. The paper focuses on the main achievements on formal language theory that have some impact on the description and explanation of natural language, points out the lack of mathematical tools to deal with some general branches of linguistics and suggests several computational theories that can help to construct a computational-friendly approach to language structure.

1 Introduction

The progressive widening of the models of computation implies a change in the scope of the problems they can tackle, as well as in the models of cognition they suggest. The first generation of formal languages, fitted into the Chomskian hierarchy, were based on rewriting, and caused the generalization of the tree-like models for describing and explaining natural language.

Contextual grammars were among the first computing devices that did not use rewriting. With them, new typologies of languages emerged, not exactly corresponding to the Chomskian hierarchy, that gave rise to new perspectives in the formalization of natural language.

A further step in the development of formal languages was the idea of several devices collaborating for achieving a common goal, represented by grammar systems. This new model, besides providing new horizons to formal linguistics, allowed, for the first time, the possibility of approaching syntax and natural language processing with formal methods. With the interaction of several independent, but communicated modules, grammar systems favored the emergence of both cognitive modeling and language generation simulations in a consistent formal framework.

In the last decades natural computing has become the most extended framework where new models for formal language theory have been developed. Natural computing refers to the study of the computational power of the processes

---

[*] This work was partially supported by DGUI CAM/UAM, project CCG08-UAM/TIC-4425.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 464–471, 2009.
© Springer-Verlag Berlin Heidelberg 2009

observed in nature, being these biologically inspired or not. The main reason for adopting such theoretical perspective is the need to reach a more realistic human-designed computing, both understanding the processes the nature carries out and taking advantage of the natural mechanisms that science is discovering.

Among the most spread theories in the field of natural computing, artificial immune systems, swarm intelligence and molecular and cell computing can be highlighted as the most fruitful and promising areas.

The first contributions to the field of artificial immune systems (AIS) date from the eighties [12] and nineties [5]. The model is based in the observation of the structure and function of the immune system, that can be successfully used to the resolution of computational problems, with direct applications to engineering and information technology.

Ant systems [11] are the most spread models within the general area of swarm computing [6]. This framework is chiefly related to artificial intelligence, engineering and robotics, but it has two features that are quite suggestive for formal language theory: a) collective behavior of decentralized and self-organized systems, and b) environment interaction of the system as a whole.

Nevertheless, the greatest revolution for both, computer scientists, and theoretical linguists, has come form the area of biomolecular computing. In the nineties, the emergence of biology as a pilot science had a direct impact in theoretical computer science by means of molecular computing, DNA computing and cell computing. Computability models taken directly from nature, more precisely from the behavior of DNA replication and communities of cells, provided a natural formalized model for explaining another natural communication system: natural language.

In the sequel, we explain the main achievements of formal language theory for linguistics, that have been carried out in the areas of contextual grammars, grammar systems and bioinspired computation.

2 Contextual Grammars

Most current natural language approaches show several facts that somehow invite to the search of new formalisms to account in a simpler and more natural way for natural languages. Two main facts lead us to look for a more natural computational system to give a formal account of natural languages: a) natural language sentences cannot be placed in any of the families of the Chomsky hierarchy in which current computational models are basically based, and b) rewriting methods used in a large number of natural language approaches seem to be not very adequate, from a cognitive perspective, to account for the processing of language.

Contextual grammars were introduced in [13] and are based on the idea of modeling some natural aspects from descriptive linguistics, as for example the acceptance of a word only in certain contexts. A contextual grammar produces a language starting from a finite set of words (axioms) and iteratively adding contexts (pair of words) to the currently generated words. Unlike Chomsky grammars, contextual grammars do not involve nonterminals and they do not have derivation rule. The have just one general rule, namely, to adjoin contexts. There have been defined many different types of contextual grammars, but all of them are based on context adjoining.

It is paradoxical that although contextual grammars were motivated by natural languages, they have been studied for about 25 years as a mathematical object, without exploiting their linguistic relevance. Research on contextual grammars have shown the mathematical richness of this theory. But, there still being a need of more research on the adequacy of this device for natural language description and modeling.

3 Grammar Systems

One of the most persistent issues in linguistics has been the question of determining where natural languages are located on 'Chomsky hierarchy.' Debate started several years ago, but the question still open in linguistics: can natural languages be characterized using context-free grammars or can't they? Several attempts have been made to prove the non-context- freeness of natural languages. Many authors have brought as demonstration of the non-context-freeness of natural languages examples of structures that are present in natural languages and that cannot be described using a context-free grammar.

Grammar systems can be considered a tool that could allow us to produce the non-context-free structures present in natural languages using context-free grammars. Easy generation of *non-context-free structures* using context-free rules, *modularity*, *parallelism* and *interaction* are some of the reasons that, we think, could support the suitability of grammar systems in linguistics. But we could adduce other defining features of such systems that could enable them to be applied to a large range of topics in linguistics, from pragmatics to language change. *Cooperation*, *communication*, *distribution*, *interdependence among parts*, *emergent behavior,* etc. are -in addition to the above-mentioned- features that make grammar systems a potentially good candidate to account for linguistic issues.

What is a grammar system? Roughly speaking, a grammar system is a set of grammars working together, according to a specified protocol, to generate a language. Note that while in classical formal language theory *one* grammar (or automaton) works individually to generate (or recognize) *one* language, here we have *several* grammars working together in order to produce *one* language. There are two basic classes of grammar systems: a) *Cooperating Distributed Grammar Systems* (CDGS) which work in a sequential way [8] and 2) *Parallel Communicating Grammar Systems* (PCGS) which function in parallel [17].

A CDGS consists of a finite set of generative grammars that cooperate in the derivation of a common language. Component grammars generate the string in turns (thus, sequentially), under some cooperation protocol. The basic model of CDGS presents sequentiality in its work and homogeneity in the cooperation protocol, however, variants have been introduced that have some parallelism in their function (teams) and that change the initial homogeneity into heterogeneity of modes of cooperation (hybrid systems). The basic model has been extended, also, by the addition of 'extra' control mechanisms.

A PCGS consists of several usual grammars, each with its own sentential form. In each time unit, each component uses a rule, which rewrites the associated sentential form. Cooperation among agents takes place thanks to the so-called query symbols that permit communication among components. In PCGS, we also find variants of the basic model. Modifications in the type of components or in the way strings are communicated produce several new types of PCGS (cf. [10]). For more information and formal results about all these variants see [9] and [10].

4 BioInspired Models of Computation

We have already highlighted the difficulty of fitting natural language in Chomsky hierarchy. Now, if to this we add: 1) that languages that have been generated following a molecular computational model are placed in-between CS and CF families; 2) that genetic model offers simpler alternatives to the rewriting rules; 3) and that genetics is a natural informational system as natural language is, we have the ideal scene to propose biological models in linguistics.

The idea of using biological methods in the description and processing of natural languages is backed up by a long tradition of interchanging methods in biology and natural/formal language theory. Some examples can be Pawlak dependency grammars; transformational grammars for modeling gene regulations; stochastic context-free grammars for modeling RNA; definite clause grammars and cut grammars to investigate gene structure and mutations and rearrangement in it; tree-adjoining grammars used for predicting RNA structure of biological data, etc.

Here, we present an overview of different bio-inspired methods that during the last years have been successfully applied to several natural language issues: DNA computing, membrane computing and networks of evolutionary processors.

4.1 DNA Computing

One of the most developed lines of research in natural computing is *molecular computing*, a model based on molecular biology, which arose mainly after [1]. An active area in molecular computing is *DNA computing* [16] inspired in the way DNA performs operations to generate, replicate or change the configuration of the strings.

Application of molecular computing methods to natural language syntax gives rise to *molecular syntax* [3]. Molecular syntax takes as a model two types of mechanisms used in biology (especially in genetic engineering) in order to modify or generate DNA sequences: *mutations* and *splicing*. Mutations refer to changes performed in a linguistic string, being this a phrase, sentence or text. Splicing is a process carried out involving two or more linguistic sequences. It is a good framework for approaching syntax, both from the sentential or dialogical perspective.

Methods used by molecular syntax are based on basic genetic processes: *cut, paste, delete* and *move*. Combining these elementary rules most of the complex structures of natural language can be obtained, with a high degree of simplicity.

This approach is a test of the generative power of splicing for syntax. It seems, according to the results achieved, that splicing is quite powerful for generating, in a very simple way, most of the patterns of the traditional syntax. Moreover, the new perspectives and results it provides, could mean a transformation in the general perspective of syntax.

From here, we think that bio-linguistics, applied in a methodological and clear way, is a powerful and simple model that can be very useful a) to formulate some systems capable of generating the larger part of structures of language, and b) to define a formalization that can be implemented and may be able to describe and predict the behavior of natural language structures.

4.2 Membrane Computing

Membrane systems –introduced in [15]– are models of computation inspired by some basic features of biological membranes. They can be viewed as a new paradigm in the field of natural computing based on the functioning of membranes inside the cell. Membrane systems can be used as generative, computing or decidability devices. This new computing model has several intrinsically interesting features such as, for example, the use of multisets and the inherent parallelism in its evolution and the possibility of devising computations which can solve exponential problems in polynomial time.

Linguistic Membrane Systems (LMS) [4] aim to model linguistic processes, taking advantage of the flexibility of membrane systems and their suitability for dealing with some fields where contexts are a central part of the theory. LMS can be easily adapted to deal with different aspects of the description and processing of natural languages. The most developed applications of LMS are semantics and dialogue.

Membrane systems are a good framework for developing a semantic theory because they are evolving systems by definition, in the same sense that we take meaning to be a dynamic entity. Moreover, membrane systems provide a model in which contexts, either isolated or interacting, are an important element which is already formalized and can give us the theoretical tools we need.

Membrane systems have also been applied to the design of effective and user-friendly computer dialogue systems. Taking into account a pragmatic perspective of dialogue and based on speech acts, multi-agent theory and dialogue games, Dialogue Membrane Systems have arisen, as an attempt to compute speech acts by means of membrane systems. Considering membranes as agents, and domains as a personal background and linguistic competence, the application to dialogue is almost natural, and simple from the formal point of view.

4.3 NEPS-Networks of Evolutionary Processors

Networks of Evolutionary Processors (NEPs) are a new computing mechanism directly inspired in the behavior of cell populations. Every cell is described by a set of words (DNA) evolving by mutations, which are represented by operations on these words. At the end of the process, only the cells with correct strings will survive. In spite of the biological inspiration, the architecture of the system is directly related to the connection machine and the logic flow paradigm. Moreover, the global framework for the development of NEPs has to be completed with the biological background of DNA computing [16], membrane computing [15] – that focalizes also in the behavior of cells – and specially with the theory of grammar systems [9], which share with NEPs the idea of several devices working together and exchanging results. NEPs were introduced in [7] and [14]. With all this background and theoretical connections, it is easy to understand how NEPs can be described as agential bio-inspired context-sensitive systems. Many disciplines are needed of these types of models that are able to support a biological framework in a collaborative environment. The conjunction of these features allows applying the system to a number of areas, beyond generation and recognition in formal language theory.

NEPs have significant intrinsic multi-agent capabilities together with the environmental adaptability that is typical of bio-inspired models. Some of the characteristics of NEPs architecture are the following: *modularity*, *contextualization*, *synchronization*,*evolvability*, *learnability*. Inside of the construct, every agent is autonomous, specialized, context-interactive and learning-capable. Social competences among processors are specified by a graph-supported structure and filtering-regulated communication and coordination.

Because of those features, NEPs seems to be a quite suitable model for tackling natural languages [2]. Some linguistic disciplines, as pragmatics or semantics, are context-driven areas, where the same utterance has different meanings in different contexts. To model such variation, a system with a good definition of environment is needed. NEPs offer some kind of solution to approach formal semantics and formal pragmatics from a natural computing perspective.

Finally, the multimodal approach to communication, where not just production, but also gestures, vision and supra-segmental features of sounds have to be tackled, refers to a parallel way of processing. NEPs allow to work in the same goal, but in different tasks, to every one of the modules. The autonomy of every one of the processors and the possible miscoordination between them can also give account of several problems of speech.

5 Future Trends

The future trends on the application of formal language theory to linguistics are closely related to the introduction and further study of three main ideas: *evolution*, *context interaction* and *network structure*.

In what refers to the concept of *evolution* in formal language theory, the term has been, up to now, synonym for *change over the time*. Being this true, it implies that the main aspects of Darwinian evolution: random change by replication and natural selection, still await to be developed. This fact has an impact to theoretical linguistics, that cannot take advantage of the formalizations provided by mathematics.

Evolution is one of the chief processes natural languages undergo. The old metaphor of language as a biological system that evolves as a species is not necessary right. But, it is clear that the main methods introduced by Darwinian evolution should be tested as a methodology for approaching diachronic aspects of natural language. A consistent theoretical framework as well as an efficient computer simulation are promising lines of research for historical linguistics in the future.

When approaching language evolution, an important related aspect that cannot be dismissed is *cultural evolution.* Grammar systems provide interesting tools for starting this line of research, that could be integrated with networks of evolutionary processors in order to account for this aspect, that have a direct impact on the change of language structures over the time.

Context is again one of the most important areas of research that need to be improved in linguistics in general. It is clear that natural language comprehension lies not only in the description of its internal structures, but also in the relationship between utterances and the context in which they are produced and interpreted. The main concepts of *context*, *context awareness* and *context interaction* are currently being well developed in swarm intelligence, that exploits the possibilities of interaction among individuals and environment in a more optimal way. Especially, the behavior of the swarm as a whole in contrast with the single-agent behavior can help to analyze, formalize and simulate important parts of linguistics like semantics and pragmatics.

The idea of *network* is becoming one of the key points not only in science, but also in the society. Net-working, net-thinking and net-solving can be important parameters in the future. Also language is seen as a collaborative construct, generated thanks to the participation of different cognitive capabilities and the convergence of several symbolic and evolutionary capacities. The idea of net was implicit in grammar systems, but it failed in arising due to the lack of development of the initial theory. Finally, NEPs provided a first formalization of symbolic network that allows a different type of interaction between the agents of systems, since them can interact not only with the environment, but among them.

Artificial immune systems can also have a contribution to network formalization, by means of immune network algorithms, that provide a framework for regulation of immune systems, easily applicable to some phenomena, like language generation or textual relations.

Therefore, the full development of networking theories in bioinspired formal languages can help to tackle some parts of linguistics requiring interaction and context, like pragmatics and dialogue, which are crucial for artificial intelligence, and that do not have a suitable model for development so far. Cell computing, artificial immune systems and even swarm computing can help to develop this idea of symbolic network with contextual and environmental capabilities.

References

1. Adleman, L.M.: Molecular Computation of Solutions to Combinatorial Problems. Science 226, 1021–1024 (1994)
2. Bel-Enguix, G., Jiménez-López, M.D.: Analysing Sentences with Networks of Evolutionary Processors. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005. LNCS, vol. 3562, pp. 102–111. Springer, Heidelberg (2005)

3. Bel-Enguix, G., Jiménez-López, M.D.: Biosyntax. An Overview. Fundamenta Informaticae 64, 1–12 (2005b)
4. Bel-Enguix, G., Jiménez-López, M.D.: Linguistic Membrane Systems and Applications. In: Ciobanu, G., Păun, G., Pérez Jiménez, M.J. (eds.) Applications of Membrane Computing, pp. 347–388. Springer, Berlin (2005c)
5. Bersini, H., Varela, F.J.: Hints for adaptive problem solving gleaned from immune networks. In: Parallel Problem Solving from Nature, First Workshop PPSW 1, Dortmund, FRG (1990)
6. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems, Oxford (1999)
7. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Networks of Evolutionary processors. Acta Informatica 39, 517–529 (2003)
8. Csuhaj-Varjú, E., Dassow, J.: On Cooperating/Distributed Grammar Systems. Journal of Information Processing and Cybernetics (EIK) 26, 49–63 (1990)
9. Csuhaj-Varjú, E., Dassow, J., Kelemen, J., Păun, G.: Grammar Systems. Gordon and Breach, London (1994)
10. Dassow, J., Păun, G., Rozenberg, G.: Grammar Systems. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 2, pp. 155–213. Springer, Berlin (1997)
11. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics–Part B 26(1), 29–41 (1996)
12. Farmer, J.D., Packard, N., Perelson, A.: The immune system, adaptation and machine learning. Physica D 2, 187–204 (1986)
13. Marcus, S.: Contextual grammars. Revue Romaine des Mathématiques Pures et Appliquées 14, 1525–1534 (1969)
14. Martín-Vide, C., Mitrana, V., Pérez-Jiménez, M., Sancho-Caparrini, F.: Hybrid Networks of Evolutionary Processors. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 401–412. Springer, Heidelberg (2003)
15. Păun, G.: Computing with Membranes. Journal of Computer and System Sciences 61, 108–143 (2000)
16. Păun, G., Rozenberg, G., Salomaa, A.: DNA Computing. New Computing Paradigms. Springer, Berlin (1998)
17. Păun, G., Sântean, L.: Parallel Communicating Grammar Systems: The Regular Case', Annals of the University of Bucharest. Mathematics-Informatics Series 38, 55–63 (1989)

PNEPs, NEPs for Context Free Parsing: Application to Natural Language Processing[*]

Alfonso Ortega[1], Emilio del Rosal[2], Diana Pérez[1], Robert Mercaş[3], Alexander Perekrestenko[3], and Manuel Alfonseca[1]

[1] Dep. Ing. Inf. EPS Universidad Autónoma de Madrid, Spain
`alfonso.ortega@uam.es, diana.perez@uam.es, manuel.alfonseca@uam.es`
[2] Escuela Politécnica Superior Universidad San Pablo C.E.U, Spain
`emilio.rosalgarcia@ceu.es`
[3] GRLMC, Rovira i Virgili University, Dept. of Romance Filology, Tarragona, Spain
`robertgeorge.mercas@estudiants.urv.cat,`
`alexander.perekrestenko@estudiants.urv.cat`

Abstract. This work tests the suitability of NEPs to parse languages. We propose PNEP, a simple extension to NEP, and a procedure to translate a grammar into a PNEP that recognizes the same language. These parsers based on NEPs do not impose any additional constrain to the structure of the grammar, which can contain all kinds of recursive, lambda or ambiguous rules. This flexibility makes this procedure specially suited for Natural Languge Processing (NLP). In a first proof with a simplified English grammar, we got a performance (a linear time complexity) similar to that of the most popular syntactic parsers in the NLP area (Early and its derivatives). All the possible derivations for ambiguous grammars were generated.

1 NEPs for Natural Language Processing: Analysis

Computational Linguistics researches linguistic phenomena that occur in digital data. Natural Language Processing (NLP) is a subfield of Computational Linguistics that focuses on building automatic systems able to interpret or generate information written in natural language [1]. This is a broad area which poses a number of challenges, both for theory and for applications.

Machine Translation was the first NLP application in the fifties [2]. In general, the main problem found in all cases is the inherent ambiguity of the language [3].

A typical NLP system has to cover several linguistic levels:

- **Phonological:** Sound processing to detect expression units in speech.
- **Morphological:** Extracting information about words, such as their part of speech and morphological characteristics [4][5]. Best systems has an accuracy of 97% in this level [6].

---

[*] This work was partially supported by MEC, project TIN2008-02081/TIN and by DGUI CAM/UAM, project CCG08-UAM/TIC-4425.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 472–479, 2009.
© Springer-Verlag Berlin Heidelberg 2009

- **Syntactical:** Using parsers to detect valid structures in the sentences, usually in terms of a certain grammar. One of the most efficient algorithms is the one described by Earley and its derivatives [7,8,9]. It provides parsing in polynomial time, with respect to the length of the input (linear in the average case; n^2 and n^3, respectively, for unambiguous and ambiguous grammars in the worst case) This paper is focused on this step.
- **Semantic:** Finding the most suitable knowledge formalism to represent the meaning of the text.
- **Pragmatic:** Interpreting the meaning of the sentence in a context which makes it possibe to react accordingly.

The two last levels are still far from being solved [10]. Typical NLP systems usually cover the linguistic levels previously described in the following way:

⇒ Morphologycal analysis ⇒ Syntax analysis ⇒ Semantic interpretation ⇒
Discourse text processing ⇒ OCR/Tokenization

A computational model that can be applied to NLP tasks is a network of evolutionary processors (NEPs). NEP as a generating device was first introduced in [11] and [12]. The topic is further investigated in [13], while further different variants of the generating machine are introduced and analyzed in [14-18].

In [19], a first attempt was made to apply NEPs for syntactic NLP parsing. Our paper focuses the same goal: testing the suitability of NEPs to tackle this task. We have previously mentioned some performance characteristics of one of the most popular families of NLP parsers (those based on Early's algorithm). We will conclude that our approach has a similar complexity.

While [19] outlines a bottom up approach to natural language parsing with NEPs, in the present paper we suggest a top-down strategy and show its possible use in a practical application.

2 Introduction to jNEP

The jNEP [20] Java written program, freely available at http://jnep.edelrosal.net, can simulate almost all NEPs in the literature. The software has been developed under three main principles: 1) it rigorously complies with the formal definitions found in the literature; 2) it serves as a general tool, by allowing the use of the different NEP variants and can easily be adapted to possible future extensions of the NEP concept; 3) it exploits the inherent parallel/distributed nature of NEPs.

jNEP consists of three main clases (*NEP*, *EvoluionaryProcessor* and *Word*), and three Java interfaces (StoppingCondition, *Filter* and *EvolutionaryRule*)

The design of the NEP class mimics the NEP model definition. In jNEP, a NEP is composed of evolutionary processors and an underlying graph (attribute *edges*), used to define the net topology.

The *NEP* class coordinates the main dynamic of the computation and manages the processors (instances of the *EvolutionaryProcessor* class), forcing them to perform alternate evolutionary and communication steps. Furthermore, the computation is stoped whenever this is needed.

The Java interfaces are used for those components which more frequently change between different NEP variants: *StoppingCondition*, *Filter* and *EvolutionaryRule*. jNEP implements a wide set of these three components. More can easily be added in the future.

Currently *jNEP* has two lists of choices for the selection of the mode in which it runs: parallel or distributed. For the first list, concurrency is implemented by means of *Threads* and *Processes*, while in the second one, the supported platforms are standard JVM and clusters of computers (by means of JavaParty).

Depending on the operating system, the Java Virtual Machine used and the concurrency option chosen, jNEP will work in a slightly different manner. The users should select the best combination for their needs.

jNEP reads the definition of the NEP from an XML configuration file that contains special tags for any relevant component in the NEP (alphabet, stopping conditions, the complete graph, each edge, each evolutionary processor with its rules, filters and initial contents).

Although some fragments of these files will be shown in this paper, all the configuration files mentioned here can be found at (http://jnep.e-delrosal.net). Despite the complexity of these XML files, the interested reader can see that the tags and their attributes have self-explaining names and values. The reader may refer to the *jNEP user guide* for further detailed information.

3 Top Down Parsing with NEPs and jNEP

Other authors have previously studied the relationships between NEPs, regular, context-free, and recursively enumerable languages [14-18]. [21] shows how NEPs simulate the application of context free rules ($A \to \alpha, A \in V, \alpha \in V^*$ for alphabet V): a set of additional nodes is needed to implement a rather complex technique to rotate the string and locate A in one of the string ends, then delete it and adding all the symbols in α. PNEPs use context free rules rather than classic substitution ($A \to B, A, B \in V$), as well as insertion and deletion NEP rules. In this way, the expressive power of NEP processors is bounded, while providing a more natural and comfortable way to describe the parsed language for practical purposes.

PNEPs implement a top down parser for context free grammars. The parser is able to generate all the possible derivations of each string in the language generated by the grammar. Its temporal complexity is bounded by the length of the analyzed string. This bound can be used to stop the computation when processing incorrect strings, thus avoiding running the PNEP for a possible infinite number of steps.

The PNEP is built from the grammar in the following way: (1) We assume that each derivation rule in the grammar has a unique index that can be used to reconstruct the derivation tree. (2) There is a node for each non terminal. Each node applies to the strings all the derivation rules for its non terminal.

The filters, as well as the graph layout, allow all the nodes to share all the intermediate steps in the derivation process. (3) There is an additional output node, in which the *parsed string* can be found: this is a version of the input, enriched with information that will make it possible to reconstruct the derivation tree (the rules indices). (4) The graph is complete.

Obviously the same task can be performed using a trivial PNEP with only a node for all the derivation rules. However, the proposed PNEP is easier to analyze and more usefull to distribute the work among several nodes.

We shall consider as an example the grammar $G_{a^n b^n c^m}$ induced by the following derivation rules (notice that indexes have been added in front of the corresponding right hand side):

$$X \Rightarrow (1)SO \ , \ S \Rightarrow (2)aSb|(3)ab \ , \ O \Rightarrow (4)Oo|(5)oO|(6)o$$

It is easy to prove that the language corresponding to this grammar is $\{a^n b^n o^m \mid n, m > 0\}$. Furthermore, the grammar is ambiguous, since every sequence of o symbols can be generated in two different ways: by producing the new terminal o with rule 4 or with rule 6.

The input filters of the output node describe parsed copies of the initial string. In other words, strings whose symbols are preceded by strings of any length (including 0) of the possible rules indexes. As an example, a parsed version of the string $aabboo$ would be $12a3abb5o6o$.

We will assume in this paper that a PNEP Γ is formally defined as follows:

$\Gamma = (V, N_1, N_2, ..., N_n, G)$, where V is an alphabet and for each $1 \leq i \leq n$, $N_i = (M_i, A_i, PI_i, PO_i)$ is the i-th evolutionary node processor of the network. The parameters of every processor are:

- M_i is a finite set of context-free evolution rules
- A_i is the set of initial strings in the i-th node.
- PI_i and PO_i are subsets of V^* representing respectively the input and the output filters. These filters are defined by the membership condition, namely a string $w \in V^*$ can pass the input filter (the output filter) if $w \in PI_i (w \in PO_i)$. In this paper we will use two kind of filters:
  - Those defined as two components (P, F) of *P*ermitting and *F*orbidding contexts (a word w passes the filter if $(\alpha(w) \subseteq P) \wedge (F \cap \alpha(w) = \Phi)$).
  - Those defined as regular expressions r (a word w passes the filter if $w \in L(r)$, where $L(r)$ stands for the language defined by the regular expression r).

Finally, $G = (N_1, N_2, ..., N_n, E)$ is an undirected graph (whose edges are E), called the underlying graph of the network.

We will now describe the way in which our PNEP is defined, starting from a certain grammar. Given the context free grammar $G = \{\Sigma_T = \{t_1, ..., t_n\}, \Sigma_N = \{N_1, ..., N_m\}, A, P\}$ with $A \in \Sigma_N$ its axiom and $P = \{l_j \rightarrow \gamma_j \mid j \in \{1, .., k\}, l_j \in \Sigma_N \wedge \gamma_j \in (\Sigma_T \cup \Sigma_N)^*\}$ its set of k production rules the PNEP is defined as

$$\Gamma_G = (V = \Sigma_T \cup \Sigma_N \cup \{1, ..., k\}, node_{output}, N_1, N_2, ..., N_m, G)$$

where (1) $node_{output}$ is the output node; (2) G is a complete graph and (3) the *input node* A, the only one with a non empty initial content (A). Each non terminal node N_i in the PNEP has a substitution rule for each derivation rule in the grammar applicable to it. This rule changes the nonterminal by a string made by uppending the right hand side of the derivation rule with the index of the rule in P. For example, the PNEP for grammar $G_{a^nb^no^m}$ described above has a node for nonterminal S with the following substitution rules: $\{S \rightarrow 2aSb, S \rightarrow 3ab\}$ The input filters of these nodes allow all strings containing some copy of their non terminal to input the node. Strings that do not contain a copy of the non terminal pass the output filter. We can get this behavior by using the set with the non terminal symbol of the node ($\{N_i\}$) both as the permitted input filter and as the forbidden output filter.

The input filter for the output node $node_{output}$ has to describe what we have called *parsed strings*. Parsed strings will contain numbers, corresponding to the derivation rules which have been applied, among the symbols of the initial string. We can easily create a regular expression and define the input filter by means of membership. For example, in order to parse the string *aabbo* with the grammar we are using above, the regular expression can be $\{1,2,3,4,5,6\}* a\{1,2,3,4,5,6\}*a\{1,2,3,4,5,6\}*b\{1,2,3,4,5,6\}*b\{1,2,3,4,5,6\}*o$. Our PNEP will stop computing whenever a string enters the output node.

The complete PNEP for our example ($\Gamma_{a^nb^no^m}$) is defined as follows:

- Alphabet $V = \{X, O, S, a, b, o, 1, 2, 3, 4, 5, 6\}$
- Nodes
  - $node_{output}$: $A_{output} = \Phi$ is the initial content; $M_{output} = \Phi$ is the set of rules; $PI_{output} = \{$ (regular expression membership filter); $\{\{1,2,3,4,5,6\} * a\{1,2,3,4,5,6\} * a\{1,2,3,4,5,6\} * b\{1,2,3,4,5,6\} * b\{1,2,3,4,5,6\} * o\}\}$; $PO_{output} = \Phi$ is the output filter
  - N_X: $A_X = \{X\}$; $M_X = \{X \rightarrow 1SO\}$; $PI_X = \{P = \{X\}, F = \Phi\}$; $PO_X = \{F = \{X\}, P = \Phi\}$
  - N_S: $A_S = \Phi$; $M_S = \{S \rightarrow 2aSb, S \rightarrow 3ab\}$; $PI_S = \{P = \{S\}, F = \Phi\}$; $PO_S = \{F = \{S\}, P = \Phi\}$
  - N_O: $A_O = \Phi$; $M_O = \{O \rightarrow 4oO, O \rightarrow 5Oo, O \rightarrow 5o\}$; $PI_O = \{P = \{O\}, F = \Phi\}$; $PO_O = \{F = \{O\}, P = \Phi\}$
  - It has a complete graph
  - It stops the computation when some string enters $node_{output}$

Some of the strings generated by all the nodes of the PNEP in succesive communication steps when parsing the string *aboo* are shown below (each set corresponds to a different step): $\{X\} \Rightarrow \{1SO\} \Rightarrow \{..., 13abO, ...\} \Rightarrow$ $\{..., 13ab4Oo, 13ab5oO, ...\} \Rightarrow \{..., 13ab46oo, ..., 13ab5o6o, ...\}$

The last set contains two different derivations for *aboo* by ($G_{a^nb^nno^m}$), that can enter the output node and stop the computation of the PNEP.

It is easy to reconstruct the derivation tree from the parsed strings in the output node, by following their sequence of numbers. For example, consider the parsed string $13ab6o$ and its sequence of indexes 136; *abo* is generated in the following steps: $X \Rightarrow$ (rule 1 $X \Rightarrow SO$) SO , $SO \Rightarrow$ (rule 3 $S \Rightarrow ab$) abO , $abO \Rightarrow$ (rule 6 $O \Rightarrow o$) abo

In section 2 we have described the structure of the xml input files for *jNEP*. Two the sections of the xml representation of the NEP $\Gamma_{a^n b^n o^m}$ (the output node and the node for axiom X) are shown below.

```
<NODE initCond="">
   <EVOLUTIONARY_RULES>
   </EVOLUTIONARY_RULES>
   <FILTERS>
     <INPUT type="RegularLangMembershipFilter" regularExpression="[1-6]*a[1-6]*b[1-6]*o[1-6]*o"/>
     <OUTPUT type="1" permittingContext="" forbiddingContext=""/>
   </FILTERS>
 </NODE>

 <NODE initCond="X">
   <EVOLUTIONARY_RULES>
     <RULE ruleType="substitution" actionType="ANY" symbol="X" newSymbol="1_S_0"/>
   </EVOLUTIONARY_RULES>
   <FILTERS>
     <INPUT type="1" permittingContext="X" forbiddingContext=""/>
     <OUTPUT type="1" permittingContext="" forbiddingContext="X"/>
   </FILTERS>
 </NODE>
```

The nodes for other non terminal symbols are similar, but with an empty ("") *initial condition* and their corresponding derivation rules.

4 An Example for Natural Language Processing

As described in section 1, the complexity of the grammars used for syntactic parsing depends on the desired target. These grammars are usually very complex, which makes them one of the bottlenecks in NLP tasks.

We will use the grammar deduced from the following derivation rules (whose axiom is the non terminal Sentence). This grammar is similar to those devised by other authours in previous attempts to use NEPs for parsing (natural) languages [19]. We have added the index of the derivation rules, that will be used later.

Sentence → (1) NounFraseStandard PredicateStandard
| (2) NounFrase3Singular Predicate3Singular
NounFrase3Singular → (3) DeterminantAn VowelNounSingular
| (4) DeterminantSingular NounSingular
| (5) Pronoun3Singular
NounFraseStandard → (6) DeterminantPlural NounPlural
| (7) PronounNo3Singular
NounFrase → (8) NounFrase3Singular | (9) NounFraseStandard
PredicateStandard → (10) VerbStandard NounFrase
Predicate3Singular → (11) Verb3Singular NounFrase
DeterminantSingular → (12) a | (13) the | (14) this
DeterminantAn → (15) an
VowelNounSingular → (16) apple
NounSingular → (17) boy
Pronoun3Singular → (18) he | (19) she | (20) it
DeterminantPlural → (21) the | (22) several | (23) these
NounPlural → (24) apples | (25) boys
PronounNo3Singular → (26) I | (27) you | (28) we | (29) they
VerbStandard → (30) eat
Verb3Singular → (31) eats

It is worth noticing that this grammar is very simple. As described in section 1, NLP syntax parsing usually takes as input the results of the morphological analysis. In this way, the previous grammar can be simplified by removing the derivation rules for the last 9 non terminals (from DeterminantSingular to Verb3Singular): those symbols become terminals for the new grammar.

Notice, also, that this grammar implements grammatical agreement by means of context free rules. For each non terminal, we had to use several different *specialized versions*. For instance, NounFraseStandard and NounFrase3Singular are specialized versions of non terminal NounFrase. These rules increases the complexity of the grammar.

We can build the PNEP associated with this context free grammar by following the steps described in section 3.

Let us consider the English sentence *the boy eats an apple* Some of the strings generated by the nodes of the PNEP in succesive communication steps while parsing this string are shown below (we show the initials, rather than the full name of the symbols). A left derivation of the string is highlighted: { S } ⇒ { ..., 2 NF3S P3S, ... } ⇒ { ..., 2 4 DS NS P3S, ... } ⇒ { ..., 2 4 13 the NS P3S, ... } ⇒ { ..., 2 4 13 the 17 boy P3S, ... } ⇒ { ..., 2 4 13 the 17 boy 11 V3S NF, ... } ⇒ { ..., 2 4 13 the 17 boy 11 31 eats NF, ... } ⇒ { ..., 2 4 13 the 17 boy 11 31 eats 8 NF3S, ... } ⇒ { ..., 2 4 13 the 17 boy 11 31 eats 8 3 DA VNS, ... } ⇒ { ..., 2 4 13 the 17 boy 11 31 eats 8 3 15 an VNS, ... } ⇒ { ..., 2 4 13 the 17 boy 11 31 eats 8 3 15 an 16 apple, ... }

If we analyze now an incorrect sentence, such as *the boy eat the apple*, the PNEP will continue the computation after the steps summarized above, because in this case it is impossible to find a parsed string. It is easy to modify our PNEP to stop when this circumstance happens.

5 Conclusions and Further Research Lines

We have found that a very simple and straightforward use of NEPs, associated to context-free grammars, results in a parsing technique with similar characteristics to the most popular syntactic parsers in the NLP area. PNEPs seem to be an appropiate and natural approach to tackle both parsing of context-free languages and the syntactic processing of natural languages.

In the future we plan to use a more realistic grammar for NLP purposes; to incorporate this technique into a linguistic corpus in which we are currently working; to test our approach with realistic NLP benchmarks; and to explore the possible design of compiler tools based on PNEPs.

References

1. Volk, M.: Introduction to Natural Language Processing, Course CMSC 723 / LING 645 in the Stockholm University, Sweden (2004)
2. Weaver, W. (55), Translation, Machine Translation of Languages: Fourteen Essays, 15–23
3. Mitkov, R.: The Oxford Handbook of Computational Linguistics. Oxford University Press, Oxford (2003)
4. Mikheev, A.: Periods, capitalized words, etc. Computational Linguistics 28(3), 289–318 (2002)

5. Alfonseca, E.: An Approach for Automatic Generation of on-line Information Systems based on the Integration of Natural Language Processing and Adaptive Hypermedia techniques, PhD thesis, Computer Science Department, UAM (2003)
6. Brants, T.: TnT–a statistical part-of-speech tagger. In: Proceedings of the 6th Conference on Applied Natural Language Processing, pp. 224–231 (2000)
7. Earley, J.: An efficient context-free parsing algorithm. Communications of the ACM 13(2), 94–102 (1970)
8. Seifert, S., Fischer, I.: Parsing String Generating Hypergraph Grammars, pp. 352–367. Springer, Heidelberg (2004)
9. Zollmann, A., Venugopal, A.: Syntax augmented machine translation via chart parsing. In: Proccedings of the Workshop on Statistic Machine Translation, HLT/NAACL, New York (June 2006)
10. Gomez, C., Javier, F., Valle Agudo, D., Rivero Espinosa, J., Cuadra Fernandez, D.: Methodological approach for pragmatic annotation. Procesamiento del lenguaje Natural, 209–216 (2008)
11. Csuhaj-Varju, E., Salomaa, A.: Networks of parallel language processors. In: Păun, G., Salomaa, A. (eds.) New Trends in Formal Languages. LNCS, vol. 1218. Springer, Heidelberg (1997)
12. Csuhaj-Varju, E., Mitrana, V.: Evolutionary systems:a language generating device inspired by evolving communities of cells. Acta Informatica 36 (2000)
13. Castellanos, J., Martin-Vide, C., Mitrana, V., Sempere, J.M.: Solving np-complete problems with networks of evolutionary processors. In: Mira, J., Prieto, A.G. (eds.) IWANN 2001. LNCS, vol. 2084, pp. 621–628. Springer, Heidelberg (2001)
14. Castellanos, J., Leupold, P., Mitrana, V.: On the size complexity of HNEPs. Theoretical Computer Science 330(2), 205–220 (2005)
15. Manea, F.: Using ahneps in the recognition of context-free languages. In: Proceedings of the Workshop on Symbolic Networks, ECAI 2004 (2004)
16. Manea, F., Mitrana, V.: All np-problems can be solved in polynomial time by AHNEPs of constant size. Inf. Process. Lett. 103(3), 112–118 (2007)
17. Margenstern, M., Mitrana, V., Perez-Jimenez, M.: Accepting hybrid networks of evolutionary processors. In: Pre-proceedings of DNA, vol. 10, pp. 107–117 (2004)
18. Martin-Vide, C., Mitrana, V., Perez-Jimenez, M., Sancho-Caparrini, F.: Hybrid networks of evolutionary processors. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 401–412. Springer, Heidelberg (2003)
19. Bel Enguix, G., Jimenez-Lopez, M.D., Mercaş, R., Perekrestenko, A.: Networks of evolutionary processors as natural language parsers. In: Proc. ICAART (2009)
20. del Nuez, E., Castaeda, R., Ortega, C.: Simulating NEPs in a cluster with jNEP. In: Proceedings of International Conference on Computers, Communications and Control, ICCCC 2008 (2008)
21. Csuhaj-Varju, E., Martin Vide, C., Mitrana, V.: HNEPs are Computationally Complete. Acta Informatica

A Hyprolog Parsing Methodology for Property Grammars*

Veronica Dahl[1,2], Baohua Gu[1], and Erez Maharshak[1,2]

[1] School of Computing Science, Simon Fraser University,
Burnaby, B.C., Canada, V5A 1S6
[2] Departament de Filologies Romàniques, Universitat Rovira i Virgili
Av. Catalunya 35, 43002 Tarragona, Spain
{veronica,bgu}@cs.sfu.ca, emaharsh@sfu.ca

Abstract. *Property Grammars*, or PGs, belong to a new family of linguistic formalisms which view a grammar as a set of linguistic constraints, and parsing as a constraint satisfaction problem. Rigid hierarchical parsing gives way to flexible mechanisms which can handle incomplete, ambiguous or erroneous text, and are thus more adequate for new applications such as speech recognition, internet mining, controlled languages and biomedical information. The present work contributes a) a new parsing methodology for PGs in terms of Hyprolog – an extension of Prolog with linear and intuitionistic logic and with abduction; and b) a customisable extension of PGs that lets us model also concepts and relations to some degree. We exemplify within the domain of extracting concepts from biomedical text.

Keywords: Hyprolog, parsing, text retrieval, Semantic Property Grammars, concept extraction.

1 Introduction

New approaches to parsing include substantial efforts to obtain useful results even for incomplete or ungrammatical input. More flexible models than the traditional, hierarchical parsing models have emerged around them, largely motivated by new developments in related areas: speech recognition systems have matured enough to be helpfully embedded in spoken command systems, e.g. for the handicapped; the unprecedented explosion of text available online since the internet's advent calls for ultra-flexible text processing schemes, geographically distributed enterprises are evolving controlled languages as a kind of interlingua in which their employees from different countries can communicate despite possible errors and imperfections. Even outside such specialized needs, a parser

---

[*] An earlier version of this paper has been presented "On Semantically Constrained Property Grammars", Proceedings of the 5th International Workshop on Constraints and Language Processing (CSLP 2008), Villadsen, J., Christiansen, H., (Eds), ROSKILDE UNIVERSITY COMPUTER SCIENCE RESEARCH REPORT #122, 2008, pp. 20–32.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 480–487, 2009.
© Springer-Verlag Berlin Heidelberg 2009

should ideally be able to extract meaning from text produced in real life conversation, which typically is underspecified, often not perfectly grammatical, and sometimes erroneous. Imperfections can result from normal human error in actual speech, or be introduced by machines, as in the case of text produced from speech recognition systems, which, while evolved enough to be usable, are renowned for their error-proneness.

Our work evolves from the initial proposal by G. Bes for a formal grammar distributed into five properties (the 5P formalism), which matured in turn into Property Grammars [2], in which syntactic structure is expressed by means of relations between categories rather than in terms of hierarchy. For instance, a Property Grammar parse of the noun phrase "every blue moon" results in a set of satisfied properties (e.g. *linear precedence* holds between the determiner and the noun, between the adjective and the noun, and between the determiner and the adjective; the noun's requirement for a determiner is satisfied, etc.) and a set of unsatisfied properties, which is empty for this example. In contrast, "Every moon blue" would yield a violation of linear precedence between the adjective and the noun, indicated by placing this relationship in the set of unsatisfied properties.

In this paper we provide a customisable extension of PGs that lets us also model concepts and relations to some degree, and we provide a parser for generic grammars and for PGs in particular, formulated in terms of HyProlog. HyProlog [8] is an extension of Prolog with assumptions and abduction, based on CHR (Constraint Handling Rules). Assumptions are implemented using linear and intuitionistic implication.

We illustrate our new PG model through its application to concept and relation extraction from biomedical texts.

2 Background

2.1 Property Grammars

Property based Grammars define any natural language in terms of a small number of properties: **linear precedence** (e.g. within a verb phrase, a transitive verb must precede the direct object); **dependency** (e.g., a determiner and a noun inside a noun phrase must agree in number), **constituency** (e.g. a verb phrase can contain a verb, a direct object,...), **requirement** (e.g. a singular noun in a noun phrase requires a determiner), **exclusion** (e.g., a superlative and an adjectival phrase cannot coexist in a noun phrase), **obligation** (e.g. a verb phrase must contain a verb), and **unicity** (e.g. a prepositional phrase contains only one preposition). The user defines a grammar through these properties instead of defining hierarchical rewrite rules as in Chomskyan based models. In addition, properties can be relaxed by the user in a simple modular way. For instance, we could declare "precedence" as relaxable, with the effect of allowing ill-formed sentences where precedence is not respected, while pointing out that they are ill-formed (this feature is useful for instance in language tutoring systems).

The result of a parse is, then, not a parse tree per se (although we do provide one, just for convenience, even in the case of ill-formed input), but a list of satisfied and a list of unsatisfied properties. See the table below as a toy example.

Input	`les cellules endothéliales immunotoxines peptides proapoptotiques (the endothelial ... cells)`
Output	`cat(np, [sing, masc], sn(det(les), n(cellules), ap(adj(endothéliales)), n(immunotoxines), n(peptides), n(proapoptotiques)), [prec(det,n), dep(det,n), requires(n, det), exclude(name, det), excludes(name, n), dep(sa, n), excludes(name, sa), excludes(sa, sup)], [unicity(n)])`

Property Grammars allow us to extract syntactic constructs from text, as first proposed in [6]. This functionality is useful to deal with part of the input for which no information can be built. It is for example the case with possibly long lists of juxtaposed *NP*, frequent in spoken languages but for which no specific syntactic relation can be given. But it is also interesting for some applications in which the entire syntactic structure or, in other words, the description of all possible syntactic categories is necessary. This is the case for question-answering, information extraction, or terminological applications based on *NP* recognition.

3 Semantic Property Grammars

Only one of the properties in the original PG formalism – dependency – is meant to include semantics, but there is no clear specification of how. Here we propose that, in addition to the mostly syntactic lists of satisfied and unsatisfied properties, a list of specifically Semantic Properties be associated as well with the category being parsed. This lets us model also concepts and relations to some degree.

We next introduce this extension by means of examples taken from biomedical text – one of the applications we are currently working on, which relies on a type hierarchy of concepts, or ontology, of the domain (in our case, biomedical) having been made available for consultation by the parser.

3.1 Extracting Semantic Concepts and Relationships

Within noun phrases: A noun is interpreted by our parser as the relational symbol of a semantic relationship to be extracted, and the arguments of this relation are constructed from the noun's various complements, appropriately typed after consultation of the domain-dependent ontology. E.g., the noun phrase: "The activation of NF-kappa-B via CD-28" parses semantically into the list of properties:

```
[protein('NF-kappa-B'), gene('CD-28'),
 activation('NF-kappa-B', 'CD-28')]
```

This shows the relationship obtained, i.e., `activation(‘NF-kappa-B’,‘CD-28’)`, together with the types that our concept hierarchy associates with each of the arguments of the relationship (i.e, `‘NF-kappa-B’` is of type *protein*, whereas `‘CD-28’` is of type *gene*.

Within verb phrases: Verbs also induce relationships whose arguments are the semantic representations of the verb's syntactic arguments. E.g., in the sentence: "Retinoblastoma proteins negatively regulate transcriptional activation", the verb *regulate* marks a relation between two concepts – *retinoblastoma proteins* and *transcriptional activation*. Our parser constructs a list of properties which identifies the semantic types of the relationship's arguments as well as the relationship itself, just as was done for noun phrases.

```
[protein(‘retinoblastoma proteins’),
 process(‘transcriptional activation’),
 regulate(‘retinoblastoma proteins’,
 ‘transcriptional activation’)]
```

Disambiguation on the fly: Our parser's consultation of a biomedical ontology is useful not only to gather type information about the entities involved in the relationships being constructed, but also to disambiguate in function of context. For instance, usually *binding site* refers to a *DNA* domain or region, while sometimes it refers to a *protein* domain or region. Catching the latter meaning is not trivial since both *c-Myc* and *G28-5* are protein molecules. However, our parser looks for semantic clues from surrounding words in order to disambiguate: in sentence (1) below, *promoters* points to the DNA region binding site, whereas in sentence (2), *ligands* points to the protein meaning of binding site. Our parser can calculate an entity's appropriate type by consulting domain-specific clues. More details can be found in [9].

(1) Transcription factors USF1 and USF2 up-regulate gene expression via interaction with an E box on their target **promoters**, which is also a **binding site** for **c-Myc**.

(2) The functional activity of **ligands** built from the **binding site** of **G28-5** is dependent on the size and physical properties of the molecule both in solution and on the cell surfaces.

4 Our Parsing Methodology

4.1 Background: HyProlog

Hyprolog is an extension of Prolog with assumptions and abduction, useful for hypothetical reasoning and other applications, running on top of Sicstus Prolog, and particularly exploiting Constraint Handling Rules or CHR [10] . Its three main components are:

Assumptions: Assumptions are a logic programming incarnation of linear and intuitionistic implications [11] which includes as well a new type called *timeless* assumptions, which have been found particularly useful, among other things, for parsing. Here we will only be concerned with linear and intuitionistic assumptions, which we now present in an intuitive manner. For fuller or more formal details, see [8].

Assumptions are facts (or rules, but for our needs here we consider only facts) that are added to a Prolog program dynamically, making them available from the point in which they are called, and during the entire continuation. Linear assumptions can be used at most once, whereas intuitionistic assumptions can be used as many times as needed. Linear assumptions being made are noted through the "+" sign and intuitionistic ones, through "*". To use, or consume, either type of assumption, we just precede it by the sign "-".

For instance, the call:

```
?- *note(p(5)), -note(p(X)), -note(p(Y)).
```

will bind both X and Y with 5, since intuitionistic assumptions are reusable, whereas the call

```
?- +note(p(5)), -note(p(X)), -note(p(Y)).
```

will bind X to 5 and then fail.

Assumptions and consumptions are similar to the Prolog primitives "assert" and "retract", except that they are available during the entire continuation of the computation, and that they are backtracked upon.

Constraint Handling Rules, or CHR:

A CHR program is a finite set of rules of the form `{Head ==> Guard|Body}` where `Head` and `Body` are conjunctions of atoms and `Guard` is a test constructed from built-in predicates; the variables in `Guard` and `Body` occur also in `Head`; in case the `Guard` is the local constant "true", it is omitted together with the vertical bar. Its logical meaning is the formula $\forall(\texttt{Guard} \rightarrow (\texttt{Head} \rightarrow \texttt{Body}))$ and the meaning of a program is given by conjunction.

A *derivation* starting from an initial state called a *query* of ground constraints is defined by applying rules as long as it adds new constraints to the store. A rule as above *applies* if it has an instance (`H==>G|B`) with `G` satisfied and `H` in current store, and it does so by adding `B` to the store.

There are three types of CHR rules:

- **Propagation rules** which add new constraints (body) to the constraint set while maintaining the constraints inside the constraint store for the reason of further simplification.
- **Simplification rules** which also add as new constraints those in the body, but remove as well the ones in the head of the rule.
- **Simpagation rules** which combine propagation and simplification traits, and allow us to select which of the constraints mentioned in the head of the rule should remain and which should be removed from the constraint set.

Abduction: Abduction is the unsound but useful rule of inference which concludes – or abduces – a from the knowledge of b and of the rule that a implies b.

Abduction can be simply incorporated through CHR by declaring certain predicates as abducibles, which when generated and not resolvable, will simply remain in the constraint store. E.g. if every time it rains I go to the cinema, and going to the cinema has been declared as abducible, when querying: cinema, there being no definitions for it, it will remain in the constraint store, marked as abduced thanks to the declaration which states it is an abducible predicate.

4.2 A Hyprolog Parser for Property Grammars

A Hyprolog program is written as a Prolog program with additional declarations of assumptive and abductive predicates. In this paper we neglect to write all declarations, since it is apparent from our discussion which predicates are assumptions and which are abducibles.

We record all categories (including lexical categories) as assumptions, and rule application repeatedly combines two categories (one of which is a phrase or a phrase head) between which all properties hold into new category, also recorded as an assumption. The process stops when no more categories can be inferred. Syntactic categories are described in the notation:

```
+cat(Cat,Features,Graph,Sat,Unsat,Semantics,Start,End)
```

where `Cat` names a syntactic category stretching between the start and end points `Start` and `End`, respectively; `Features` contains the list of its syntactic features, such as gender and number, `Graph` represents the portion of parse tree corresponding to this category and is automatically constructed by the parser; `Sat` and `Unsat` are respectively the list of syntactic properties respectively satisfied and unsatisfied by this category; and `Semantics` holds the Semantic Properties list.

The sentence's words and related information, thus expressed as assumptions, can be viewed as resources to be consumed in the process of identifying higher level constituents from them.

Lists of satisfied and unsatisfied properties are created by our single rule, so that incorrect or incomplete input is admitted but the anomalies are pointed out from the list of unsatisfied properties. Our single rule's form is described in the Appendix found at www.sfu.ca/`~emaharsh/`.

Note that the parser's single rule consumes two resources before creating a new one, so each rule application decreases the (initially finite) number of resources available by one. The process stops when no more rule applications are possible, leaving if successful a category "sentence" stretching between the start and end points of the input sentence, and containing its full characterization (satisfied and unsatisfied properties, semantics, etc.).

5 Related Work, Concluding Remarks

Several linguistic theories make an intensive use of the constraint based approach to parsing, in particular HPSG (see [16]), minimalism [3] or the Optimality Theory (see [17]). As in the last two, we aim at very few general operations: We work with just one rule at the heart of parsing, and unlike minimalism, need not to filter candidate structures.

One other formalism that shares the aims and some of the features of Property Grammars are Dependency Grammars (cf. [20] and on this point [13]), a purely equational system in which the notion of generation, or derivation between an abstract structure and a given string, is also absent. However, whereas in Dependency Grammars, the property of dependence plays a fundamental role, in the framework we are considering it is but one of the many properties contributing to a category's characterization. Perhaps the work that most relates to the syntactic implementation parts of our work is Morawietz's [14], which implements deductive parsing [19] in CHR, and proposes different types of parsing strategies (including one for Property Grammars) as specializations of a general bottom-up parser. Efficiency however is not addressed beyond a general discussion of possible improvements, so while theoretically interesting, this methodology is in practice unusable due to combinatorial explosion. Moreover, it produces all properties that apply for each pair of categories without keeping track of how these categories are formed in terms of their subcategories, so there is no easy way to make sense of the output in terms of a complete analysis of a given input string.

Our parser, as we saw, keeps track of the output in the lists of properties (syntactic and semantic) that it constructs, and it reduces the problem of combinatorial explosion since every rule application condenses two assumptions into one. However further work is needed for a rigorous analysis of efficiency and complexity. Comparison with other systems and benchmarks regarding efficiency is also part of the future work.

References

1. Abney, S.: Parsing By Chunks. In: Berwick, R., Abney, S., Tenny, C. (eds.) Principle-Based Parsing. Kluwer Academic Publishers, Dordrecht (1991)
2. Blache, P.: Property Grammars: A Fully Constraint-Based Theory", in H. Christiansen et al. (eds), Constraint Solving and NLP, LNCS, Springer, 2005.
3. Chomsky, N.: The Minimalist Program. MIT Press, Cambridge (1995)
4. Christiansen, H., Dahl, V.: HYPROLOG: A New Logic Programming Language with Assumptions and Abduction. In: Gabbrielli, M., Gupta, G. (eds.) ICLP 2005. LNCS, vol. 3668, pp. 159–173. Springer, Heidelberg (2005)
5. Dahl, V., Blache, P.: Directly Executable Constraint Based Grammars. In: Proc. of Journees Francophones de Programmation en Logique avec Contraintes (2004)
6. Dahl, V., Blache, P.: Extracting Selected Phrases through Constraint Satisfaction. In: Proceeding of ICLP workshop on CSLP (2005)
7. Dahl, V., Tarau, P.: Assumptive Logic Programming. In: Proc. ASAI 2004, Cordoba, Argentina (2004)

8. Dahl, V., Tarau, P., Li, R.: Assumption Grammars for Natural Language Processing. In: Naish, L. (ed.) Proc. Fourteenth International Conference on Logic Programming, pp. 256–270. MIT Press, Cambridge (1997)
9. Dahl, V., Gu, B.: A CHRG Analysis of ambiguity in Biological Texts. In: Proceedings of the 4th International Workshop on Constraints and Language Processing (CSLP), Denmark (2007)
10. Frühwirth, T.: Theory and Practice of Constraint Handling Rules. Journal of Logic Programming 37, 1–3 (1998)
11. Girard, J.-Y.: Linear Logic. Theoretical Computer Science 50, 1–102 (1987)
12. Kakas, A.C., Kowalski, R.A., Toni, F.: The role of abduction in logic programming. In: Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 5, pp. 235–324 (1998)
13. Mel'čuk, I.: Dependency Syntax. SUNY Press (1988)
14. Morawietz, F.: Chart parsing as contraint propagation. In: Proceedings of COLING 2000 (2000)
15. Osborne, M.: MDL-based DCG Induction for NP Identification. In: Proceedings of CoNLL 1999 (1999)
16. Pollard, C., Sag, I.: Head-driven Phrase Structure Grammars. CSLI, Chicago University Press (1994)
17. Prince, A., Smolensky, P.: Optimality Theory: Constraint Interaction in Generative Grammars, Technical Report RUCCS TR-2, Rutgers Center for Cognitive Science (1993)
18. Sag, I., Wasow, T.: Syntactic Theory. A Formal Introduction. CSLI (1999)
19. Shieber, S., Schabes, Y., Pereira, F.: Principles and implementation of deductive parsing. Journal of Logic Programming 24(1-2), 3–36 (1995)
20. Tesnière, L.: Éléments de syntaxe structurale, Klincksieck (1959)

Adaptable Grammars for Non-Context-Free Languages

Henning Christiansen

Research group PLIS: Programming, Logic and Intelligent Systems
Department of Communication, Business and Information Technologies
Roskilde University, P.O.Box 260, DK-4000 Roskilde, Denmark
`henning@ruc.dk`
`http://www.ruc.dk/~henning/`

Abstract. We consider, as an alternative to traditional approaches for describing non-context-free languages, the use of grammars in which application of grammar rules themselves control the creation or modification of grammar rules. This principle is shown to capture, in a concise way, standard example languages that are considered as prototype representatives of non-context-free phenomena in natural languages. We define a grammar formalism with these characteristics and show how it can be implemented in logic programming in a surprisingly straightforward way, compared with the expressive power. It is also shown how such adaptable grammars can be applied for describing meta-level architectures that include their own explicit meta-languages for defining new syntax.

1 Introduction

An important aspect of formal linguistics is to search for suitable grammar formalisms that can describe non-context-free languages; both the formal expressibility as well as *how* these languages are described, are important (are the grammars "natural" in some sense, or which conception of language does a class of grammars represent).

Many classical approaches to handle extra-context-free aspects apply specialized derivation relations so that everything is captured in terms of rewriting; two-level grammars [1] and contextual grammars [2] are examples of this. Other directions use extra-grammatical add-on's to context-free grammars such as first order logic in Definite Clause Grammars [3] and arbitrary mathematical functions in Attribute Grammars [4].

We consider here a third approach by turning the grammar itself into a dynamic entity, so that the application of grammar rules can create new or modify existing grammar rules. The scope of grammar rules can be controlled analogously to the way Definite Clause Grammars and Attribute Grammars pass information around to different parts of the string being analyzed or constructed. The difference is that here we pass the entire and dynamically evolving grammar around.

We present a formalism that we call Adaptable Grammars, based on an earlier proposal of ours [5,6,7], which has not been considered in a formal linguistics setting before, but confined to applications for programming languages.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 488–495, 2009.
© Springer-Verlag Berlin Heidelberg 2009

These grammars were first introduced with the misleading name of 'generative grammars', later changed into 'adaptable grammars'; other authors have used the name 'Christiansen grammars', e.g., [8,9,10,11]. Recently [10] have used these grammars for grammatical, evolutionary programming; the authors motivate their approach by the observation that with such grammars, they can do with shorter derivations of target programs, thus shorter chromosomes and therefore faster convergence.

The first trace of the principle of grammars that adapt to the context seems to be [12] from the early 1960ies, but no general formalism was developed; an extensible grammar formalism was developed in [13] for describing one particular system, and other proposals are [8,14]; see also [6,7] for a detailed comparison with other selected formalism.

In the present paper, we define an adaptable grammar formalism which borrows formal concepts from first-order logic and logic programing. We show a generic implementation written as a meta-interpreter in Prolog, which is strikingly simple compared with the high expressibility of the class of adaptable grammars that it implements. We show how these grammars can capture standard non-context-free languages used in the literature (e.g., [15]) as prototypical representatives for the central natural language properties of *reduplication*, *crossed dependencies*, and *multiple agreements.* Finally we show how such grammars can be used to describe meta-level architectures that include their own explicit meta-languages for defining new syntax.

2 Definition

We assume the terminology of first-order logic, including logical variables, and logic programs (pure Prolog programs) as well as related notions such as (ground) instances for formulas and terms.

Definition 1. *An* adaptable grammar *is a quintuple* $\langle \Sigma, N, \Pi, [\![-]\!], R\rangle$ *where* Σ *is a finite* alphabet*,* N *a set of* nonterminal *symbols which are logical predicate symbols of arity at least 1,* Π *is a logic program,* $[\![-]\!]$ *is the* denotation function *which is a (partial) function from terms to grammars, and* R *is a set of grammar rules (below). Nonterminals and atoms defined as predicates in* Π *are assumed to be disjoint. Each nonterminal symbol has a distinguished argument called its* grammar argument *(or* grammar attribute*). A* grammar rule *is of the form*

$$lhs \texttt{ --> } rhs.$$

where lhs is a nonterminal and rhs a finite sequence of elements which may be terminal or nonterminal symbols or first-order atoms defined by Π*.*

We apply a notation inspired by definite clause grammars so that terminals in grammar rules are written in square brackets, and atoms referring the program component in curly brackets. Furthermore, the grammar argument of a nonterminal symbol is assumed to be the last of its arguments, and we drag it outside the standard parentheses and attach it by a hyphen, e.g., instead of `n(x,y,G)`, we prefer `n(x,y)-G` when `n` is a nonterminal of arity 3. Terms within rules that represent grammars are generally written using Prolog's list syntax but may be written in other ways depending on the actual denotation function. In general, we expect that the denotation function only changes the rule sets, so that the other components are fixed within the derivations made with a given initial grammar.

As it follows from the results of [16], the inclusion of logic programs in the formalism does not add anything new as they can be seen as grammars that generate the empty string. They are included for notational convenience only.

Definition 2 (derivation). *Given an adaptable grammar* $\langle \Sigma, N, \Pi, [\![-]\!], R\rangle$*, an* application instance *of one of its rules* $r \in R$ *is a ground instance* r' *of* r *in which grammar arguments denote grammars and any logical atom in* r' *is satisfied in* Π*. Whenever* $\alpha\beta\gamma$ *is a sequence of ground grammar symbols,* β *being a nonterminal of the form* N-G *with an application instance* β`-->`δ *of a rule in* $[\![G]\!]$*, we write*

$$\alpha\beta\gamma \Rightarrow \alpha\delta'\gamma$$

where δ' *is a copy of* δ *with any atom referring to* Π *taken out. The relation* $\Rightarrow^*$ *refers to the reflexive, transitive closure of* $\Rightarrow$*.*

The language *defined by a given ground nonterminal* N-G *is the set of terminal strings* σ *for which* N-$G \Rightarrow^* \sigma$*.*

Notice the difference between grammars and terms that represent them. A grammar attribute may be a list of rule representations containing duplicates, and the "processing" of it may take multiplicity into account.

3 Plain Vanilla Implementation of Adaptable Grammars

Adaptable grammars as we have defined them above can be implemented in Prolog in a straightforward way, based on well-tested techniques. We will assume a non-ground representation of grammars, i.e., using logical variables to denote logical variables. This representation has the advantage that programs can be written very succinctly but requires a bit of care to avoid mixing up quantification levels. We take this for granted and refer to standard literature on this topic, e.g., [17], and avoid these potential technical problems in our examples.

Instead of magically predicting the right application instances of grammar rules in order to analyze a given string, we use unification when a rule is applied, and embedded code chunks are executed in the order they are encountered in the current derivation; these are also standard techniques for execution of Prolog programs and language recognition with Definite Clause Grammars [3], so we take correctness for granted. Our implementation is similar to a meta-interpreter implementation of Definite Clause Grammars with the single extension that grammar rules are picked in the grammar argument instead of in Prolog's global database. An extra predicate for renaming variables in grammar rules is needed, implemented here in a traditional one-line fashion. The following few code lines are sufficient to check whether a given string S can be derived from an adaptable grammar nonterminal NG by the call `derive(`NG`, `S`)`.

```
derive(NG, S):- derive(NG, S, []).

derive([], S, S).

derive([T|Ts], [T|S1], S2):- derive(Ts, S1, S2).

derive({Code}, S, S):- Code.

derive((A,B), S1,S3):- derive(A, S1, S2), derive(B, S2, S3).

derive(N-G, S1, S2):-
    renameVars(G,GFresh),
    member( (N-G --> Rhs), GFresh),
    derive(Rhs,S1,S2).

renameVars(X,Y):- assert(quax(X)), retract(quax(Y)).
```

It is illustrated below how new grammars rules are created and passed around.

4 Important Non-Context-Free Languages in Adaptable Grammars

In formal linguistics, three different phenomena are often emphasized as central for natural language; they are called *reduplication*, *crossed dependencies* and *multiple agreements.* These phenomena are exposed in the following languages (see, e.g., [15]):

$$L_1 = \{rcr \mid r \in \{a,b\}^*\}$$
$$L_2 = \{a^n b^m c^n d^m \mid n,m \geq 1\}$$
$$L_3 = \{a^n b^n c^n \mid n \geq 1\}$$

These are straightforward to describe in adaptable grammars as we will show below. We use the language

$$RR = \{rr \mid r \text{ is a string of } \mathtt{a}\text{s and } \mathtt{b}\text{s}\}.$$

as a prototypical example and use it also to illustrate the Prolog representation in detail. We can characterize RR by a grammar that generates the first half part of the string in a standard way and in parallel maintains a rule for generating the second half.

Derivation is started from a nonterminal `r-`G where G is bound to a representation of the grammar which has three rules, one rule for terminating the string by adding a second half, one for generating an `a` and which adds `a` to the termination rule, and a similar one for `b`. This adaptable grammar can be conveniently represented in a Prolog source text by a Prolog rule as follows.

```
rrgram(  [ ( r-G --> [] ),
           ( r-G --> [a],{C1}, r-G1 ),
           ( r-G --> [b],{C2}, r-G1 )    ]):-

 C1 = (G =[(S-->List)|RestG],
       append(List,[a],NewList),
       G1=[(S-->NewList)|RestG]),
 C2 = (G= [(S-->List)|RestG],
       append(List,[b],NewList),
       G1=[(S-->NewList)|RestG]).
```

It could have been written equivalently as a Prolog fact, but we use the substitutions in the rule body in order to obtain a more readable representation; variables `C1` and `C2` abbreviate the embedded code that transforms the grammar attribute. It should be noticed that neither this grammar nor its application creates circular Prolog structures. The denotations function which is implicit in the `rename` predicate ensures this. Testing whether a given string belongs to *RR* can now be made by the following query.

```
?- rrgram(G), derive(r-G, [a,b,b,a,a,b,b,a]).
yes
```

It appears that the rule `r-G --> [a,b,b,a]` is generated and applied as the last one in the derivation. We have chosen the language *RR* instead of L_1, as it may be more difficult to model in some formalisms due to the lack of a “middle demarcation” *c*. Changing our grammar above to L_1 is a matter of changing the first rule of the initial grammar into `r-G --> [c]`.

The principle applied in the grammar for *RR* provides immediately the clue for describing L_2 and L_3. For L_2 we can use an initial grammar consisting of the following rules (sketched).

```
S --> [a], S1
S1-G --> [a], {code₁}, S1-G1
S1-G --> [b], S2-G
S2-G --> [b], {code₂}, S2-G1
S2-G --> Cs-G, Ds-G
Cs-G --> [c]
Ds-G --> [d]
```

The embedded code fragment $code_1$ adds a `c` to the right-hand side of the rule for `Cs`, and $code_2$ similarly a `d` to the rule for `Ds` in order to form the relevant grammar to be unified with the `G1` variables.

The language L_3 can be described in exactly the same way, so that each time the rule for generating an `a` is applied, rules for `b`s and `c`s are extended with one more letter.

5 Adaptable Grammars for Grammar-Definitions and other Meta-languages

The language in a discourse may develop as the discourse goes along, new usages are introduced or usages may achieve new meanings, which can be "stored" in grammar rules. We may say that the semantic-pragmatic context (which is something very different from the syntactic context considered in Contextual Grammars [2]) is growing along the discourse and in this way enriches the linguistic potential.

We consider here meta-level architectures, as they arise in parser generators (e.g., YACC [18]), as archetypical in that they provide explicit language constructs that represent linguistic enrichment.

Indeed, one of the most striking features of adaptable grammars is their ability to characterize meta-level architectures. A formal grammar, for example, may itself by written in an idiosyncratic language of the sort typically called a meta-language. Context-free grammars may be written with the left and right-hand sides separated by, e.g, "`::=`" or "`-->`".

We describe here the syntax of a meta-level system that accepts the definition of a context-free grammar *G* in a specific syntax together with an arbitrary sample string, supposed to be described by *G*. The following is an example of a correct formulation in this meta-level system; for simplicity, nonterminals must be one of "`start`", "`n1`", "`n2`", "`n3`", and "`empty`", and terminals must be chosen among "`a`" and "`b`"; "`start`" is assumed always to be the start nonterminal.

```
start ::= n1 n2 .
n1 ::= a b .
n2 ::= b a .
sample a b b a
```

The meta-level grammar for this system can be written as the following adaptable grammar; notice that "`G`" typically stands for a variable in the position of a grammar attribute, whereas "`Gvar`" typically *denotes* such a variable.

```
s-G --> gram(C)-G, [sample], start-C
gram([(empty-Gvar--> [])])-G --> []
gram([R|Rs])-G --> rule(R)-G, gram(Rs)-G
rule((H-->B))-G --> head(Gvar,H)-G,[::=],body(Gvar,B)-G
head(Gvar,H)-G --> nonterm(Gvar,H)-G
body(Gvar, ([T],Bs))-G --> term(T)-G, body(Gvar,Bs)-G
body(Gvar,(N,Bs))-G --> nonterm(Gvar,N)-G, body(Gvar,Bs)-G
body(Gvar,empty-Gvar)-G --> [.]
term(a)-G --> [a]
term(b)-G --> [b]
term(c)-G --> [d]
term(e)-G --> [e]
term(f)-G --> [f]
nonterm(Gvar,start-Gvar)-G --> [start]
nonterm(Gvar,n1-Gvar)-G --> [n1]
nonterm(Gvar,n2-Gvar)-G --> [n2]
nonterm(Gvar,n3-Gvar)-G --> [n3]
```

When the first rule is applied for `s`-G, where G is the term representing this grammar, firstly `gram(C)`-G is processed, producing a new adaptable grammar C which is bound to the variable `C`; then `start`-C is started for the investigation of a sample sentence.

Analyzing the text fragment above with our Prolog implementation, a test print can show that the following adaptable grammar is generated internally and forwarded to `start-C`.

```
start-Gvar --> n1-Gvar, n2-Gvar, empty-Gvar
n1-Gvar --> [a], [b], empty-Gvar
n2-Gvar --> [b], [a], empty-Gvar
empty-Gvar --> []
```

Programming languages include also devices that may be seen as language extenders. Declaring, say, a variable `n` of type `int` means that we can now write `n` in positions of the program where an `int` is expected; before the declaration was added, such usages were illegal, ungrammatical so to speak. In other words, the indicated variable declaration can be seen as the creation a new grammar rule `int-G --> [n]`. Programming languages were in focus in earlier publications on adaptable grammars; see [5,6,7] for more examples of programming language constructs described in this way.

6 Conclusion

We have defined a highly expressive grammar formalism which captures non-context-free language features by treating the grammar itself as a dynamic entity, which can be elaborated and changed by the application of grammar rules. We have shown a vanilla-like meta-interpreter which makes it possible to test and experiment with adaptable grammars. We have justified the usefulness of this grammar formalism by showing grammars for important examples of non-context-free languages.

A more powerful and logically more satisfactory implementation may be possible using a constraint-based denotation function, which may be implemented using 'instance of' constraints [19] with a potential for producing new rules in abductive or even inductive ways from rule applications. Constraint-based abduction in discourse analysis, as applied e.g., by [20,21], may also be interesting to combine with the approach described in the present paper.

References

1. van Wijngaarden, A., Mailloux, B.J., Peck, J.E.L., Koster, C.H.A., Sintzoff, M., Lindsey, C.H., Meertens, L.G.L.T., Fisker, R.G.: Revised report on the algorithmic language algol 68. Acta Informatica 5, 1–236 (1975)

2. Marcus, S.: Contextual grammars. Rev. roum. de math. pures et appl. 14, 1473–1482 (1969)
3. Pereira, F.C.N., Warren, D.H.D.: Definite clause grammars for language analysis—A survey of the formalism and a comparison with augmented transition networks. Artificial Intelligence 13(3), 231–278 (1980)
4. Knuth, D.E.: Semantics of context-free languages. Mathematical Systems Theory 2(2), 127–145 (1968)
5. Christiansen, H.: Syntax, semantics, and implementation strategies for programming languages with powerful abstraction mechanisms. In: 18th Hawaii International Conference on System Sciences, vol. 2, pp. 57–66 (1985)
6. Christiansen, H.: The syntax and semantics of extensible languages. Datalogiske skrifter 14 (Tech. rep). Computer Science Section, Roskilde University, Denmark (1988)
7. Christiansen, H.: A survey of adaptable grammars. SIGPLAN Notices 25(11), 35–44 (1990)
8. Shut, J.N.: Recursive adaptable grammars. Master's thesis, Computer Science Department, Worcester Polytechnic Institute, Worcester Massachusetts, USA (1993)
9. Jackson, Q.T.: Adapting to Babel: Adaptivity and Context-Sensitivity in Parsing. Ibis Publications, Plymouth (2006)
10. Ortega, A., de la Cruz, M., Alfonseca, M.: Christiansen grammar evolution: Grammatical evolution with semantics. IEEE Trans. Evolutionary Computation 11(1), 77–90 (2007)
11. Weise, T.: Global optimization algorithms - theory and application. Electronic manuscript (e-book); linked checked January 2008 (2007)
12. di Forino, A.C.: Some remarks on the syntax of symbolic programming languages. Communications of the ACM 6(8), 456–460 (1963)
13. Wegbreit, B.: Extensible programming languages. Harward University, Cambridge (2006); Garland Publishing Inc., New York & London (1980)
14. Burshteyn, B.: On the modification of the formal grammar at parse time. SIGPLAN Notices 25(5), 117–123 (1990)
15. Marcus, S., Paun, G., Martín-Vide, C.: Contextual grammars as generative models of natural languages. Computational Linguistics 24(2), 245–274 (1998)
16. Deransart, P., Maluszynski, J.: Relating logic programs and attribute grammars. Journal of Logic Programming 2(2), 119–155 (1985)
17. Hill, P.M., Gallagher, J.: Meta-programming in logic programming. In: Handbook of Logic in Artificial Intelligence and Logic Programming, pp. 421–497. Oxford Science Publications/ Oxford University Press, Oxford (1994)
18. Johnson, S.C.: Yacc: Yet another compiler-compiler. Technical Report CSTR 32, Bell Laboratories, Murray Hill, N.J., USA (1975)
19. Christiansen, H.: Automated reasoning with a constraint-based metainterpreter. Journal of Logic Programming 37(1-3), 213–254 (1998)
20. Christiansen, H., Dahl, V.: Meaning in Context. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) CONTEXT 2005. LNCS, vol. 3554, pp. 97–111. Springer, Heidelberg (2005)
21. Christiansen, H., Dahl, V.: HYPROLOG: A new logic programming language with assumptions and abduction. In: Gabbrielli, M., Gupta, G. (eds.) ICLP 2005. LNCS, vol. 3668, pp. 159–173. Springer, Heidelberg (2005)

β-Reduction and Antecedent–Anaphora Relations in the Language of Acyclic Recursion[⋆]

Roussanka Loukanova

Computational Linguistics, Uppsala University

Abstract. In this paper, I introduce briefly the formal language of Acyclic Recursion L_{ar}^{λ} (see Moschovakis [9]). Then I present the β-reduction rule of the calculus of the referential synonymy in L_{ar}^{λ} and some arguments for its restricted manifestation with respect to preserving the algorithmic sense of the L_{ar}^{λ}-terms. This paper demonstrates further evidence for the need of restricted β-reduction. It also contributes to exploration of the inherent potentials of L_{ar}^{λ} for underspecified semantic representations. I show that the restricted β-reduction, in combination with "underspecified" terms, gives rise to various distinctions between antecedent-anaphora and co-denotation relations in NL.

1 Background and Recent Development

In a sequence of papers, Moschovakis developed a class of languages of recursion, as a new approach to the mathematical notion of algorithm, for development of computational semantics of languages, e.g., see Moschovakis [9] for FLR, and Moschovakis [10] for L_{ar}^{λ}. In particular, the language and theory of acyclic recursion L_{ar}^{λ} is intended for modeling the logical concepts of meaning and synonymy, from the perspective of the theory of computability, by targeting adequateness of computational semantics of NL. L_{ar}^{λ} is a higher order type theory, which is a proper extension of Gallin's TY_2 (Gallin [3]) and, thus, of Montague's Intensional Logic, IL (Montague [8]). The formal syntax of L_{ar}^{λ} allows only recursion terms with acyclic systems of assignments, while the FLR allows cyclicity, but is limited with respect to its type system. The languages of recursion have two semantic layers: denotational semantics and referential intensions. The recursion terms of L_{ar}^{λ} are essential for encoding the two-fold semantic information. **Denotational Semantics:** For any given semantic structure $\mathfrak{A}$, there is at most one, well-defined denotation function, den, from terms and variable assignments to objects in the typed domains of $\mathfrak{A}$. **Intensional Semantics:** The notion of intension in the languages of recursion covers the most essential, computational aspect of the concept of meaning. Intuitively, $\mathsf{Int}(A)$ is the *algorithm* for computing its denotation $\mathsf{den}(A)$. The acyclicity condition guarantees termination of the recursive steps that compute $\mathsf{den}(A)$. In general, an algorithmic process may not terminate. Thus, the languages of recursion offer a formalisation of central

---

[⋆] My work on this paper was conducted during my visit at TiLPS, Tilburg University, in the Autumn, 2008. I would like to thank my colleagues there for the fruitful time.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 496–503, 2009.
© Springer-Verlag Berlin Heidelberg 2009

computational aspects of Frege's distinction between denotation and sense, with two central semantic "levels": *denotation* and *algorithmic sense*. The algorithmic sense of a term A is its referential intension that is the algorithm for computing its denotation. The denotation function $\mathsf{den}(A)$ defined in L^{λ}_{ar} captures Montague's notion of intension, i.e., the Carnap intensions that are functions from indexes (including possible worlds) to extensions in the indexes. The notion of referential intension is different from the notion of Carnap intension, and has no counterpart in higher order type theories, like TY_2.

2 Short Introduction to the Language and Theory of Acyclic Recursion

Types The types of L^{λ}_{ar} is the smallest set *Types* defined recursively (by using a notational variant of Backus-Naur Form (BNF) for CFGs):

$$\sigma :\equiv e \mid t \mid s \mid (\tau_1 \rightarrow \tau_2) \qquad (\mathit{Types})$$

Intuitively, the types are used in a typical fashion: e is for the entities of the domain; t is for truth values; s is for states (e.g., a world, time, space location, speaker, etc.) $(\tau_1 \rightarrow \tau_2)$ is for unary functions from objects of type τ_1 to objects of type τ_2. The following are some (very) practical abbreviations used throughout the paper:

1. $\widetilde{\mathsf{t}} \equiv (\mathsf{s} \rightarrow \mathsf{t})$, the type of the propositions, as in some type systems like TY_2, Montague's IL, etc.
2. $\widetilde{\mathsf{e}} \equiv (\mathsf{s} \rightarrow \mathsf{e})$, the type of the individual concepts
3. $\widetilde{\tau} \equiv (\mathsf{s} \rightarrow \tau)$, where $\tau \in \mathit{Types}$
4. $\tau_1 \times \tau_2 \rightarrow \tau \equiv (\tau_1 \rightarrow (\tau_2 \rightarrow \tau))$, where $\tau_1, \tau_2, \tau \in \mathit{Types}$

2.1 Syntax of L^{λ}_{ar}

The vocabulary of L^{λ}_{ar} consists of pairwise disjoint sets:

Constants: $K_\tau = \{\mathsf{c}_0, \mathsf{c}_1, \ldots, \mathsf{c}_{k_\tau}\}$, for each type τ, a finite set of constants of type τ;

Pure Variables: $\mathit{PureVars}_\tau = \{\mathsf{v}_0, \mathsf{v}_1, \ldots\}$, for each type τ, a set of pure variables of type τ;

Recursion Variables: $\mathit{RecVars}_\tau = \{\mathsf{p}_0, \mathsf{p}_1, \ldots\}$, for each type τ, a set of recursion variables, called also locations, of type τ.

The Terms of L^{λ}_{ar} The recursive rules[1] for the set of L^{λ}_{ar} terms can be expressed by using a notational variant of BNF, with the assumed types given as superscripts:

$$\begin{aligned} A :\equiv\ & \mathsf{c}^{\tau} : \tau \mid x^{\tau} : \tau \mid B^{(\sigma \rightarrow \tau)}(C^{\sigma}) : \tau \mid \lambda v^{\sigma}\,(B^{\tau}) : (\sigma \rightarrow \tau) \\ & \mid A_0^{\sigma}\ \mathsf{where}\ \{p_1^{\sigma_1} := A_1^{\sigma_1}, \ldots, p_n^{\sigma_n} := A_n^{\sigma_n}\} : \sigma \end{aligned}$$

---

[1] A detailed definition will be provided in the final version of the paper, with a proper inclusion of the acyclicity condition "inside" the recursion, as the above notational variant of BNF should be taken.

where $\{p_1 := A_1, \dots, p_n := A_n\}$ is a set of assignments that satisfies the acyclicity condition[2] defined as follows:

Acyclic System of Assignments: For any terms $A_1 : \sigma_1$, …, $A_n : \sigma_n$, and locations $p_1 : \sigma_1, \dots, p_n : \sigma_n$ (where $n \geq 0$), such that for all i, j, if $i \neq j, 1 \leq i, j \leq n$, then $p_i \neq p_j$, the system of assignments

$$\{p_1 := A_1, \dots, p_n := A_n\}$$

is called *acyclic* iff the following condition, called *acyclicity condition*, is satisfied: there is a function $\mathit{rank} : \{p_1, \dots, p_n\} \longrightarrow \mathbb{N}$ such that, for all $p_i, p_j \in \{p_1, \dots, p_n\}$, if p_j occurs free in A_i, then $\mathit{rank}(p_j) < \mathit{rank}(p_i)$.

Intuitively, an acyclic system $\{p_1 := A_1, \dots, p_n := A_n\}$ defines recursive "computations" of what is to be assigned to the locations $p_1, \dots, p_n$, which end after a finite number of steps; ranking means that p_i may depend on the free locations occurring in A_i.

2.2 Denotational Semantics of L^{λ}_{ar}

The semantics of L^{λ}_{ar} defines denotations of the terms with respect to a (typical) typed frame of semantic objects and an interpretation of the constants in the given typed frame. The denotations of the terms $\mathsf{den}(A)(g)$ in L^{λ}_{ar}-structures are defined in a typical way, except for the recursion *where*-terms.

For any frame $\mathbb{T} = \{\mathbb{T}_\sigma \mid \sigma \in \mathit{Types}\}$ and the set G of all assignments $g \colon \mathit{PureVars} \cup \mathit{RecVars} \longrightarrow \mathbb{T}$, the *denotation function* is defined by recursion on the structure of the terms:

(D1) $\mathsf{den}(x)(g) = g(x)$; $\mathsf{den}(c)(g) = \mathcal{I}(c)$;
(D2) $\mathsf{den}(A(B))(g) = \mathsf{den}(A)(g)(\mathsf{den}(B)(g))$;
(D3) $\mathsf{den}(\lambda x(B))(g) : \mathbb{T}_\tau \longrightarrow \mathbb{T}_\sigma$, where $x : \tau$ and $B : \sigma$, is the function defined as follows: for every $t \in \mathbb{T}_\tau$, $[\mathsf{den}(\lambda x(B))(g)]\big(t\big) = \mathsf{den}(B)(g\{x := t\})$;
(D4) $\mathsf{den}(A_0 \text{ where } \{p_1 := A_1, \dots, p_n := A_n\})(g) =$
$\mathsf{den}(A_0)(g\{p_1 := \overline{p}_1, \dots, p_n := \overline{p}_n\})$,
where, for all $i \in \{1, \dots, n\}$ and all τ_i such that $p_i : \tau_i$, the values $\overline{p}_i \in \mathbb{T}_{\tau_i}$, are defined by recursion on $\mathsf{rank}(p_i)$:

$$\overline{p_i} = \mathsf{den}(A_i)(g\{p_{k_1} := \overline{p}_{k_1}, \dots, p_{k_m} := \overline{p}_{k_m}\}),$$

where $p_{k_1}, \dots, p_{k_m}$ are all the recursion variables $p_j \in \{p_1, \dots, p_n\}$, such that $\mathsf{rank}(p_j) < \mathsf{rank}(p_i)$,

Congruence Relation: $A \equiv_{\mathsf{congr}} B$, iff one can be obtained from the other by renaming of bound variables, without causing variable collisions, and re-ordering of the assignment equations within acyclic recursion constructs (or without any changes, i.e., $A \equiv_{\mathsf{congr}} A$).

---

[2] Note that without the acyclicity requirement, we have an extended formal language L^{λ}_{r}, which admits cyclic recursion, and is not in the subject of this paper.

The Reduction Calculus of L^{λ}_{ar} is provided by a set of reduction rules (see, Moschovakis [10]) for simplifying any given term A into its *canonical form* $\mathsf{cf}(A)$, which, intuitively, represents the algorithm for calculating the meaning of A (if it exists) in the simplest way, from the basic facts that are computed incrementally and "stored" in locations (recursion variables). Two major results are essential to the algorithmic senses of the terms:

Canonical Form Theorem: For each term A, there is a unique, up to congruence, irreducible term denoted by $\mathsf{cf}(A)$, such that:

1. $\mathsf{cf}(A) \equiv A$ or $\mathsf{cf}(A) \equiv A_0 \,\mathsf{where}\, \{p_1 := A_1, \ldots, p_n := A_n\}$
2. $A \Rightarrow \mathsf{cf}(A)$

Referential Synonymy Theorem: Two terms A, B are referentially synonymous, $A \approx B$, iff there are explicit, irreducible terms (of appropriate types), $A_0, A_1, \ldots, A_n, B_0, B_1, \ldots, B_n,\ n \geq 0$, such that:

1. $A \Rightarrow_{cf} A_0 \,\mathsf{where}\, \{p_1 := A_1, \ldots, p_n := A_n\}$,
2. $B \Rightarrow_{cf} B_0 \,\mathsf{where}\, \{p_1 := B_1, \ldots, p_n := B_n\}$,
3. $\models A_i = B_i$, for all $i = 0, \ldots, n$, i.e., $\mathsf{den}(A_i)(g) = \mathsf{den}(B_i)(g)$ for all variable assignments g.

Intuitively, $A \approx B$ iff the same algorithm "computes" their denotations (which are equal), in case when A and B have algorithmic senses; otherwise (i.e., A and B are immediate), A and B have the same denotations. Some other practical abbreviations that are used through the paper:

1. $\tau_1 \times \tau_2 \rightarrow \tau \equiv (\tau_1 \rightarrow (\tau_2 \rightarrow \tau))$, where $\tau_1, \tau_2, \tau \in \mathit{Types}$
2. $A(B_1, B_2) \equiv [A(B_1)](B_2) : \tau$, where
   $A : (\tau_1 \rightarrow (\tau_2 \rightarrow \tau))$, $B_1 : \tau_1$ and $B_2 : \tau_2$
3. $\mathit{like}(\mathit{john}, \mathit{mary}) \equiv \mathit{like}(\mathit{john})(\mathit{mary})$
   Note that, in linguistics literature, the more popular variant of the order of the arguments is: $\mathit{like}(\mathit{john}, \mathit{mary}) \equiv \mathit{like}(\mathit{mary})(\mathit{john})$. This is inessential, especially when the terms are not associated with any specific NL syntax.
4. $\tau_1 \times \ldots \times \tau_n \rightarrow \tau \equiv (\tau_1 \rightarrow \ldots \rightarrow (\tau_n \rightarrow \tau))$,
   where $n \geq 1$ and $\tau_1, \ldots, \tau_n, \tau \in \mathit{Types}$
5. $A(B_1, \ldots, B_n) \equiv [A(B_1) \ldots](B_n) : \tau$, where
   $A : (\tau_1 \rightarrow \ldots \rightarrow (\tau_n \rightarrow \tau))$, $B_1 : \tau_1, \ldots, B_n : \tau_n$ $(n \geq 1)$

3 β-Reduction in L^{λ}_{ar}

The calculus of the referential synonymy of L^{λ}_{ar} has the expected rules (see Moschovakis [10], Table 8), except for the β-reduction: a very weak form of β-reduction is valid for the referential synonymy of L^{λ}_{ar}:

$$\big(\lambda u\, C\big)(v) \approx C\{u :\equiv v\}, \text{ where} \qquad (\beta\text{-reduction})$$

C is explicit and irreducible,

u and v are pure variables, and the substitution $C\{u :\equiv v\}$ is free.

Given that *like* is a constant, and j and m are recursion variables (locations): the following statements are valid in the calculus of referential synonymy:

$$like \approx \lambda x \left(\lambda y \, like(x, y)\right) \tag{1a}$$

$$\begin{aligned} like(j, m) \equiv like(j)(m) &\approx \Big(\Big(\lambda x \left(\lambda y \, like(x)(y)\right)\Big)(j)\Big)(m) \\ &\equiv \left(\lambda x \, \lambda y \, like(x, y)\right)(j, m) \end{aligned} \tag{1b}$$

No reduction rule applies to the above terms, to justification of the referential synonymy equivalences. The β-reduction of L^{λ}_{ar} does not apply to the right hand side of the referential synonymy (1b). The equivalences (1a) and (1b) are valid by the Referential Synonymy Theorem, because the terms on both sides of each of the referential equivalences are explicit and irreducible, and have the same denotations for all variable assignments.

$$\lambda x \, like(x, x)(mary) \Rightarrow \lambda x \, like(x, x)(m) \text{ where } \{m := mary\} \tag{2a}$$

$$like(mary, mary) \Rightarrow like(m_1, m_2) \text{ where} \{m_1 := mary, \, m_2 := mary\} \tag{2b}$$

The terms in (2a) are not referentially synonymous to the ones in (2b), since their canonical forms do not comply with the Referential Synonymy Theorem. This is a good argument in favour of the restricted form of β-reduction. Intuitively, the terms in (2a) and (2b) do not carry the same information.

4 Expressiveness of L^{λ}_{ar}

As proved in Moschovakis [10], L^{λ}_{ar} is a proper extension of Gallin's TY_2, and, thus, of Montague's IL, as seen by the following terms, from Moschovakis [10]:

$$\text{John loves his wife and he honors her.} \xrightarrow{\text{render}} \tag{3a}$$

$$love(john, wife(his)) \,\&\, honors(he, her) \tag{3b}$$

$$\xrightarrow{\text{co-index}}_{\lambda} \lambda x_j \Big[love(x_j, wife(x_j)) \,\&\, honors(x_j, her)\Big](john) \tag{3c}$$

$$\xrightarrow{\text{co-index}}_{\lambda} \lambda x_j \Big[\lambda y_w \left[love(x_j, y_w) \,\&\, honors(x_j, y_w)\right](wife(x_j))\Big](john) \tag{3d}$$

The term (3d) is an IL term, which predicates a property (represented by the λ-subterm) to the denotation of *john*. The propositional content of the NL sentence, which is a conjunction of two declarative sentences, is more adequately represented by a recursion term, e.g., by the following one:

$$\text{John loves his wife and he honors her.} \xrightarrow{\text{render}} \tag{4a}$$

$$love(john, wife(his)) \,\&\, honors(he, her) \tag{4b}$$

$$\xrightarrow{\text{co-index}}_{\text{ar}} \left[love(j, w) \,\&\, honors(j, w)\right] \text{ where } \{j := john, \, w := wife(j)\} \tag{4c}$$

$$\begin{aligned} \Rightarrow \left[L \,\&\, H\right] \text{ where } \{&L := love(j, w), \, H := honors(j, w), \\ &j := john, \, w := wife(j)\} \end{aligned} \tag{4d}$$

The terms in (3d) and (4d) are not algorithmically equivalent, i.e., they are not equivalent with respect to referential intension. In addition, by a corollary of a fundamental theorem of L^{λ}_{ar} (Moschovakis [10], sec.3.24): (4d) is not algorithmically equivalent to any explicit λ-term, since at least one location occurs (actually, two: j and w) in more than one part of its canonical form.

5 Antecedent–Anaphora Relations

Typed λ-calculus with recursion terms offers more favoring arguments in comparison to other classical typed λ-calculi. It gives a possibility for distinguishing between co-reference, i.e., co-denotation relations, and antecedent–anaphora relations.

5.1 Co-indexing of Arguments: Antecedent–Anaphora Relations

In L^{λ}_{ar}, there are two different ways to *co-index* argument slots for representation of anaphoric relations. Firstly, as typically in λ-calculi, co-indexing can be represented by filling up several argument slots, in a sub-term, with the same pure variable, and λ-abstraction over that pure variable. Let us call this kind of indexing λ-*co-indexing.* It was done in the sequence of terms (3c) and (3d), and in the following term:

$$\text{Mary likes herself.} \xrightarrow{\text{render}} \lambda x\, like(x,x)(mary) \tag{5a}$$

$$\Rightarrow_{\mathsf{cf}} \lambda x\, like(x,x)(m) \text{ where } \{m := mary\} \tag{5b}$$

Alternatively, co-indexing can be represented by filling up several argument slots, in a sub-term (often, but not always, that is the head part of a recursion term), with the same location variable; the respective antecedent term is assigned to that location. Let us call this kind of indexing ar-*co-indexing.* This was done in the term (4c), and in the following term:

$$\text{Mary likes herself.} \xrightarrow{\text{render}} like(m,m) \text{ where } \{m := mary\} \tag{6}$$

The terms (5b) and (6) are referentially, i.e., algorithmically equivalent. Which one is more appropriate can be decided by a specific computational syntax of NL, or other linguistic investigations.

5.2 Co-denotation vs Antecedent–Anaphora

The co-denotation of different occurrences of sub-expressions is expressed faithfully by two different locations. For example, two different occurrences of the proper name Mary in one expression, as in the NL sentence (7), give rise to co-denotation (typically called co-reference) relation. The co-denotation relation is demonstrated by the assignment of the same explicit and irreducible term *mary* to the different locations m_1 and m_2.

$$\text{Mary likes Mary.} \xrightarrow{\text{render}} like(m_1,m_2) \text{ where } \{m_1 := mary,\ m_2 := mary\} \tag{7}$$

5.3 Two Kinds of Antecedent–Anaphora Relations

The language L_{ar}^{λ} offers further distinctions between semantic representations of expressions with pronominal sub-expressions.

Antecedent–anaphora via co-indexing. Capturing antecedent–anaphora via co-indexing was presented above, with the terms (5b) and (6). The term (4d) represents one of the possible interpretations of the NL sentence "John loves his wife and he honors her". In a typical utterance of this sentence, the occurrences of the pronouns "his", "he", and "her" participate in anaphoric relations, with their respective antecedents in the same sentence. This is captured by a co-indexing operator, applied over terms by using recursion variables, that identifies arguments with recursion variables, and thus represents antecedent–anaphora relation. However, it is possible for the same sentence to be used differently, depending on utterance situations, so that these (occurrences of) pronouns denote other individuals that are provided by the context discourse, some of whom are not mentioned in the sentence. Without any specific utterance, this sentence has an abstract linguistic meaning, where the pronouns are semantically underspecified.

Antecedent–anaphora via speaker's reference. Using free recursion variables of L_{ar}^{λ} gives the possibility for representation of quantifier scope underspecification, as demonstrated in Loukanova [7]. Rendering pronominal expressions by sub-terms that have occurrences of free locations, represents their semantic "underspecification" and leaves the resolving of the pronominal denotations to the context of use.

$$\text{John loves his wife and he honors her.} \xrightarrow{\text{render}} \tag{8a}$$

$$\begin{aligned}&\big[L \,\&\, H\big] \text{ where } \{L := love(j, w),\ H := honors(h_1, h_2),\\ &\qquad j := john,\ w := wife(h_3)\} \qquad \text{(underspecified)}\end{aligned} \tag{8b}$$

The underspecified term (8b) gives rise to different specifications:

$$\begin{aligned}&\big[L \,\&\, H\big] \text{ where } \{L := love(j, w),\ H := honors(h_1, h_2),\\ &\qquad j := john,\ w := wife(h_3),\ h_1 := j,\ h_2 := w,\ h_3 := j\}\end{aligned} \tag{9a}$$

$$\begin{aligned}&\big[L \,\&\, H\big] \text{ where } \{L := love(j, w),\ H := honors(h_1, h_2),\\ &\qquad j := john,\ w := wife(h_3),\ h_1 := j',\ h_2 := w',\ h_3 := j''\}\end{aligned} \tag{9b}$$

The terms (4d) and (9a), are not referentially synonymous, although, they are denotationally equivalent Neither of them is referentially synonymous with any explicit term. Assuming that j, j', and j'', w, w' are pairwise different constants, the referential synonymy of (9a) and (9b) will depend on the denotations of these constants, by the Theorem of Referential Synonymy.

6 Conclusion

Underspecified semantic representation became major effort of contemporary research, see Bunt [2] for a comprehensive overview of the field. A semantic theory that supports partiality does not represent inherent language ambiguities per se. Such semantic theories supporting partiality are, for example, the original, and classic by now, Situation Semantics (see, e.g., Barwise and Perry [1], and Devlin [4]), which was followed by Muskens' Partial Type-theoretic Grammars (see Muskens [11]).

The unique expressiveness of the language and theory of L_{ar}^{λ} offers powerful, from computational perspective, algorithmic treatment of Frege's distinction between sense and denotation. L_{ar}^{λ} has, strictly stronger than Montague's IL (Montague [8]), potentials for applications to computational semantics of natural language, for which it was originally developed by Moschovakis [10]. In addition, Loukanova [7] demonstrated that L_{ar}^{λ} has inherent facilities for representing semantic underspecification of quantifiers. The logical calculus of L_{ar}^{λ} has a very restricted form of β-reduction with respect to preserving referential, i.e., computational synonymy. This paper demonstrates that the restricted β-reduction brings in adequateness for semantic representation of co-referential and antecedent relations. In combination with open L_{ar}^{λ} terms that have free recursion variable (locations), the restricted β-reduction gives rise to various distinctions between antecedent-anaphora and co-denotation relations. In particular, the rich expressiveness of the L_{ar}^{λ} language offers unique distinctions between semantic representations of natural language expressions with pronominal sub-expressions.

References

1. Barwise, J., Perry, J.: Situations and Attitudes. MIT Press, Cambridge (1983)
2. Bunt, H.: Semantic Underspecification: Which Technique for What Purpose? In: Bunt, H., Muskens, R. (eds.) Computing Meaning. Studies in Linguistics and Philosophy 83, vol. 3, pp. 55–85 (2007)
3. Gallin, D.: Intensional and Higher-Order Modal Logic. North-Holland, Amsterdam (1975)
4. Devlin, K.: Situation theory and situation semantics. In: Gabbay, D., Woods, J. (eds.) Handbook of the History of Logic, vol. 7, pp. 601–664. Elsevier, Amsterdam (2008)
5. Kalyvianaki, E.: Factual Content in Algorithmic Natural Language Semantics. In: Proceedings of the ESSLLI 2007 Student Session. Ireland (2007)
6. Kalyvianaki, E.: Algorithmic Natural Language Semantics. PhD Thesis
7. Loukanova, R.: Typed Lambda Language of Acyclic Recursion and Scope Underspecification. In: Muskens, R. (ed.) Workshop NDTTG, Proceedings, Ireland (2007)
8. Montague, R.: Formal Philosophy. In: Thomason, R.H. (ed.) Selected papers of Richard Montague. Yale University Press (1974)
9. Moschovakis, Y.N.: Sense and denotation as algorithm and value. In: Oikkonen, J., Vaananen, J. (eds.) Lecture Notes in Logic, vol. 2, pp. 210–249. Springer, Heidelberg (1994)
10. Moschovakis, Y.N.: A logical calculus of meaning and synonymy. Linguistics and Philosophy 29, 27–89 (2006)
11. Muskens, R.: Meaning and Partiality. CSLI Publications (1995)

Permutation Languages in Formal Linguistics

Benedek Nagy

Department of Computer Science, Faculty of Informatics
University of Debrecen, Debrecen, Hungary
4010 Debrecen, PO Box. 12
nbenedek@inf.unideb.hu

Abstract. Derivations using branch-interchanging and the language family ($\mathbf{L}_{perm}$) obtained by context-free and interchange ($AB \rightarrow BA$) rules are analysed. This family is not closed under intersection with regular sets, therefore the obtained family of languages $\mathbf{L}_{perm \cap reg}$ is also interesting. Some non-trivial properties are shown. Important languages of mildly context-sensitive classes are shown to be belonging to $\mathbf{L}_{perm \cap reg}$. Closure properties and other properties are detailed. Relations to partial and semi-commutations and to parallel processes are shown.

Keywords: languages, Chomsky hierarchy, interchange (permutation) rules, semi-linear languages, mildly context-sensitivity, commutations.

1 Introduction

The Chomsky type grammars and the generated language families are one of the most basic and most important fields of theoretical computer science. The context-free grammars (and languages) are widely used due to their generating power and simple way of derivation. The derivation trees represent the context-free derivations. There is a big gap between the efficiency of context-free and monotone (or context-sensitive) grammars. There are very simple non-context-free languages as well, for instance $\{a^{2^n} | n \in \mathbb{N}\}$, $\{a^n b^n c^n | n \in \mathbb{N}\}$, $\{ww | w \in \{a, b\}^*\}$ etc. So, context-free grammars are not enough to describe several phenomena of the world [6] including natural languages, but the context-sensitive family is too large and has some unpleasant properties. Therefore several branches of extensions of context-free grammars were introduced generating proper subfamilies of context-sensitive languages. Some examples are the matrix grammars [1], indexed grammars [2], macro grammars [9], programmed grammars [20] and time-variant grammars [21]. In ordered grammars the rules are partially ordered. In most of these approached the derivations are controlled by some other mechanism (see [6] as a textbook on the topic).

In this paper we consider some variations of permutation grammars. A permutation grammar is a special monotone grammar in which each non-context-free production contains the same multiset of symbols in both sides. Sillars introduced this concept [24], moreover it was shown that context-sensitive rules are more effective than the composition of permutations along with the type of

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 504–511, 2009.
© Springer-Verlag Berlin Heidelberg 2009

substitution allowed by context-free rules. It is mentioned, also in [3], that the generative power of these grammars is strictly between the context-free and context-sensitive grammars. Further Mäkinen analysed these grammars concentrating on conditions such that the generating power remains in the context-free class. The language class $\mathbf{L}_{perm}$ generated by context-free and type $AB \to BA$ rules were analysed in [16]. We note that these languages are of interest for concurrency theory and theory of parallel processes, as well, where the order of some processes can be interchanged. In semi-commutation two consecutive letters of a word can be interchanged if they were in a given order [5]. In literature the so-called partial commutation is also known [23] as an operation interchanging two consecutive letters of a word independently of their order. Partial commutations have been intensively investigated over the two last decades in connection with parallel processing. Semi-commutation can be viewed as a generalisation of partial commutation. They are intensively studied in connection to Petri-nets. Allowing to permute any two letters of the words its commutative closure is obtained. Commutative context-free languages were analysed by their connections to Petri-nets and Basic Parallel Processes in [7]. Commutative (parallel) context-free automata is described in [14] in connection to Basic Process Algebra and Rewrite Transition Systems as well. A survey on this topic is [4]. Some further result about commutative context-free languages can be found in [8]. The commutative closure of every semi-linear language is a permutation language, therefore all commutative context-free languages (that is the same set as the family of commutative regular languages) are in $\mathbf{L}_{perm}$. Pentus showed that Lambek grammars have the same power as context-free grammars [18]. Commutative versions of this grammar were also studied as extensions to recognize mildly context-sensitive classes of languages. A survey on this topic is [15], where type logical grammars are discussed.

In this paper we introduce a related family of languages. Since $\mathbf{L}_{perm}$ is not closed under intersection by regular languages, the obtained family $\mathbf{L}_{perm \cap reg}$ is analysed here. It has some non-usual properties, such as non closure under regular operations. Linguistic examples will also be presented. Conclusions and some open problems close the paper.

2 Basic Notions and Notations

In this section some definitions about formal language theory are recalled and our notations are fixed. Let $\mathbb{N}$ denote the non-negative integers. We assume that the reader is familiar with the concepts of Chomsky-type grammars and languages (for details see [10,19,22]). A grammar is a construct $G = (N, T, S, H)$, where N, T are the non-terminal and terminal alphabets, $S \in N$ is the initial letter. H is the set of rules. We use λ to denote the empty word. For any word and sentential form u we will use $|u|$ to denote its length. Two grammars are weakly equivalent (we will use shortly the term equivalent) if they generate the same language modulo the empty word. Depending on the possible structures of the derivation rules various classes of grammars are defined, such as

- monotone grammars: each rule $v \to u$ satisfies the condition $|v| \leq |u|$, where $v \in (N \cup T)^* N (N \cup T)^*$ and $u \in (N \cup T)^*$.
- context-free grammars: for every rule the next scheme holds: $A \to v$ with $A \in N$ and $v \in (N \cup T)^*$.
- regular grammars: each derivation rule is one of the following forms: $A \to w$, $A \to wB$; where $A, B \in N$ and $w \in T^*$.

A language is regular/ context-free/ context-sensitive if it can be generated by a regular/ context-free/ monotone grammar, respectively. For these families the notations $\mathbf{L}_{reg}$, $\mathbf{L}_{CF}$ and $\mathbf{L}_{CS}$ are used. It is well known that $\mathbf{L}_{reg} \subsetneq \mathbf{L}_{CF} \subsetneq \mathbf{L}_{CS}$.

Now, the definition of partial commutations is recalled from [23]. Let the alphabet T and a set Z containing pairs of letters (a, b) $(a, b \in T)$ be given such that $a \neq b$ for any pairs of Z. Let the words w and w' over T be given such that $w = uabv$ and $w' = ubav$ with $(a, b) \in Z$ or $(b, a) \in Z$ and $u, v \in T^*$. The transformation of w into w' is called a partial commutation. For a word $w \in T^*$ the set of words w' such that there exists a sequence of partial commutations which transforms w into w' is called the commutation class of w. The concept semi-commutation is an extension of partial commutations. A semi-commutation is a relation defined over an alphabet T [5]. It is an irreflexive relation: it is a subset of $T \times T \setminus \{(a, a) | a \in T\}$. A semi-commutation system is a rewriting system associated to a semi-commutation relation Θ: (T, P), where P is the set $\{ab \to ba | (a, b) \in \Theta\}$. It can be viewed as a modification of a partial commutation using ordered pairs, i.e., if $w = uabv$, then $w' = ubav$ can be obtained only if $(a, b) \in Z$.

The shuffle operation is an extension of concatenation. Let v and u be two words over an alphabet T. The shuffle of v and u is defined as $u \bowtie v = \{u_1 v_1 ... u_n v_n | u = u_1 ... u_n, v = v_1 ... v_n, u_i, v_i \in T^*, 1 \leq i \leq n, n \geq 1\}$. Consequently, the shuffle of languages L_1 and L_2: $L_1 \bowtie L_2 = \{w |$ there are words $u \in L_1$ and $v \in L_2$ such that $w \in u \bowtie v\}$.

Let the terminal alphabet $T = (a_1, a_2, ..., a_n)$ be ordered. For each word its Parikh-vector is assigned: $\Psi : T^* \to \mathbb{N}^n$, $\Psi(w) = (|w|_{a_1}, |w|_{a_2}, ..., |w|_{a_n})$, where $w \in T^*$ and $|w|_{a_i}$ is the number of occurrences of the letter a_i in w. The set of Parikh-vectors $\Psi(L) = \{\Psi(w) | w \in L\}$ is called the Parikh-set of the language L. Two languages are letter-equivalent if and only if their Parikh-sets are identical. A language is linear (in Parikh-sense) if its Parikh set can be written in the form of a linear set: $\left\{ \underline{v}_0 + \sum_{i=1}^{m} x_i \underline{v}_i \,\middle|\, x_i \in \mathbb{N}, \text{ for some } \underline{v}_j \in \mathbb{N}^n, 0 \leq j \leq m \right\}$. A language is semi-linear (in Parikh-sense) if its Parikh set is a finite union of linear sets. We will use shortly the term semi-linear for this property. It is known that every context-free language is semi-linear (Parikh theorem). Non semi-linear context-sensitive languages are known (For instance $L_\square = \{a^{n^2} | a \in T\}$).

Now we leave the Chomsky hierarchy. We will consider other language classes.

3 Families of Permutation Languages

In this section the definitions of the language classes are presented. First we recall a definition of the basic class $\mathbf{L}_{perm}$.

Definition 1. *A grammar* $G = (N, T, S, H)$ *is a permutation grammar if every rule in* H *is one of the following forms:* $A \to v$ *(context-free rules, with* $A \in N$ *and* $v \in (N \cup T)^*$*) and* $AB \to BA$ *(interchange or permutation rules), with* $AB \in \{NT, TN, NN\}$*. The generated language family* $\mathbf{L}_{perm}$ *is the class of permutation languages.*

We note here that in [3] longer permutation rules are also defined with no fixed points, while in [11,16] the length of both sides of these rules are 2. In all of these papers the permutation rules contain only non-terminal symbols. Our definition is slightly modified, i.e., it is allowed that one of the symbols of a permutation rule is a terminal. Therefore we shall prove that it coincides with the original definition.

Proposition 1. *The class of languages that can be generated by grammars* (N, T, S, H) *in which each non-context-free rule is of the form* $AB \to BA$ *with* $A, B \in N$ *is exactly* $\mathbf{L}_{perm}$*.*

Proof. It is obvious that our definition is an extension of the definition from [11,16], i.e., all permutation grammars according to that definition are also permutation grammars according to ours. Now, consider the other direction. Let $G = (N, T, S, H)$ be a permutation grammar by Definition 1. There can be two types of permutation rules of H which contains terminal symbol: $aB \to Ba$ and $Ab \to bA$. They can be eliminated as follows: For each terminal a a new nonterminal X_a is introduced. All occurrences of the terminal a in the original rules are replaced with X_a and a new rule $X_a \to a$ is added to H. In this way the obtained grammar G' fulfills the original old definition and generates the same language. ☐

The non-context-free language containing all words over the alphabet $\{a, b, c\}$ containing the same number of a's, b's and c's can easily be generated by a permutation grammar. In [16] it was shown that for every permutation grammar there is an equivalent one with only rules of the following forms: $A \to a$, $A \to B$, $A \to BC$, $AB \to BA$ with $a \in T$, and $A, B, C \in N$. This form is called the normal form of permutation grammars. It is also known that all permutation languages are semi-linear in Parikh-sense. Since there is no forced use of the permutation rules, every permutation language contains the context-free basis language that can be obtained by only the context-free rules of the grammar. To overcome this limitation one may need additional techniques. Now, we define a special derived family of permutation languages.

Definition 2. *Let* $\mathbf{L}_{perm \cap reg}$ *denote the family of languages obtained by the intersection of a permutation and a regular language. So* $L \in \mathbf{L}_{perm \cap reg}$ *if and only if* $L = L' \cap L''$ *where* $L' \in \mathbf{L}_{perm}$ *and* $L'' \in \mathbf{L}_{reg}$*.*

This family is a non-trivial extension of $\mathbf{L}_{perm}$, since $\mathbf{L}_{perm}$ is not closed under intersection by regular languages as we shall see later. Using $L'' = T^*$ one can see that all permutation languages are in $\mathbf{L}_{perm\cap reg}$.

4 Examples: Linguistic Applications – Place in the Chomsky-Hierarchy

In this section we study the three basic languages contained by the mildly context-sensitive classes. They are shown to be in $\mathbf{L}_{perm\cap reg}$. We also place our language classes in the Chomsky hierarchy.

As we already mentioned each language L of $\mathbf{L}_{perm}$ contains a context-free basis language. This basis language is letter-equivalent to L. Based on this observation and by noting that the semi-linear language class is closed under intersection we infer the following statement.

Proposition 2. *All languages in $\mathbf{L}_{perm\cap reg}$ are semi-linear.*

All the linguistically important languages are semi-linear (the sets of mildly context-sensitive languages are included in the set of semi-linear languages). The language *multiple agreement* $L_{abc} = \{a^n b^n c^n | n \in \mathbb{N}\}$ is important from linguistical point of view. It is one of the famous mildly context-sensitive languages. This language does not contain any context-free languages in such a way that they are letter-equivalent. Therefore $L_{abc} \notin \mathbf{L}_{perm}$. As we already mentioned the language containing the same number of a's, b's and c's is in $\mathbf{L}_{perm}$. Thus, L_{abc} can be obtained by the intersection of this language and the regular set $a^*b^*c^*$. Therefore $L_{abc} \in \mathbf{L}_{perm\cap reg}$.

With similar method one can obtain another important language, the *cross-agreement* $L_{abcd} = \{a^n b^m c^n d^m | n, m \in \mathbb{N}\}$: The language generated by the grammar $(\{A, B, C, D, S\}, \{a, b, c, d\}, S, \{S \to AD, A \to \lambda, A \to aAC, D \to \lambda, D \to BDd, CB \to BC, B \to b, C \to c\})$ intersected by the regular language $a^*b^*c^*d^*$ results L_{abcd}.

We consider now the double marked version of the *copy* language $L_{wccw} = \{wccw | w \in \{a, b\}^*\}$. Consider the grammar $(\{A, B, C, D, E, S\}, \{a, b, c\}, S, \{S \to ED, E \to EaA, E \to EbB, E \to C, Aa \to aA, Ab \to bA, Ba \to aB, Bb \to bB, Ca \to aC, Cb \to bC, AD \to DA, BD \to DB, A \to a, B \to b, C \to c, D \to c\})$. The generated language intersected by the regular language $(a+b)^*cc(a+b)^*$ gives L_{wccw}.

Now, by our previous results, we are placing the new language classes in the Chomsky-hierarchy:

Theorem 1

$$\mathbf{L}_{CF} \subsetneq \mathbf{L}_{perm} \subsetneq \mathbf{L}_{perm\cap reg} \subsetneq \mathbf{L}_{CS}.$$

All inclusions are strict, the language $\{w \in \{a, b\}^* || w|_a = |w|_b\}$ is context-free. The language $\{w \in \{a, b, c\}^* || w|_a = |w|_b$ and $|w|_a = |w|_c\}$ is a non context-free, but it is included in $\mathbf{L}_{perm}$. Moreover $L_{abc} \notin \mathbf{L}_{perm}$, but $L_{abc} \in \mathbf{L}_{perm\cap reg}$. Over the 1-letter terminal alphabet the interchange rules do not add anything to the generating power of context-free grammars. Moreover in this case only regular languages can be generated since it is known that in this case all semi-linear languages are regular. Thus $\mathbf{L}^{(1)}_{perm} = \mathbf{L}^{(1)}_{reg}$ and $\mathbf{L}^{(1)}_{reg}$ is closed under intersection $\mathbf{L}^{(1)}_{perm\cap reg} = \mathbf{L}^{(1)}_{reg}$. (We used the notation $\mathbf{L}^{(1)}$ for language classes over the unary alphabet.) The context-sensitive language $L_{\square}$ is not linear and it is not semi-linear, therefore it is not in $\mathbf{L}_{perm\cap reg}$.

5 Properties of Permutation Language Classes

Let L be a language and Z be a set of ordered pairs of letters from the alphabet. Let the language $L_{semi-com(Z)} = \{w' | w \in L$ and w' can be obtained from w with semi-commutations by $Z\}$. Let $Z' = Z \cup \{(x,y) | (y,x) \in Z\}$, then $L_{semi-com(Z')} = L_{parcom(Z)}$ is the language of the commutation classes of the words of L. Let $Z'' = \{(x,y) | x \neq y,\ x,y \in T\}$. Then $L_{com} = L_{semi-com(Z'')}$ is the commutative closure of L.

Theorem 2. *Let $L \in \mathbf{L}_{CF}$ and Z be a set of ordered pairs of letters from the alphabet. Then $L_{semi-com(Z)}, L_{parcom(Z)}, L_{com} \in \mathbf{L}_{perm}$.*

Proof. Let G be a grammar generating L in a Chomsky normal form. Let us introduce new non-terminals A_a for all $a \in T$, and change all occurrences of the terminals of the original rules to the respective non-terminal, moreover add new rules $A_a \to a$. Now there is a non-terminal for each terminal from which only the terminal symbol can be derived. For all ordered pairs (a,b) of Z we add the interchange rule $A_a A_b \to A_b A_a$ to the rule set. The obtained permutation grammar generates exactly $L_{semi-com(Z)}$. As partial commutations and commutative closure can be expressed as semi-commutations the rest of the proof is obvious. □

Since the commutative closure of a semi-linear language is the commutative closure of a regular language, all semi-linear commutative languages are in $\mathbf{L}_{perm}$.

In [16] it is shown that $\mathbf{L}_{perm}$ is closed under the regular operations, under shuffle, under mirror image. It is not closed under intersection by regular languages and under complement. Now another closure property of $\mathbf{L}_{perm}$ and some closure properties of $\mathbf{L}_{perm\cap reg}$ will be analysed.

Proposition 3. *The language family $\mathbf{L}_{perm}$ is closed under homomorphisms.*

Proof. Let G be a permutation grammar in normal form that generates L. Let $h : T^* \to (T')^*$ be a morphism. Let G' be the grammar obtained from G by changing every rule $A \to a$ (where $a \in T$) to rule $A \to v$ where $v = h(a)$. It is clear that the generated language is exactly $h(L)$. □

Proposition 4. *The family $\mathbf{L}_{perm\cap reg}$ is closed under intersection by regular languages.*

Proof. Let $L \in \mathbf{L}_{perm \cap reg}$ and $L' \in \mathbf{L}_{reg}$. We show that $L \cap L' \in \mathbf{L}_{perm \cap reg}$. By definition, there are languages $L_1 \in \mathbf{L}_{perm}$ and $L_2 \in \mathbf{L}_{reg}$ such that $L = L_1 \cap L_2$. Let $L'' = L_2 \cap L'$. Since the family of regular languages is closed under intersection, $L'' \in \mathbf{L}_{reg}$. Then using the associativity of the intersection:

$$L \cap L' = (L_1 \cap L_2) \cap L' = L_1 \cap (L_2 \cap L') = L_1 \cap L''$$

which proves our statement. □

Theorem 3. *The family $\mathbf{L}_{perm \cap reg}$ is not closed under the regular operations union and concatenation.*

Proof. Now we give hints for the proofs. They are based on the following fact. If $L_1, L_2 \in \mathbf{L}_{perm \cap reg}$ such that different regular languages must be used in the intersection to obtain them, then there is no appropriate regular language for their combination. Consider the following examples. Let $L_1 = \{a^n b^n c^n d^n e^n f^n | n \in \mathbb{N}\}$ and $L_2 = \{a^n w f^n | w = ve^n \bowtie d^m,$ where $v = b^m \bowtie c^m$ for some $n, m \in \mathbb{N}, m > n\}$. $L_1, L_2 \in \mathbf{L}_{perm \cap reg}$. Let us consider $L_\cup = L_1 \cup L_2$ that is not in $\mathbf{L}_{perm \cap reg}$. Now, let $L_1 = \{w \in \{a, b, c, d, e, f\}^* || w|_a + |w|_b = |w|_c + |w|_d$ and $|w|_a + |w|_b = |w|_e + |w|_f\}$. L_1 is a permutation language and so it is in $\mathbf{L}_{perm \cap reg}$. Let $L_2 = \{a^n b^{2n} c^{3n} d^{4n} e^{5n} f^n | n \geq 0\}$. $L_2 \in \mathbf{L}_{perm \cap reg}$. Consider the language $L_c = L_1 L_2$ that is not in $\mathbf{L}_{perm \cap reg}$. □

Based on the previous non closure properties, noting that the concatenation of two languages of $\mathbf{L}_{perm \cap reg}$ is in the same class if disjoint alphabets T and T' are used, we can easily infer the following fact.

Corollary 1. *The language family $\mathbf{L}_{perm \cap reg}$ is not closed under homomorphism and inverse homomorphism.*

Finally, we can conclude the following statement.

Corollary 2. *The language family $\mathbf{L}_{perm \cap reg}$ is not a trio and not an anti-trio.*

6 Conclusions, Further Remarks

We dealt with permutation languages, allowing some reordering among the branches of the derivation trees extending the Chomsky hierarchy with two classes between the context-free and context-sensitive ones. We showed that all the three famous mildly context-sensitive languages are in $\mathbf{L}_{perm \cap reg}$. Unusual closure properties of this class were presented. It is an interesting open problem to answer whether the parsing problem of the languages of $\mathbf{L}_{perm}$ is polynomial. If so, then $\mathbf{L}_{perm \cap reg}$ is a mildly context-sensitive family. It is also open to know if emptiness and other related problems are decidable for $\mathbf{L}_{perm \cap reg}$.

Acknowledgements

This work has been supported by the Agency for Research Fund Management and Research Exploitation (KPI) and National Office for Research and Technology (NKTH) of Hungary.

References

1. Abraham, S.: Some questions of phrase-structure grammars. Comp. Linguistics 4, 61–70 (1965)
2. Aho, A.: Indexed grammars–An extension of context-free grammars. J. ACM 15, 647–671 (1968)
3. Book, R.V.: On the structure of context-sensitive grammars. Int. Journal of Computer and Information Sciences 2, 129–139 (1973)
4. Burkart, O., Caucal, D., Moller, F., Steffen, B.: Verification on infinite structures. In: Bergstra, J.A., Ponse, A., Smolka, S.A. (eds.) Handbook of Process Algebra, pp. 545–623. Elsevier, Amsterdam (2001)
5. Clerbout, M., Latteux, M., Roos, Y.: Semi-Commutations. In: Diekert, V., Rozenberg, G. (eds.) The book of traces, pp. 487–552. World Scientific, Singapore (1995)
6. Dassow, J., Păun, G.: Regulated rewriting in formal language theory. EATCS Monographs on Theoretical Computer Science, vol. 18. Springer, Berlin (1989)
7. Esperanza, J.: Petri Nets, Commutative Context-Free Grammars, and Basic Parallel Processes. Fundamenta Informaticae 30, 23–41 (1997)
8. Kortelainen, J.: Remarks about Commutative Context-Free Languages. Journal of Computer and System Sciences 56, 125–129 (1998)
9. Fischer, M.: Grammars with macrolike productions. Ph.D. Dissertation, Harvard University, Cambridge, Massachusetts (1968)
10. Hopcroft, J.E., Ullmann, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading (1979)
11. Mäkinen, E.: On permutative grammars generating context-free languages. BIT 25, 604–610 (1985)
12. Mateescu, A.: On context-sensitive grammars. In: Martin-Vide, C., Mitrana, V., Păun, G. (eds.) Formal languages and applications. Studies in Fuziness and Soft Computing, vol. 148, pp. 139–161. Springer, Heidelberg (2004)
13. Mateescu, A., Rozenberg, G., Salomaa, A.: Shuffle on trajectories: syntactic constraints. Theoretical Computer Science 197, 1–56 (1998)
14. Moller, F.: A Taxonomy of Infinite State Processes. Electonic Notes in Theoretical Computer Science 18, 20 pages (1998)
15. Moortgat, M.: Categorial Type Logics. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language, pp. 93–177. Elsevier, Amsterdam (1997)
16. Nagy, B.: Languages generated by context-free grammars extended by type AB →BA rules. Journal of Automata, Languages and Combinatorics (accepted)
17. Penttonen, M.: One-sided and two-sided context in formal grammars. Information and Control 25, 371–392 (1974)
18. Pentus, M.: Lambek grammars are context-free. In: Proc. of LICS, pp. 429–433 (1993)
19. Révész, G.E.: Introduction to Formal Languages. McGraw-Hill, New York (1983)
20. Rosenkrantz, D.: Programmed grammars and classes of formal languages. J. ACM 16, 107–131 (1969)
21. Salomaa, A.: On grammars with restricted use of productions. Ann. Acad. Sci. Fenn., Ser. A, No. 454 (1969)
22. Salomaa, A.: Formal Languages. Academic Press, New York (1973)
23. Schott, R., Spehner, J.-C.: Two optimal parallel algorithms on the commutation class of a word. Theoretical Computer Science 324, 107–131 (2004)
24. Sillars, W.: Formal properties of essentially context-dependent languages. Ph.D. Dissertation, Pennsylvania State Univ. (1968)

Thomas: Practical Applications of Agents and Multiagent Systems

Javier Bajo[1] and Juan M. Corchado[2]

[1] Pontifical University of Salamanca, Compañía 5, 37002 Salamanca, Spain
`jbajope@upsa.es`
[2] University of Salamanca, Plaza de la Merced S/N, 37008 Salamanca, Spain
`corchado@usal.es`

Abstract. This paper presents a brief summary of the contents of the special session on practical applications held in the framework of IWANN 2009. The special session has been supported by the THOMAS (TIN2006-14630-C03-03) project and aims at presenting the results obtained in the project, as well as at exchanging experience with other researchers in this field.

Keywords: Multiagent systems, Agents technology.

1 Introduction

Research on Agents and Multi-Agent Systems has matured during the last decade and many effective applications of this technology are now deployed. An international forum to present and discuss the latest scientific developments and their effective applications, to assess the impact of the approach, and to facilitate technology transfer, has become a necessity.

The Special Session on Practical Applications of Agents and Multiagent Systems (http://iwann.usal.es/mas), in the framework of the 10th International Work-Conference on Artificial Neural Networks (IWANN 2009) provides a unique opportunity to bring multi-disciplinary experts and practitioners together to exchange their experience in all aspects of Agents and Multi-Agent Systems, especially those concerned with applications, methods, techniques and tools for open multi-agent systems.

The session intends to bring together researchers and developers from industry and academic world to report on the latest scientific and technical advances on the application of multi-agent systems, discuss and debate the major issues, and showcase the latest systems using agent based technology. It is a multidisciplinary discipline that may attract scientist and professionals to IWANN and to provide a different field in which to apply ANN based technology. It promotes a forum for discussion on how agent-based techniques, methods, and tools help system designers to accomplish the mapping between available agent technology and application needs. Other stakeholders should be rewarded with a better understanding of the potential and challenges of the agent-oriented approach.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 512–513, 2009.
© Springer-Verlag Berlin Heidelberg 2009

This special session has been supported by the THOMAS research project (TIN2006-14630-C03-03), which aim is to advance and contribute methods, techniques and tools for open multiagent systems, principally in the aspects related to organisational structures. THOMAS is a coordinated project in which the University of Salamanca, the Technical University of Valencia and the University of Rey Juan Carlos cooperate to find new solutions in the field of the multiagent systems. This special session provides a framework to disseminate the results obtained in the project and to exchange knowledge with other researchers in the field of the agent technology.

2 Special Session on Practical Applications of Agents and Multiagent Systems Details

This volume presents the papers that have been accepted for the 2009 edition. These articles capture the most innovative results and this year's trends: Multi-Agent Systems (MAS) Applications: commerce, health care, industry, internet, etc.; Agent and MAS architectures; Agent development tools; MAS middleware; Agent languages; Engineering issues of MAS; Web services and agents; Agents and grid computing; Real-time multi-agent systems; Agent-based social simulation; Security in MAS; Trust and reputation in MAS; Improving user interfaces and usability with agents; Information recovery with MAS; Knowledge management with MAS; Software Agents in Ubiquitous Computing; Agent technologies for Ambient Intelligence; Software Agents in Industry; Planning and scheduling in MAS; Agent Technologies for Production Systems; Service-Oriented Computing and Agents; Agents for E-learning and education; Mobile computation and mobile Communications. Each paper has been reviewed by three different reviewers, from an international committee composed of 15 members from 7 different countries, and the members of the IWANN 2009 committee. From the 22 submissions received, 17 were selected for full presentation at the conference.

3 Special Session Acknowledgements

We would like to thank all the contributing authors, as well as the members of the Program Committee and the Organizing Committee for their hard and highly valuable work. Their work has helped to contribute to the success of this special session. We also would like to thank the IWANN 2009 for giving us the opportunity of organizing the special session, for their help and support. Thanks for your help, the special session on practical applications of agents and multiagent systems wouldn't exist without your contribution.

Acknowledgments. This work has been supported by the MEC TIN2006-14630-C03-03 project.

INGENIAS Development Process Assisted with Chains of Transformations

Iván García-Magariño, Rubén Fuentes-Fernández, and Jorge J. Gómez-Sanz

Dept. Software Engineering and Artificial Intelligence
Facultad de Informática
Universidad Complutense de Madrid, Spain
`ivan_gmg@fdi.ucm.es, ruben@fdi.ucm.es, jjgomez@sip.ucm.es`

Abstract. This paper presents a chain of model transformations to guide and support the application of the INGENIAS development process. The *MTGenerator* tool generates these transformations with a *Model Transformation By-Example* approach, that is, automatically from pairs of model prototypes. The *MTGenerator* has the advantage over similar approaches and tools of being able to generate many-to-many transformation rules between non-connected graphs of elements. The work in this paper sets the foundation for future research on software processes aided by integrated standard transformations. Two case studies illustrate the generation of transformations with the *MTGenerator* and its use in the process. They also show the applicability of the approach to different application domains.

Keywords: Model-Driven Development, Model Transformation, Transformation By-Example, Multi-Agent Systems, INGENIAS.

1 Introduction

The *INGENIAS* methodology [4] for the development of *Multi-agent Systems* (MAS) is founded on *Model-driven Development* (MDD) [6] principles. The generation of *Model Transformations* (MTs) in INGENIAS has been facilitated with the *MTGenerator* tool. This tool follows a Model-transformation By-Example (MTBE) [7] approach. It generates MTs from model prototypes, which are example pairs of the source and target models that the target MT has to consider. In addition, its *matching mechanism* allows the propagation of attribute information from the source to the target models of the MT. The *MTGenerator* overcomes the limitation of other MTBE tools [7,8,9] to generate many-to-many transformation rules. This kind of transformations is required in MAS modeling, as the two case studies presented in this paper show. The *MTGenerator* tool has been recently introduced in [3] and it is available at our website[1].

This paper considers the application of the *MTGenerator* to specify standard transformations that support the INGENIAS methodology. In any software process, there are repetitive tasks that happen once and again in different projects.

---

[1] http://grasia.fdi.ucm.es (in "Software" section)

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 514–521, 2009.
© Springer-Verlag Berlin Heidelberg 2009

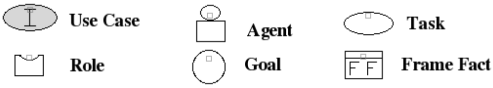


Fig. 1. Some relevant concepts of the *INGENIAS* notation

They depend more on the process than on model contents. Designers can use the tool to generate MTs that automate for them these tasks. In this way, they save effort in the application of the methodology.

To illustrate this approach, the paper presents a specific chain of MTs to guide the INGENIAS development. These MTs are specified in a high level of abstraction with pairs of model prototype examples, and the generated code for the *ATLAS Transformation Language* (ATL) is omitted for the sake of brevity. The chain of MTs starts from the use-cases and workflows diagrams. From them, and following the INGENIAS process recommendations, it creates other parts of the specifications using MTs. The paper also shows how to use the *MTGenerator* tool to generate the MTs that implement that chain. Finally, it applies this chain of MTs to two different practical cases: the crisis-management and the Delphi method to evaluate document-relevance. The first practical case [2] manages a crisis situation in a city. A poisonous material has been released, and the central services are not enough to help all the affected citizens. For this reason, a MAS coordinates the citizens, so the citizens with medical capabilities can help other citizens. The second practical case [1] considers the development of a MAS to evaluate documents following the *Delphi* method. In this MAS, a moderator agent coordinates a group of experts agents, by means of rounds of questionnaires. Although the two practical cases have different scopes, the same chain of transformations for the INGENIAS process can be applied to both of them. Therefore, the *MTGenerator* can produce MTs that designers can reuse in several MAS development cases. In order to ease the understanding of the models used in this discussion, Fig. 1 presents the relevant elements of the INGENIAS notation.

The remaining of the paper is organized as follows. Section 2 describes the chain of MTs that guide the INGENIAS development. It interleaves the introduction of the MTs with their application in the two practical cases. Section 3 discusses some conclusions about the chain of MTs and future lines of research.

2 A Chain of Model Transformations to Guide the INGENIAS Development

This section describes a process to model in INGENIAS with the support of MTs. These MTs guide and assist designers in the specification, although they have the possibility of manual edition if necessary. This process tries to be general in two ways: MTs can be reused in other similar INGENIAS developments; designers can define other MTs with our MTBE tool to support additional needs, following similar steps to those done in the current project.

2.1 The Steps of the Chain of MTs

An INGENIAS development based on MTBE begins choosing which models are the seeds to generate partially other models. The chain of MTs considered here regards the use-case and workflow models as these sources. Use cases provide in INGENIAS the most abstract modeling level for requirements and are used for early elicitation. From these types of source models, the chain generates several modeling artifacts for the development. The steps are the following:

1. Definition of the use cases (first group of source models).
2. A MT creates the *Role* definitions from the use cases.
3. Definition of the workflows (second group of source models).
4. A MT creates the definitions of *Tasks* from the workflows.
5. For each *Task*, a MT creates an INGENIAS *Code Component*.
6. A MT creates the *Interactions* according to the workflow.
7. A MT creates the agent deployments.
8. A MT generates the tests.

The remaining sub-sections focus on the MTs for steps 2 and 4. Since the concepts of use cases, roles and workflows are widely present in different domains, we expect a high reusability of the resulting MTs beyond the INGENIAS methodology. For these MTs, the sub-sections show how to generate them using MTBE with the *MTGenerator*. This generation consists on specifying the example pairs of source and target models for each MT. The application of the resulting MTs is illustrated with the Crisis and Delphi case studies. The remaining MTs of the chain have also been applied to both practical cases, but they are omitted in this paper for the sake of brevity. However, an appendix with the remaining MTs can be downloaded from the Grasia website[2].

2.2 MT for Generating Roles from Use Cases

In INGENIAS, the use case diagrams (see an example in Fig. 3) usually include roles, use cases where they participate, and goals satisfied by these cases. A later step must link roles with the goals they pursue. In addition, there must be at least one agent that plays each role (see Fig. 5). Since these tasks are common in INGENIAS developments, this work proposes to create automatically the role definitions with a MT.

The model pairs in Fig. 2 can generate this MT. In the first pair, the source model contains a role and the target model an agent that plays this role. The agent identifier is obtained from the role identifier. The propagation of the value from the source to the target model uses the matching-control mechanism provided by the tool. In the second pair, the source model contains a goal connected to a role through a use case, and the target model has the same goal directly linked with the role. In this way, the MT links roles with their goals according to the existent use cases.

---

[2] http://grasia.fdi.ucm.es (in "Papers" section)

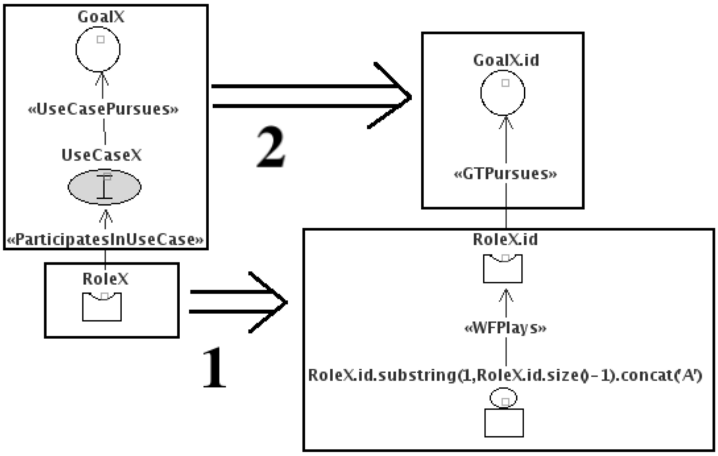


Fig. 2. MT from use cases to role and agent definitions. Each arrow represent a pair of source and target models. The numbers in rules indicate the order of application.

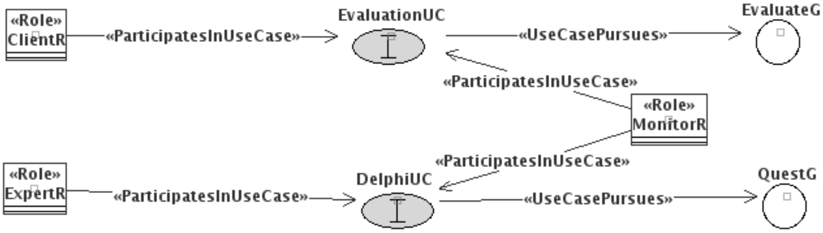


Fig. 3. Use cases for the Delphi MAS

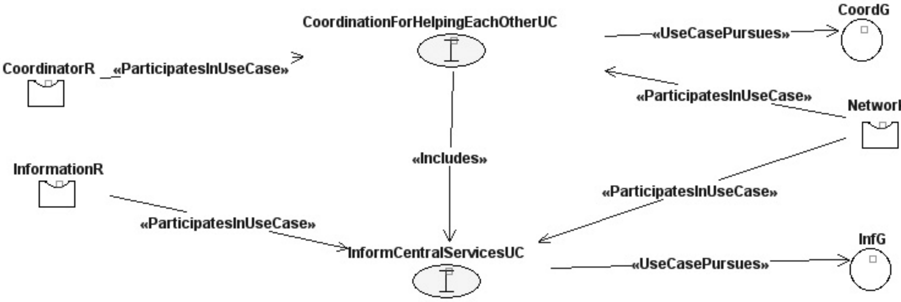


Fig. 4. Use cases for the Crisis MAS.

The proposed MT can be applied to different INGENIAS specifications. This paper applies in two practical cases, the Crisis and Delphi ones cited in the introduction. The MT generates from the use cases of the Delphi MAS (see

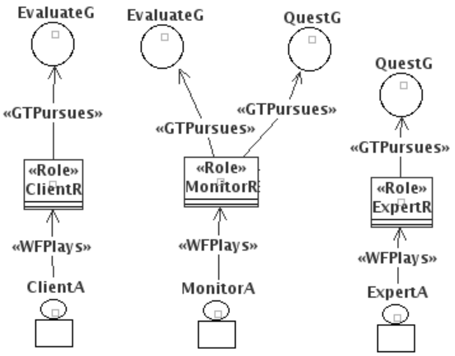


Fig. 5. *Roles* and *Agents* of the Delphi MAS

Figure 3) the definition of the corresponding agents (see Figure 5). Similarly, the use cases of Crisis MAS (see Figure 4) are transformed in the definition of its corresponding roles.

This example shows that the same MT can be applied to several MAS designs, which use different names of the modeling elements. That is, the MTBE approach of the tool generates general MTs applicable to different models.

2.3 MT to Get the Tasks Related with a Workflow

The initial definition of a workflow in INGENIAS regards it as a chain of tasks (see an example in Fig. 8) connected with *WFConnects* relationships. Designers refine this definition giving the inputs and outputs of each task in different diagrams. This refinement commonly distinguishes between the first task of the workflow and the others.

An event launched by some user's application usually triggers the first task of a workflow. Fig. 6 shows the pairs to specify the MT required for this refinement. Its source model contains the initial task of the workflow. The target model

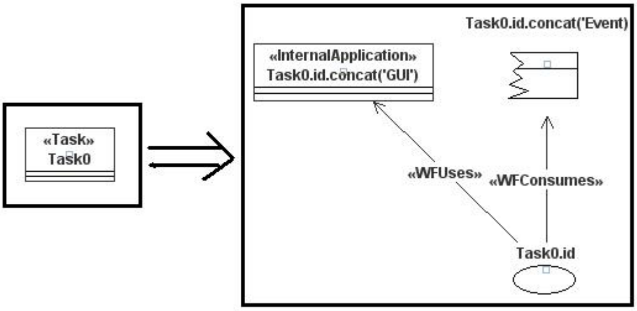


Fig. 6. MT for the initial task of a workflow

shows the task consuming an event from an INGENIAS internal application that represents the user's application. To determine which the initial task of a workflow is, engineers use the criteria that it is the only one not preceded by any other task. However, this cannot be specified within the source model of the prototype pair. Designers of the MT has to define manually a constraint in the code of the generated MT with the Object Constraint Language (OCL) [5]. According to the INGENIAS metamodel, the expression of the constraint is:

```
MMA!WFConnects.allInstances()->select(e|WFConnectstarget.
      WFConnectstargettask.id=cin.id).isEmpty()
```

The constraint declares that there is no other task connected with this one with a *WFConnects* relationship. Therefore, it is the first of the workflow. This constraint should not need to be defined if the MTBE tool included negative examples. The Grasia group is planning to include this feature in a next release of the *MTGenerator* tool.

The non-initial tasks of a workflow usually consume a *frame fact* produced by the previous task, and produce a *frame fact* to be consumed by the next task. Fig. 7 presents the MT that creates these frame facts and their connections with their related tasks. The first rule creates the frame fact and the *WFConsumes* relation, and the second one creates the *WFProduces* relation. The MT propagates the names in the source models to the target models by means of the matching mechanism. The name of each new frame fact is the concatenation of the names of the producer task and the consumer task with a suffix. The first rule of this MT is not applied to the initial task of a workflow because there is no other task that precedes it.

The rules in the MT can be applied several times to the same task if it precedes or is preceded by several tasks. For instance, the example in Fig. 8 includes the *EndOfRound* task that precedes two different tasks, so the second rule is applied twice to this task. In the other practical case (in Fig. 9), the task *CNNHelpT* also precedes two tasks.

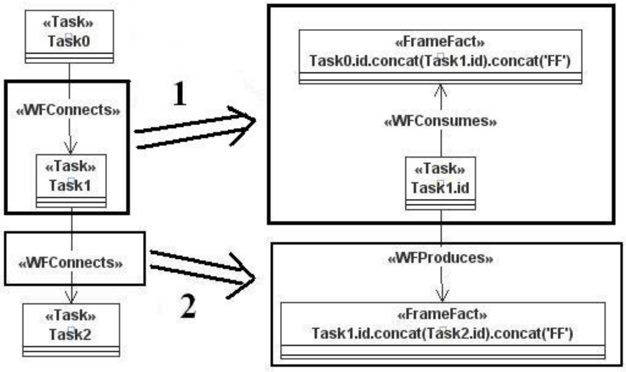


Fig. 7. MT for non-initial tasks of a workflow

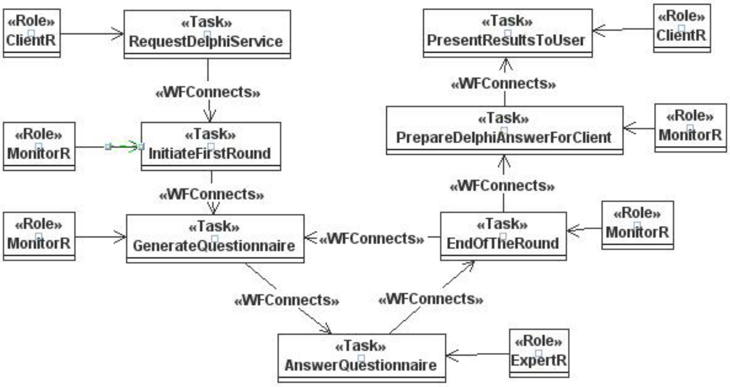


Fig. 8. Workflow for the Delphi MAS

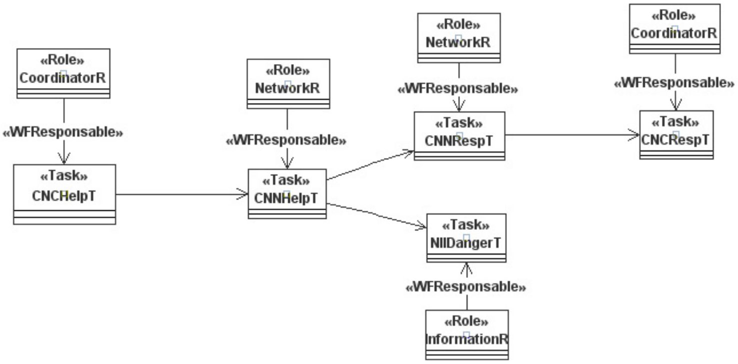


Fig. 9. Workflow for the Crisis MAS

As one can observe, the workflows of these case studies have different structures. The Delphi one can produce an iteration on the executed tasks, while the Crisis one is a workflow with a bifurcation. Despite their structural differences, the same MTs can be applied to both MAS specifications.

3 Conclusions and Future Work

This paper presents the practice of MTBE in the development of MASs, by using the INGENIAS methodology and the *MTGenerator* tool. It introduces a chain of MTs that can guide the INGENIAS development and uses *MTGenerator* to generate them. The resulting chain of MTs is applied to two case studies in order to show both the utility of the chain to guide the development, and the general applicability of the MTs generated by the tool.

The current experimentation with the *MTGenerator* has shown some open issues where the tool and its algorithm can be improved. They have to consider negative examples to reduce the need of specifying text constraints. In addition, further experiments are necessary to precisely assess the increment of productivity in the development of MASs with the execution of MTs that support the development.

Acknowledgements. This work is supported by the project "Agent-based Modelling and Simulation of Complex Social Systems (SiCoSSys)", funded by Spanish Council for Science and Innovation, with grant TIN2008-06464-C03-01.

References

1. García-Magariño, I., Gómez-Sanz, J.J., Agüera., J.R.P.: A Multi-Agent Based Implementation of a Delphi Process. In: The Seventh International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, Estoril Portugal, May 12-16, pp. 1543–1546 (2008)
2. García-Magariño, I., Gutiérrez, C., Fernández, R.F.: Organizing multi-agent systems for crisis management. In: 7th Ibero-American Workshop in Multi-Agent Systems (Iberagents 2008), Lisbon, Portugal, October 14, pp. 69–80 (2008)
3. García-Magariño, I., Rougemaille, S., Fuentes-Fernández, R., Migeon, F., Gleizes, M.-P., Gómez-Sanz, J.J.: A Tool for Generating Model Transformations By-Example in Multi-Agent Systems. In: 7th International Conference on Practical Applications of Agents and Multi-Agent Systems, PAAMS 2009 (2009) (to be published)
4. Gómez-Sanz, J.J., Fuentes, R., Pavón, J.: The INGENIAS Methodology and Tools. In: Agent-oriented Methodologies, pp. 236–276. Idea Group Publishing, USA (2005)
5. OMG. UML 2.0 OCL Specification. Object Management Group (2003)
6. Schmidt, D.C.: Model-Driven Engineering. IEEE Computer 39(2), 25–31 (2006)
7. Varró, D.: Model transformation by example. In: Nierstrasz, O., Whittle, J., Harel, D., Reggio, G. (eds.) MoDELS 2006. LNCS, vol. 4199, pp. 410–424. Springer, Heidelberg (2006)
8. Varró, D., Balogh, A.: Automating model transformation by example using inductive logic programming. In: Proceedings of the 2007 ACM symposium on Applied computing, pp. 978–984 (2007)
9. Wimmer, M., Strommer, M., Kargl, H., Kramler, G.: Towards Model Transformation By-Example. In: 40th Annual Hawaii International Conference on System Sciences. IEEE Computer Society, Los Alamitos (2007)

A Secure Group-Oriented Model for Multiagent Systems[⋆]

Jose M. Such[⋆⋆], Juan M. Alberola[⋆⋆⋆], Antonio Barella, Agustin Espinosa,
and Ana Garcia-Fornes

Departament de Sistemes Informàtics i Computació,
Universitat Politècnica de València, Camí de Vera s/n. 46022, València
`{jsuch,jalberola,tbarella,aespinos,agarcia}@dsic.upv.es`

Abstract. In this paper, a secure group-oriented model for Multiagent Systems is presented. This model is provided as a support from the Multiagent Platform level. The benefits introduced by the model are shown applying it to an existing MAS-based application.

1 Introduction

As stated in [10], Multiagent Systems (MAS) offer strong models for representing complex and dynamic real-world environments. The simulation of economies, societies and biological environments, for example, is typical application areas. On the MAS research scene there are methodologies that aim to structure agent development but it may be difficult to implement theoretical designs. Hence, it is important to stress the significance of applying theoretical development improvements in real scenarios (applications).

Real scenarios in MAS technologies are often large-scaled systems that needs Multiagent Platforms (MAP) scalable and efficient. Agent organizations becomes an important issue to be provided by a MAP which is focussed to execute large applications composed by thousands of agents, since they help us to simplify, structure, coordinate and easily develop MASs. Several research groups define theoretical proposals and methodologies oriented to agent organizations. However, only few MAPs (Jack[1], MadKit[2], Spade[3] or Zeus[4]) offers some kind of support related with agent organizations.

Real scenarios also demand systems which take into account security concerns. Security-related studies in the MAS research field have been growing in number over the last few years, as well as studies into intelligent autonomous agents and MAS based applications. This is mainly due to the fact that the understanding of the actual risk when using these sorts of applications is needed, since an agent's incorrect or inappropriate behaviour may cause non-desired effects such

---

[⋆] This work has been partially supported by CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, and projects TIN2008-04446 and PROMETEO/2008/051.

[⋆⋆] Enjoying a grant of Conselleria d'Empresa, Universitat i Ciència de la Generalitat Valenciana (BFPI06/096).

[⋆⋆⋆] Enjoying a grant of Ministerio de Ciencia e Innovación de España (AP2007-00289).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 522–529, 2009.
© Springer-Verlag Berlin Heidelberg 2009

as money and data loss. Therefore, security is a key issue that has to be taken into account when developing these sorts of applications, and the lack of security in some current MAS-based applications is one of the reasons why MAS technology is being slowly introduced into industry.

In this paper, a secure group-oriented model for MAS is presented. This model is provided as a support from the MAP level and is implemented in the Magentix MAP. The benefits of this model are shown addressing the current drawbacks of an existing MAS-based application (JGOMAS).

The rest of the article is organized as follows. Section 2 describes the secure group-oriented model. Section 3 gives the current JGOMAS design. Section 4 describes how the secure group-oriented model presented can address the current limitations of JGOMAS. Finally, in section 5 we present some concluding remarks and some improvements we could make as future work.

2 Magentix Multiagent Platform

Magentix MAP (a MultiAGENT platform integrated in LInuX)[5] aims to be scalable and efficient, mainly when it is executing large-scale MAS. This MAP has been developed using the services provided by the OS. Thus, one of the design decisions is that this MAP is written in C over the Linux OS. Magentix also provides advanced communication mechanisms such as agent groups, a manager to execute interaction protocols and a security mechanism to provide authentication, integrity, confidentiality and access control.

2.1 Group Model

The Organizational Units Manager (OUM) service provides support oriented to agent-group communication as a pre-support for agent organizations. An agent group is called organizational *unit* (from now on, unit) and can be seen as a blackbox from the point of view of external agents. Units can also be composed of nested units. Agents can interact with an agent unit in a transparent way, i. e. from the point of view of an agent outside the unit, there is no difference between interacting with a unit or with an individual agent. Interaction between an agent and a unit is carried out by the MAP through properties specified by the user. Each unit has some properties associated to it. As each agent of the platform has a unique name, each unit is identified in the platform by its *name*. In order to interact with any unit, user must specify one or more agents to receive the messages addressed to the unit: these agents are called *contact agents*. User can also specify the way in which these messages have to be delivered to the *contact agents*. This property is called the *routing type* and messages addressed to the unit will be delivered to the contact agents defined, according to one of these *routing types*:

- Unicast: The messages addressed to the unit are delivered to a single agent which is responsible for receiving messages. This type is useful when we want a single message entrance to the group. It could be useful if the group presents for example, a hierarchical structure, where the supervisor receives every message and distributed them to its subordinates.

- Multicast: Several agents can be appointed to receive messages. When a message is addressed to the unit, this message is delivered to any contact agent of the unit. It could be useful if we want to represent an anarchic scenario, where every message needs to be known by every agent without any kind of filter.
- Round Robin: There can be several agents appointed to receive messages. But each message addressed to the unit is delivered to a different contact agent, defined according to a circular policy. This type of routing messages it is useful when some agents offers the same service but we want to distribute the incomming requests to avoid the bottlenecks.
- Random: Several agents can be defined as contact agents. But the message is delivered to a single one, according to a random policy. As the previous type, it is useful for distributing the incomming requests, but it is not specified any kind of order for attending these requests.
- Sourcehash: Several agents can be the contact agents. But any given message is delivered to one of the agents responsible for receiving messages, according to the host where the message sender is situated. It is a load-balancing technique.

A unit also has a defined set of agents which make up the unit, called *members*. These agents can interact and coordinate each other and eachone plays a certain *role*. Finally, each unit has a *manager* associated to it. This agent is responsible for adding, deleting or modifying the members and contact agents. By default it is the agent which creates the unit and is the only one allowed to delete it.

All of this information regarding units in Magentix, is managed by the OUM, which stores it in the Global Unit Table (GUT). As the OUM is a distributed service, this table is replicated and synchronized on each host of the MAP. So far, the implemented version in Magentix allows every agent to search and create units and only the *manager* can modify the unit structure and delete it.

Interaction between agents and OUM service is carried out by the sending of messages. Message header and message content are represented as RDF models with XML syntax specification. We provide Magentix with an API to create, modify the structure or delete units.

2.2 Security Model

The Magentix MAP has a security model presented in [11] which is based on both the Kerberos protocol and the Linux OS access control. This model provides magentix with authentication, integrity and confidentiality. By means of this model each agent has an identity that it can prove to the rest of the agents and services of a running Magentix MAP.

With the introduction of the unit concept explained in the previous section, the security model of Magentix is revised. Magentix units are aimed at hiding the agents that are inside a unit, so that when an agent is communicating with a unit it does not need to know if it is communicating with an agent or a unit. Regarding security, an agent cannot use its identity when it is communicating on behalf of a unit, because in such a case the destination agent would know the identity of the former agent and no hiding would be possible. Moreover, a destination agent would need a mechanism to check that the agent that has sent a message on behalf of a unit is a member of the unit that it claims to be from.

Therefore, this model is extended in order to take Magentix units into account. In this way, the security infrastructure has been extended with the unit identity concept, so that all of the units created in the platform have a way to prove themselves when using the support provided by Magentix related to unit management.

With the introduction of the unit identity model, Magentix agents can now have three identity types:

- **Agent** identity. Its identity as an agent. This identity is created by the AMS when the agent is created.
- **User** identity. The identity of its owner, i.e, the identity of the user that created the agent.
- **Unit** identity. The identity of each unit that the agent is in.

An agent always has at least its **Agent** identity and its owner's **User** identity. Therefore, a Magentix agent is provided with more than one identity, so a way to indicate to the Magentix communication module which Kerberos credentials it has to use when sending a message is needed. This is done with a new field in the message header. If this field is in the message header of a message to be sent, the communication module tries to use the identity chosen; otherwise the corresponding agent identity is used. If the Kerberos credentials associated to the identity that the agent is requesting are not available, and the agent is trying to use an identity that it does not own for instance, the sending of the message fails.

Regarding access control in units, due to the way units are implemented in Magentix, only the agent that is the manager of the unit decides if a new agent can join the unit. Moreover, this decision can be based on the identity of the agent that is requesting being a member of the unit, because the agent identities are assured.

The manager of the unit also decides which agents are the contact agents. A contact agent is given with the identity of the unit, so that it can act on behalf of the unit. In this way, inter-unit relationships can be specified in order to define at communication level some well-known agent organizations models[8] such as hierachies, burocracies, federations, etc. The unit concept provides us with the control over the interactions directed to a unit and the unit identity concept provides us with the way in which an agent can act on behalf of a unit. For instance, regarding a simply hierarchy, the supervisor of the hierarchy can be mapped to the manager of the unit and the unique contact agent, so that only this agent can exchange messages on behalf of the unit. The subordinates can simply be mapped to member agents.

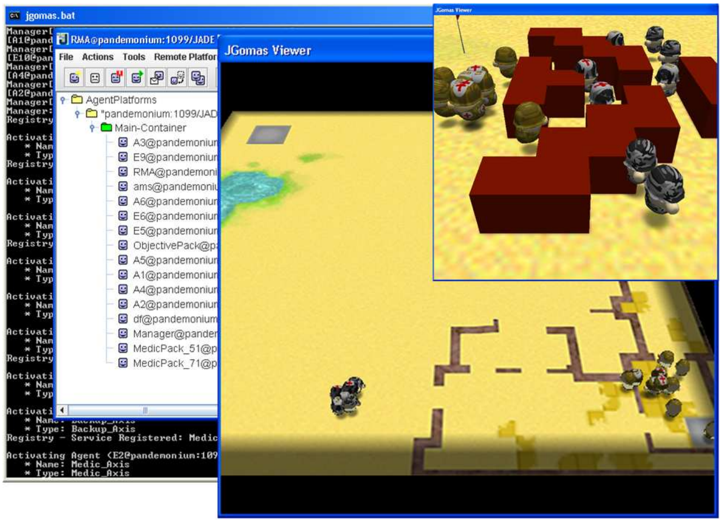
3 JGOMAS: Game-Oriented MAS

In order to show a suitable scenario for applying the secure group-oriented model explained in the previous sections, the JGOMAS application is presented. First, we describe the application in its current state, and then, we point out how the application can be improved by using the secure group-oriented model presented.

JGOMAS (an acronym for Game Oriented Multi-Agent System based on JADE [7]) is a framework that integrates a MAS and a virtual reality system as a scalable solution for Intelligent virtual environments (IVEs)[9]. Using JGOMAS as an IVE, it is possible to play a *capture-the-flag* (CTF) match. A CTF game is proposed as the kind of social interaction to simulate, where agents are grouped into two teams (allies and axis). The allies must go to the axis base, capture the flag, and take it to their base, in order to win the game. The axis agents must defend their flag against the allies and, if the flag is captured, they must return it to their base. There is a time limit for the allies to bring the flag to their base. If the time limit expires, the axis team wins the game. An example of execution is shown in Figure 1.

The architecture of JGOMAS is composed of three subsystems:

- a **MAP.** JGOMAS uses JADE as MAP. It simplifies the implementation of MASs through a middle-ware, so that an IVE designer has only to implement the intelligence of his agents, avoiding wasting time on low-level technical issues such as, for example, inter-agent communication.
- an **agent collection (MAS).** JGOMAS' MAS can be viewed as an abstraction upper layer over a MAP. Main classes of agents are *simulation*



```
 * Type: Backup_Axis
Registry - Service Registered: Medic
Activating Agent <E2@pandemonium:109
 * Name: Medic_Axis
 * Type: Medic_Axis
```

Fig. 1. JGOMAS Framework

controller and *inhabitant agents* (like player agents, for examle). Simulation controller is in charge of keeping the virtual environment's data, maintaining the consistency at any given time, while the inhabitant agents simulate beings (humans, animals, etc.) situated in the virtual world.
- a **visualization module.** One of the JGOMAS' main goals is for artificial intelligence and virtual reality systems to work independently. In fact, it is possible to run the JGOMAS MAS even if there are no graphic viewers connected. In order to make things easier for IVE designers, a basic graphic viewer displaying the 3D agents, objects, and the scenario in JGOMAS has been implemented.

A class inheritance hierarchy for agents has been defined in JGOMAS[6]. The *simulation controller* is a special agent in the MAS subsystem. In fact, there is only one running during a simulation. It can be viewed as a wrapper of the virtual environment, because it translates all agents' actions to the virtual world. In addition, this agent ensures that these actions follow specific rules, maintaining consistency throughout the simulation at all times. The *inhabitant* agents have both a physical model and intelligence. They are really the players of the current game. They have a set of basic predefined behaviors. However, the user can modify these behaviors or even add new ones. Every Inhabitant Agent performs a role and are specialized in: Soldier, Medic and FieldOps.

4 Improving JGOMAS by Replacing the MAP Subsystem

This section presents some of the limitations of JGOMAS and how they can be addressed. In this sense, we propose the use of Magentix as the MAP to be used by JGOMAS so that taking advantage of the secure group-oriented model provided by Magentix.

Although the original JGOMAS design take into account agent groups (allied and axis teams), the current implementation in Jade lacks this concept. In the current implementation, an agent has to search for allies in the Directory Facilitator where all of the agents in the platform register offering an allied service. For example, an allied agent medic has to register specifically that it offers the medic service but only for allied agents. Then, when an allied agent requires the medic service, it focusses the search specifying the allied-medic service. Every agent implements in the class whether it is an agent of either the allies team or the axis team. This ad-hoc implementation restricts the application to basic teams and known services. Therefore, an agent should be implemented whithout taking into account the team, it will simply join the corresponding team at run time.

Besides this drawback, JGOMAS currently does not take into account security concerns. Therefore, any agent of the system could act on behalf of another. This may not be a key issue if the application is used in a non-malicius environment. However, if the application is being used as a commercial application for exemple, or in an educational environment as is currently the case, it becomes a major concern. For instance, an agent could send messages as if it were the simulation controller agent, so that it is necessary to ensure the identity of every agent. Furthermore, as we introduce the requirement of agent groups, we should also ensure the identity of every agent group, because from the point of view of agents outside the group, the group is viewed as an individual agent.

Taking into account the above restrictions that arise from the current JGOMAS design, we propose a new design based on Magentix MAP, which resolves them. The unit concept provided by Magentix that we have shown in section (2.1), will allow us to group agents according our requirements, as previously discussed. With the introduction of Magentix MAP in the JGOMAS application, the allies and axis teams can be mapped directly to units. In this way, an agent can always interact with all of the agents in its team in a simple way. We can group every agent of the allied team inside the same unit. Thus, when an agent requires a medic agent, the search for the agents which offer the medic service is implicitelly restricted only to agents of the same unit. We can view the agents not only as individual agents but also as a group of agents, due to the fact that we can require an agent from our unit to help us and it makes no difference who it may be.

Moreover, as the agent and unit identities are assured by the Magentix security model, an agent can avoid cheating agents that try to act as members of the allies team but are really members of the axis team. We must point out that in this design, each agent could have two identities: its identity as agent and the identity of the unit that it is part of. We have to take into account this point because, as we do not want every agent of the unit to receive the messages addressed to the unit, we probably do not want every agent to be able to use the identity of the unit to send messages. Flexibility offered to specify the unit properties will allow us to specify which agents can send messages using the unit identity and which cannot. Using this model, when an agent receives a message from a unit, this agent does not have to check which agents are able to send messages using the unit identifier, because this identity is sure to be used by an authorized agent.

Regarding developers of JGOMAS agents, in order to organize its agent troops the unit concept is crucial. As explained in section 2.2, for example, agent hierarchies can be defined at communication level. Therefore, the allies and the axis teams can organize themselves as though they were an army composed of batallions that are simple hiearchies.

Finally, besides the funcionalities offered by Magentix to improve the JGOMAS design, Magentix architecture focusses on offering good performance and scalability especially when running large systems. With regards to the JGOMAS design and its efficiency, some tests with a lot of agents and hosts have been carried out using the current JGOMAS application and the performance became slower. Using Magentix we could probably run several hosts and thousands of agents with good response times because it is specifically designed to run these kinds of applications.

5 Conclusions

Agent group abstraction allows us to organize and coordinate agents in a MAS, specially in large systems. Moreover, security concerns are a key issue in real applications. In this paper, a secure group-oriented model is presented. This model is implemented in the Magentix MAP and provides us with the agent group abstraction and identities for both agents and agent groups.

The model is applied to a MAS-based framework for intelligent virtual environments. The current framework design has some limitations due to the lack of the agent group abstraction and security concerns. With the introduction of these concepts in the MAP, the framework can now make use of them, so that these limitations are addressed and the framework design is improved.

The model presented can be viewed as a pre-support for agent organizations. It allows defining organizational structures at communication level. However, it currently lacks some other agent organizations features like norms, prohibitions, obligations, etc. In this sense, we are working on extending the current model so that it allows the development of organization-based MAS.

References

1. Jack, `http://www.agent-software.com`
2. Madkit, `http://www.madkit.org`
3. Spade, `http://spade.gti-ia.dsic.upv.es/`
4. Zeus agent toolkit, `http://labs.bt.com/projects/agents/zeus/`
5. Alberola, J.M., Such, J.M., Espinosa, A., Botti, V., Garcia-Fornes, A.: Scalable and efficient multiagent platform closer to the operating system. Artificial Intelligence Research and Development 184, 7–15 (2008)
6. Barella, A., Carrascosa, C., Botti, V.: Agent architectures for intelligent virtual environments. In: 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 532–535. IEEE, Los Alamitos (2007)
7. Bellifemine, F., Poggi, A., Rimassa, G.: Jade: a fipa2000 compliant agent development environment. In: AGENTS 2001: Proceedings of the fifth international conference on Autonomous agents, pp. 216–217. ACM Press, New York (2001)
8. Horling, B., Lesser, V.: A survey of multiagent organizational paradigms. The Knowledge Engineering Review 19, 281–316 (2004)
9. Luck, M., Aylett, R.: Applying artificial intelligence to virtual reality: Intelligent virtual environments. Applied Artificial Intelligence 14(1), 3–32 (2000)
10. Luck, M., McBurney, P., Shehory, O., Willmott, S.: Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing). AgentLink (2005)
11. Such, J.M., Alberola, J.M., GarcÍa-Fornes, A., Espinosa, A., Botti, V.: Kerberos-based secure multiagent platform. In: Sixth International Workshop on Programming Multi-Agent Systems (ProMAS 2008), pp. 173–186 (2008)

Interactive Animation of Agent Formation Based on Hopfield Neural Networks[⋆]

Rafael Kelly[1] and Carmen Monroy[2]

[1] División de Física Aplicada, CICESE Carretera Tijuana–Ensenada Km.
107 Ensenada, B. C., 22800, Mexico
`rkelly@cicese.mx`
[2] Instituto Tecnológico de La Laguna, Apdo. Postal 49, Adm. 1,
Torreón Coahuila, 2007, Mexico
`maricarmen-monroy@hotmail.com`

Abstract. Formation of agents is of recent interest in computer sciences, robotics and control systems. Several goal formation strategies may be of interest according to the sensory capabilities of the agents. This paper addresses the formation of mobile agents in absolute positioning without order. In this paper a control system based on Hopfiled neural networks is proposed. The paper summarizes the control system and describes a JAVA-based application developed to visualize the control system behavior. This interactive animation tool improves the understanding of and intuition for a number of aspects dealing with the formation of agents such as agents dynamics. This tool allows to directly manipulate graphical representation of the systems such as initial configuration of the agents, and get instant feedback on the effects.

Keywords: Interactive simulation, agents, control systems, Hopfield network, neural networks, stability.

1 Introduction

The control of collaborative robotic agents to reach and maintain a desired formation is a recent issue in the control and robotics communities [1,2,3].

For the purpose of this paper, an agent is a point in the Cartesian plane with no kinematic constraints of motion [4]. Consider n agents denoted by $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n$. The configuration of agent i is described by $\boldsymbol{z}_i = [x_i \;\; y_i]^T \in \Re^2$ and the corresponding dynamics is $\dot{\boldsymbol{z}}_i = \boldsymbol{u}_i$ where

$$\boldsymbol{u}_i = \begin{bmatrix} u_{xi} \\ u_{yi} \end{bmatrix}$$

is the control input whose meaning is the agent velocity vector in the Cartesian plane of motion.

The position of all agents gives the group configuration denoted by $\boldsymbol{z} = [\,\boldsymbol{z}_1 \; \boldsymbol{z}_2 \; \ldots \; \boldsymbol{z}_n\,]^T \in \Re^{2n}$

[⋆] Work partially supported by CONACyT (grant 45826).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 530–536, 2009.
© Springer-Verlag Berlin Heidelberg 2009

and the corresponding model of n agents can be written as

$$\dot{\boldsymbol{z}} = \boldsymbol{u} \tag{1}$$

where $\boldsymbol{u}$ is the control input vector: $\boldsymbol{u} = [\, \boldsymbol{u}_1 \ \boldsymbol{u}_2 \ \ldots \ \boldsymbol{u}_n \,]^T \in \Re^{2n}$.

In the context of modern control theory, (1) represents the state equation, thus in order to complete the system description it remains to specify the output equation —in general, a nonlinear function of the state vector $\boldsymbol{z}$—. For the agents model, the output equation arises from the control problem statement and may depend on assumptions about the sensory capabilities of the agents, for instance, to determine the relative displacement or absolute placement of some or all of them. These constraints may derive in a hard control problem to resolve.

Formation of agents may be classified in three basic schemes: absolute positioning with order, absolute positioning without order, and relative positioning with order. The remaining of this paper is devoted to describe the absolute positioning without order formulation as well as a control systems to deal with, and to describe a Java–based tool to perform interactive animations.

2 Agent Formation Based on Hopfield Neural Networks

The formation of n agents in absolute positioning without order requires the specification of n desired position $\boldsymbol{z}_1^*, \cdots, \boldsymbol{z}_n^*$ stacked in the vector $\boldsymbol{z}^*$ as $\boldsymbol{z}^* = [\, \boldsymbol{z}_1^* \ \boldsymbol{z}_2^* \ \ldots \ \boldsymbol{z}_n^* \,]^T \in \Re^{2n}$.

However, no specific desired position is associated to any agent. The formation objective is that the agents reach desired positions regardless the order.

More formally, define the set Ψ as the permutations of n agents in n at a time. This set has $n!$ elements of dimension $2n$. The formation objective is to achieve

$$\lim_{t \to \infty} \operatorname{dist}(\boldsymbol{z}(t), \Psi) = 0 \tag{2}$$

where $\operatorname{dist}(\boldsymbol{z}(t), \Psi)$ denotes the smallest distance from $\boldsymbol{z}$ to any point in the set Ψ.

This paper proposes a control law based in the concept of associative memories [6,7]. The rational behind this idea is that each element of the set Ψ be seen as a pattern stored into the associative memory. Once a disturbed pattern —initial agents formation— is presented at the associative memory, the corresponding true pattern is recovered —the agents reach one of the formation options in Ψ—. This means that the formation objective (2) is achieved.

Based in above arguments, this paper proposes the following control law

$$\boldsymbol{u} = -A\boldsymbol{z} + W\boldsymbol{f}(\boldsymbol{z}) + \boldsymbol{b} \tag{3}$$

where $A, W \in \Re^{2n \times 2n}$ with $A > 0$, $\boldsymbol{b} \in \Re^{2n}$, and $\boldsymbol{f}(\boldsymbol{z}) \in \Re^{2n}$ is defined component wisely by

$$\boldsymbol{f}(\boldsymbol{z}) = \mathbf{tanh}(\boldsymbol{z}) = [\, \tanh(x_1) \ \tanh(y_1) \ \tanh(x_2) \ \tanh(y_2) \ \ \ldots \ \tanh(x_n) \ \tanh(y_n) \ \,]^T$$

The first step in the design is to select the matrices A and W and vector $\boldsymbol{b}$ in such a way that

$$-A\boldsymbol{x} + W\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{b} = \boldsymbol{0} \qquad \forall\ \boldsymbol{x} \in \Psi. \tag{4}$$

In general the design is a hard algebraic problem to be solved analytically. However several iterative learning procedures —training or learning rules— are available [6,7]. This issue is out of the scope of this paper. Assuming that the design was performed according to (4), then the closed–loop system dynamics obtained by substituting the control law (3) into the agents model (1) yields

$$\dot{\boldsymbol{z}} = -A\boldsymbol{z} + W\boldsymbol{f}(\boldsymbol{z}) + \boldsymbol{b}. \tag{5}$$

This is a nonlinear autonomous system which matches the structure of the well–known Hopfield's neural network [6].

It is worth noticing that in virtue of (4), the elements of Ψ are equilibria of (5). However other spurious equilibria may exist, some of them may be asymptotically stable. Therefore, a sufficient condition to ensure the formation control objective (2) in a local sense is by proving that all equilibria in the set Ψ are asymptotically stable. This implies that the agents will reach a desired formation provided that the initial configuration is sufficiently close of the desired one.

One approach to address the stability analysis is by invoking the Lyapunov's first method [5]. A sufficient condition reported by [8] to determine weather an equilibrium $\boldsymbol{z}^* \in \Psi$ is locally exponentially stable is by testing the positive definiteness of the matrix

$$A - WG(\boldsymbol{z}^*) \tag{6}$$

where

$$G(\boldsymbol{z}^*) = \begin{bmatrix} \operatorname{sech}^2(x_1^*) & 0 & \cdots & 0 \\ 0 & \operatorname{sech}^2(y_1^*) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \operatorname{sech}^2(y_n^*) \end{bmatrix}.$$

Thus, a simple test to show that an equilibrium $\boldsymbol{z}^* \in \Psi$ is asymptotically stable is by corroborating that the $2n$ eigenvalues of the symmetric part of $A - WG(\boldsymbol{z}^*)$ are positives.

3 Interactive Simulations

In the recent years, the advances in software and hardware have allowed to design tools with much better man–machine interaction, with intuitive graphical user interfaces and, more important perhaps, a high degree of interactivity. These tools are particularly useful in the study of dynamic systems in general, and specially in automatic control. Although many interactive tools have been developed for education in automatic control, see [11,12], for recent surveys, however the application of interactive tools for learning agents formation is limited.

The objective of this section is to present the application of an interactive tool of the previously described agent formation objective. This tool is focused on objects to explore different views of the system, manipulate views directly using the mouse, changing the initial configuration of the agents and immediately see the consequences on the system behavior. We have found that the tool is a valuable complement to understand the control of agents formation because the high degree of interactivity makes the tool stimulating and quickly captures the interest of the user.

One of these tools is Easy Java Simulations (EJS), a free of charge software, valuable to help create interactive simulations and animations in the Java language [13]. This software tool is target for both science and engineering students and teachers with basic programming skills but who cannot afford the big investment of time needed to create a complete graphical simulation; a summary of EJS and its application in automatic control is given in [14].

EJS provides also a catalog of graphical elements to help the user to create views of the simulated dynamic system for animation and continuous visualization that reacts to the user interaction. These elements include containers, buttons, text fields, boxes, sliders, etc. as well as elements for designing the view of the physical phenomenon like particles, bodies, vectors, etc. These graphical elements can be used to build the graphical interface just by drag–and–drop the mouse.

Dynamic systems characterized by Ordinary Differential Equations (ODE) can be simulated by EJS. These equations are described within EJS by using a self–contained facilities (ODE solvers and Java code editor). The equations of the physical laws that rule the phenomenon under study are written in a way similar that written on a blackboard. EJS provides facilities to declare the system variables, the initial conditions, and the way how the user interacts with the system and modifies the variable values.

A graphical interface was programming under EJS to illustrate the control system previously described. Three agents are graphically shown in the interface and despited as bee, dragon–fly and bat, respectively. The bee, dragon–fly and bat icons are in GIF (Graphics Interchange Format), a bitmap image format for pictures in compressed file. The remaining of this section is devoted to describe the interface.

The interface shows two areas. The main area is on the upper part of the window. This is devoted to present the animation of the agents motion. Thanks to the use of the mouse, the user can interact on–line by dragging any agents to an arbitrary initial location or to a new location without stopping or pausing the simulation.

3.1 Absolute Positioning without Order

Three agents are considered to illustrate the absolute positioning without order, that is, there is no numbering of the desired locations for the agents. The control law has the structure (3) with $A = I \in \Re^{6\times 6}$ and the desired absolute locations for the agents are

$$\boldsymbol{z}_1^* = \begin{bmatrix} -10 \\ 10 \end{bmatrix}; \quad \boldsymbol{z}_2^* = \begin{bmatrix} 10 \\ 10 \end{bmatrix}; \quad \boldsymbol{z}_3^* = \begin{bmatrix} 10 \\ -5 \end{bmatrix}.$$

The set Ψ of permutations has $n! = 6$ elements of dimension $2n = 6$, that is

$$\begin{aligned} \Psi &= \left\{ \begin{bmatrix} \boldsymbol{z}_1^* \\ \boldsymbol{z}_2^* \\ \boldsymbol{z}_3^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{z}_1^* \\ \boldsymbol{z}_3^* \\ \boldsymbol{z}_2^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{z}_2^* \\ \boldsymbol{z}_1^* \\ \boldsymbol{z}_3^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{z}_2^* \\ \boldsymbol{z}_3^* \\ \boldsymbol{z}_1^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{z}_3^* \\ \boldsymbol{z}_1^* \\ \boldsymbol{z}_2^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{z}_3^* \\ \boldsymbol{z}_2^* \\ \boldsymbol{z}_1^* \end{bmatrix} \right\}, \\ &= \left\{ \begin{bmatrix} -10 \\ 10 \\ 10 \\ 10 \\ 10 \\ -5 \end{bmatrix}, \begin{bmatrix} -10 \\ 10 \\ 10 \\ -5 \\ 10 \\ 10 \end{bmatrix}, \begin{bmatrix} 10 \\ 10 \\ -10 \\ 10 \\ 10 \\ -5 \end{bmatrix}, \begin{bmatrix} 10 \\ 10 \\ 10 \\ -5 \\ -10 \\ 10 \end{bmatrix}, \begin{bmatrix} 10 \\ -5 \\ -10 \\ 10 \\ 10 \\ 10 \end{bmatrix}, \begin{bmatrix} 10 \\ -5 \\ 10 \\ 10 \\ -10 \\ 10 \end{bmatrix} \right\}. \end{aligned}$$

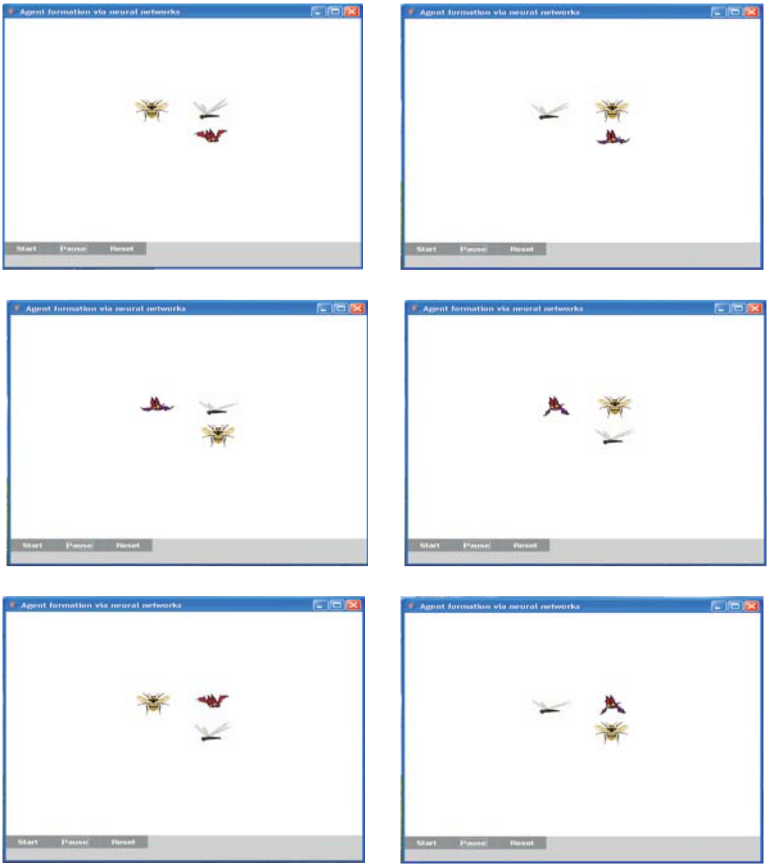


Fig. 1. Configurations associated to set Ψ: desired patterns

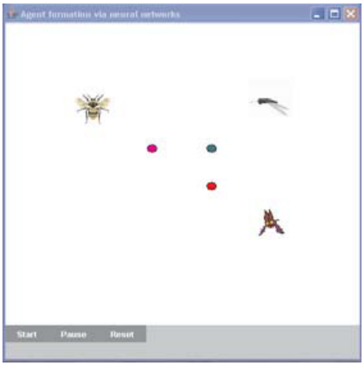


Fig. 2. graphical interface

Each element of Ψ corresponds to one of the configurations depicted in Figure 1.

After tedious but straightforward substitutions, it can be shown that the elements of Ψ satisfy the equilibria matching condition (5) and the asymptotic stability condition (6) with the following selection of W and b:

$$W = \begin{bmatrix} -991.0 & 1.0 & -1001.0 & 1.0 & -1001.0 & 1.0 \\ -998.5 & 8.5 & -998.5 & 1.0 & -998.5 & 1.0 \\ -1001.0 & 1.0 & -991.0 & 1.0 & -1001.0 & 1.0 \\ -998.5 & 1.0 & -998.5 & 8.5 & -998.5 & 1.0 \\ -1001.0 & 1.0 & -1001.0 & 1.0 & -991.0 & 1.0 \\ -991.0 & -6.5 & -991.0 & -6.5 & -991.0 & 1.0 \end{bmatrix}; \qquad \boldsymbol{b} = \begin{bmatrix} 1000 \\ 1000 \\ 1000 \\ 1000 \\ 1000 \\ 1000 \end{bmatrix}.$$

The interactive simulation of the system programmed under EJS has the graphical interface shown in Figure 2. The main area of the interface depicts the animation of the agents where the desired locations $\boldsymbol{z}^*$ are pictured in distinctive colors: magenta, green, and red. Agents are represented by a bee, a dragon–fly and a bat. The lower part of the interface contains the control buttons: Start, Pause, and Reset.

4 Conclusions

The coordinate control for formation of agents has applications in several fields such as in communications, robotics, and computer sciences. Several agent formations objective can be defined, each one leading to a specific formulation of a control problem. This paper summarized the Absolute positioning without order for agent formations and introduced a control law inspired form associative memories, espefically in the Hopfield Neural network.

In addition to the theoretical formulation this paper has exploited the open–source Java–based software EJS to develop interactive animations of the formation control system. This is a computational tool to visualize and interact with the system. This computational tool is a valuable complement to understand the behavior of the closed–loop systems because the high degree of interactivity makes the application stimulating and quickly captures the interest of the user.

References

1. Fierro, R., Song, P., Das, A., Kumar, V.: Cooperative control of robot formation. In: Murphey, R., Pardalos, P. (eds.) Cooperative Control and Optimization, Applied Optimization, ch. 5, vol. 66, pp. 73–93. Springer, Heidelberg (2002)
2. Arai, T., Pagello, E., Parker, L.E.: Editorial: Advances in multi–robot systems. IEEE Transactions on Robotics and Automation 18(5), 655–661 (2002)
3. Desai, J.P.: A graph theoretic approach for modeling mobile robot team formations. Journal of Robotic Systems 19(11), 511–525 (2002)
4. Lin, Z., Broucke, M., Francis, B.: Local control strategies for group of mobile autonomous agents. IEEE Trans. on Automatic Control 49(4), 622–629 (2004)
5. Khalil, H.: Nonlinear Systems Analysis, 2nd edn. Prentice-Hall, Englewood Cliffs (1996)
6. Haykin, S.: Neural networks: A comprehensive foundation. Macmillan College Publishing Company, Basingstoke (1994)
7. Hagan, M.T., Demuth, H.B., Beale, M.: Neural network design. PWS Publishing Company (1996)
8. Sudharsanan, S.I., Sundareshan, M.K.: Equilibrium characterization of dynamical neural networks and a systematic synthesis procedure for associative memories. IEEE Tranns. on Neural Networks 2(5) (1991)
9. Fredslund, J., Mataric, M.J.: A general algorithm for robot formation using local sensing and minimal communication. IEEE Transactions on Robotics and Automation 18(5), 837–846 (2002)
10. Gazi, V.: Formation control of a multi-agent system using nonlinear servomechanism. International Journal of Control 78(8), 554–565 (2005)
11. Dormido, S.: Control learning: present and future. In: Proc. of the 15th IFAC World Congress, Barcelona, Spain, pp. 81–103 (2002)
12. Kelly, R., Dormido, S., Monroy, C., Díaz, E.: Learning control of robot manipulators by interactive simulation. Robotica 23, 515–520 (2004)
13. Esquembre, F.: Easy Java Simulations: A software tool to create scientific simulations in Java. Computer Physics Communications 156, 199–204 (2004)
14. Sanchez, J., Esquembre, F., Martin, C., Dormido, S., Dormido–Canto, S., Canto, R.D., Pastor, R., Urquia, A.: Easy Java Simulations: an open–source tool to develop interactive virtual laboratories using MATLAB/Simulink. International Journal of Engineering Education 21(5), 798–813 (2005)

The INGENIAS Development Kit: A Practical Application for Crisis-Management

Iván García-Magariño, Celia Gutiérrez, and Rubén Fuentes-Fernández

D. Software Engineering and Artificial Intelligence
Facultad de Informática
Universidad Complutense de Madrid, Spain
`ivan_gmg@fdi.ucm.es, cegutier@fdi.ucm.es, ruben@fdi.ucm.es`

Abstract. The INGENIAS Development Kit (IDK) supports the development of fully functional Multi-agent Systems (MASs) from specification models, following a model-driven approach. This paper presents a practical application about crisis-management, in order to provide a full example of application of the IDK tool; and consequently, the specification and the code of this system are included in the IDK 2.8 distribution. The presented application manages the crisis situation of a city, in which, a poisonous material is released, and the central services are not enough to heal all the affected people. The software engineering process of the presented MAS application covers the following phases: specification, model design, implementation, and testing. Both the number of interactions and the number of participants are economized in order to increase the network efficiency, due to the real-time necessity of the MAS application.

1 Introduction

The INGENIAS Development Kit (IDK) [2] allows practitioners to create Multi-agent Systems (MASs) by defining specification models and generating the programming code. The goals of this work is to present a practical application of the last IDK distribution (2.8). For this purpose, a crisis-management case study is selected, and a solution is proposed. The presented MAS is included in the IDK 2.8 distribution, which can be downloaded from *Grasia* web[1].

In the crisis-management [1] case study, a poisonous material has been accidentally released into a city. The number of affected people is very high to be managed only with a centralized solution. The official medical services are not enough to heal all the affected people. Thus, a distributed solution is necessary, and the collaboration among the people on the ground is also necessary.

The presented solution is based on the following facts: the citizens with medical capabilities can help affected citizens who are close enough; the citizens can be quickly warned of the poison-affected locations to avoid them; the central official system must be informed of all the poison-affected locations; the communications should be efficient. Therefore, the three goals of the presented MAS

---

[1] http://grasia.fdi.ucm.es (in "Software" → "INGENIAS Development Kit")

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 537–544, 2009.
© Springer-Verlag Berlin Heidelberg 2009

are: to coordinate citizens, to keep the network efficiency, and to inform the central services.

The next section describes the requirements of the presented MAS application for crisis-management; and Section 3 describes the most relevant aspects of the other stages of the process for creating the practical application. Section 4 briefly presents some experimental results. Finally, Section 5 indicates the conclusions; and an annex with further details of the practical application can be downloaded from Grasia web[2].

2 Requirements

The requirements of the case study are formalized with use cases in the *MainUseCases* diagram (see Figure 1). The first use case is called *CoordinationForHelpingEachOtherUC*. The goal of this use case is the coordination among the people on the ground, so they can help each other. The people on the ground with medical capabilities offer help, and the poisoned people ask for help.

The second use case, which is called *InformCentralServicesUC*, informs the central services of the poison-affected locations. This makes it possible that the central services can assist areas with many affected locations.

The performance of the communication is crucial due to the urgency of the crisis situation; thus, a participant for managing the network is included in both use cases. The network-participant is the intermediator in the communications, and its goal is to improve the efficiency of the communication.

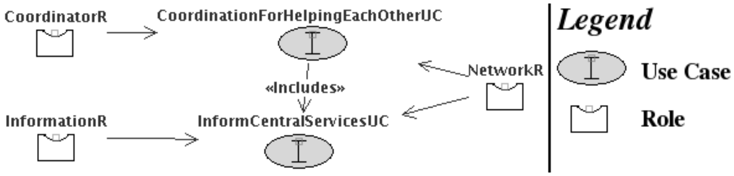


Fig. 1. Main use-cases diagram

3 The MAS for Managing the Crisis-Situation

The presented MAS satisfies the requirements mentioned at Section 2, with the following kinds of agents: *coordination-agents, network-agents* and *information-agent* . The *coordination-agents* are responsible for the coordination among the people on the ground. Each mobile device of the people on the ground is habited by a coordination-agent. Each user interacts with a coordination-agent. The *network-agents* are in charge of the proper communication among the other kinds of agents. The *information-agent* organizes the poison-affected locations and shows them to the central official services.

---

[2] http://grasia.fdi.ucm.es (in “Software” → “Additional Material for Papers”)

3.1 Main Design Decisions

Amongst other concepts, INGENIAS methodology uses *roles, goals, mental states* and *interactions*. The following sub-sections describes the most relevant design decisions related to these concepts.

Roles and Goals. The roles (see Figure 2) of the presented MAS are the following.

- *Coordinator Role* (i.e *CoordinatorR*). Several agents play this role, and each user can interact with one of them. The goal of the coordination-agents is to coordinate the people on the ground, so they can help each other; and they also can warn each other of the poison-affected locations.
- *Network Role* (i.e *NetworkR*). Several agents play this role. The network-agents are the intermediators for the communications in the presented MAS. The goals of this role is to isolate the possible communication problems and to improve the network efficiency.
- *Information Role* (i.e *InformationR*). Only one agent plays this role, and its goal is to inform of the poison-affected locations to the central services.

Mental States. The mental state of a coordination-agent contains both the location in the city and the state of the corresponding user. The user state can be any of the following values: *need-help, can-help, being-helped, helping*. The mental state of the information-agent contains a city map with the poison-affected locations. The network-agent mental states only contains the facts necessary for the communications. The initial mental states are shown at Figure 2.

Interactions. The user can both offer help, if the user has medical capabilities, and ask for help, if the user gets poisoned. In case of poisoning, the coordination-agent interacts with the network-agent with the *Coordinator-networker* interaction. In case of receiving a message of poisoning, the network-agent is responsible for three operations. Firstly, the network-agent must look for help around the poison-affected location, among other citizens, through the coordination-agents. Secondly, the network-agent must warn other citizens through the coordination-agents. Finally, the network-agent must report the poison-affected location to

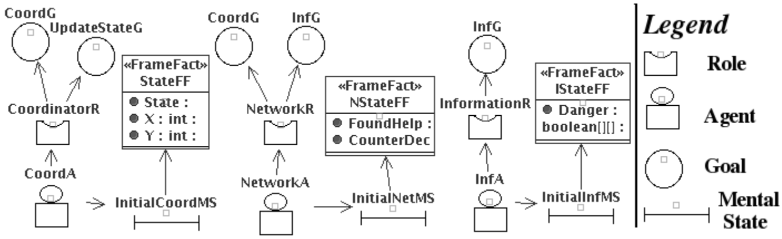


Fig. 2. Role and agent definitions of the presented MAS

the information-agent. The first and second operations are achieved with the *Networker-coordinator* interaction; whereas the third is achieved with the *Networker-informer* interaction.

In all the communications, a network-agent is the intermediate for isolating network problems. Therefore, the communication between people on the ground involves two interactions, a Coordinator-networker interaction, and a Networker-Coordination interaction. The communication between a person on the ground and the information central system involves two interactions, the Coordinator-networker interaction and the Networker-Information interaction.

In order to economize interactions, one communication among citizens satisfies two necessities: to ask for help around the affected location; and to warn other citizens of the affected location. In this application, a citizen only reports the necessity of help when being poisoned. This message is delivered to the coordinator-agents of the other citizens, and only the two following cases are possible: the receiver can help the requester and, in this case, the receiver coordination-agent ask its user to go to assist the requester citizen; otherwise, the receiver cannot help the poisoned person and, in this case, the receiver coordination-agent warns its user of the poison-affected location. Therefore, the same interaction are used for both warning and asking for help.

An alternative would be to use different interactions for each necessity. However, the current approach has a fewer number of interactions than the alternative and, consequently, increases the efficiency of the network.

In the search for help, there can be several people on the ground that can assist the same affected person. In this case, the network-agent only asks one of the coordination-agents to come, so the remaining people with medical capabilities can help other poisoned people. Thus, in the interaction protocol (see Figure 3), firstly, the networker asks for help to all the coordinators. Then, all the coordinators answer whether they can help or not. Finally, the networker only asks one coordinator to come.

An interaction is shared between the two necessary communications: among people in the ground; and between an affected person and the central information system. Both communications share the first interaction, called Coordinator-networker interaction, in which a coordinator-agent requests help to the

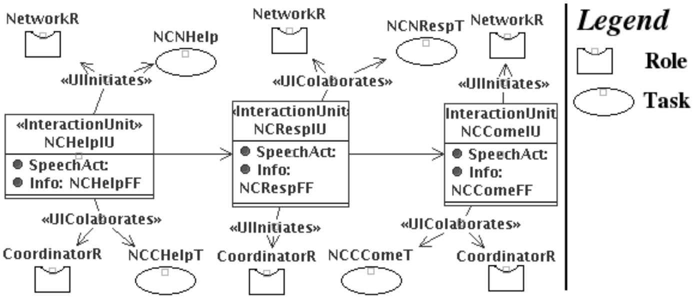


Fig. 3. Networker-coordinator protocol

network-agent. Then, the network-agent initiates two interactions, which respectively completes the two communications and are called Networker-coordinator and Networker-information. Since an interaction is shared, the number of interactions is fewer without changing the functionality, and the network efficiency increases.

Participants of the Interactions. The INGENIAS Agent Framework(IAF) adds certain participant for one-to-one and one-to-many interactions, by default. However, the selection of participants can be tuned for special needs.

For instance, a coordinator-agent establishes the interaction with one network-agent, when requesting for help. By default, the first network-agent is always selected in this one-to-one interaction. However, this default behavior makes the first network-agent get collapsed while it makes the other network-agents be isolated. For this reason, in this MAS, the same number of coordination-agents and network-agents are instantiated, and each coordination-agent is associated with a different network-agent for the communication. In case a coordination-agent cannot contact the associated network-agent, the coordination-agent randomly selects another network-agent. This policy increases the reliability and the efficiency (see Section 4) of the system because the first network-agent does not get collapsed as in the default behavior.

A network-agent can search for a coordination-agent that can help a requester coordination-agent. In this search, the network-agent initiates a one-to-many interaction. The default behavior is to add all the coordination-agents to the interaction. However, the default behavior is not the most appropriate for the following reason. The requester coordination-agent is also asked for help. The requester should not receive its own message, because it is not necessary and the unnecessary messages make the network efficiency lower. The selected policy for the Networker-coordinator interaction is the following: a network-agent searches for help for a requester coordination-agent, and the network-agent initiates a conversation with all the coordination-agents but the requester. In this manner, the unnecessary messages and unnecessary collaborators in the interactions are avoided. Thus, the efficiency (see Section 4) is increased above the default behavior.

Tasks. A group of tasks is the responsible for reacting to the events generated by the people on the ground. A person on the ground can generate several events from the *Graphical User Interface* (GUI) provided by the corresponding coordination-agent, and these events are *need-help*, *can-help*, *update-location* or *none*. For all the mentioned events, the corresponding task consumes the event, and updates the mental state of the coordination-agent. In particular, for the *need-help* event, the corresponding task launches a Coordinator-networker conversation to ask for help.

Another group of tasks is associated to the Coordinator-networker interaction and transfers the information of the requester (the agent ID[3], and its

---

[3] Agent identifier name, which is unique in the MAS.

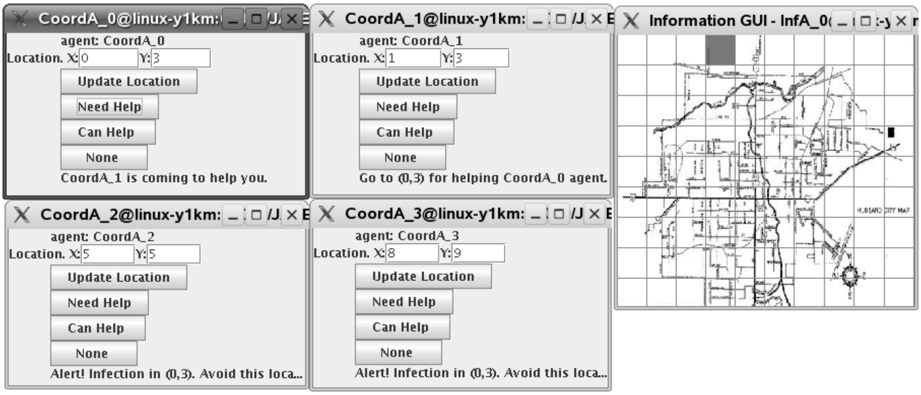
location). The most relevant task is the *CNNHelpT* task, which receives a request for help and is executed by the network-agent. This task launches two different conversations, whose goals are respectively to search for help from other coordination-agent and to inform the information-agent of the new affected location. In addition, within the coordination-network interaction, some tasks are the responsible for bringing back the response to the requester agent. The response indicates whether help is found. If help is found, the response contains the ID of the coordination-agent whose user is coming to help the poisoned user.

Another group of tasks is associated to the Networker-coordinator interaction, which is related to the search of help. Firstly, the network-agent delivers a frame fact, called *NCHelpFF*, requesting help to all the coordination-agents. This NCHelpFF frame fact contains the requester agent ID and its location. The *NCCHelpT* task is associated with the reception of the mentioned frame fact. This task checks out whether the corresponding user can heal the poisonous material effects, by consulting the coordination-agent mental state. Then, if the user has medical capabilities, the distance between the requester user and the helper user is calculated. If both users are close enough, the response is positive. The network-agent executes another task that manages all the responses and requests one user, at most, to come for helping.

Finally, some tasks are the responsible for the transfer of the new poison-affected location to the information-agent. One of these tasks updates the city map of the poison-affected locations in the information-agent mental state and shows the map with a GUI.

3.2 Implementation

The presented MAS implementation is available at IDK 2.8 distribution. A fully functional implementation skeleton is generated with IAF code generator and, then, the skeleton is manually completed with programming code for the content of tasks, the GUIs and other components.



agent: CoordA_0
Location. X: 0 Y: 3
CoordA_1 is coming to help you.

agent: CoordA_1
Location. X: 1 Y: 3
Go to (0,3) for helping CoordA_0 agent.

agent: CoordA_2
Location. X: 5 Y: 5
Alert! Infection in (0,3). Avoid this loca...

agent: CoordA_3
Location. X: 8 Y: 9
Alert! Infection in (0,3). Avoid this loca...

Fig. 4. Example of execution

Figure 4 shows an example of execution, in which, four people on the ground use the presented MAS. The CoordA_1 and CoordA_3 users have medical capabilities; thus, they press the *Can Help* button. The CoordA_0 user gets affected by the poisonous material; thus, he presses the *Need Help* button. The MAS starts to coordinate and inform the central system. CoordA_1 can help CoordA_0 because of two conditions: CoordA_1 has medical capabilities and is close enough to CoordA_0 (look at the locations in Figure 4). On the contrary, no other coordination-agent satisfies the same conditions. The response is sent back to CoordA_0 with the following message, "CoordA_1 is coming to help you". The other people on the ground are warned of the new poison-affected location with the following message, "Alert! Infection in (0,3). Avoid this location". Finally, the city map of the central services is updated. The new affected location (0,3) is indicated in the information GUI with a different background color.

3.3 Testing

The presented MAS includes a battery of tests (see the Annex in *Grasia* web), which are modeled with INGENIAS methodology. The first test, which is called *GeneralTest*, checks out several functionalities by initializing the MAS with the same values of the execution example (see Figure 4). In this case, a citizen requests for help, and there are other three citizens. One of them is too far for helping the requester. The second one has not medical capabilities, and the third one is close enough and has medical capabilities. This test checks that a citizen is asked to come for helping when the citizen satisfy two conditions: the citizen must be close enough to the requester; the citizen must have medical capabilities. Moreover, the GeneralTest also checks out the information of the central services, by checking the poison-affected locations of the map at the end of the execution.

The second test, which is called *OnlyOneComesTest*, is initialized with several people on the ground with medical capabilities (a deployment called *SeveralDoctors*). In this test, an agent asks for help, and there are several citizens with medical capabilities nearby. Thus, there are several citizen that satisfy the necessary conditions for helping the requester. The tests checks that only one citizen is asked to go for healing the requester.

4 Experimental Results

The system was run in a JADE platform with several deployments (i.e. with *nine* agents and *twenty-one* agents), and several selection-mechanisms of participants (i.e. *default selection* and *presented selection*). In these experiments, all

Table 1. Latency-times in mili-seconds

	Default Selection	**Presented Selection**
9 agents	2069 ms	1928 ms
21 agents	13890 ms	5354 ms

the users simultaneously either offered help or asked for help, several times. This experiment measured the latency between the moment a user asks for help and the moment the system replies after the coordination process. The average of these latency-times is presented in Table 1. As one can observe, the presented selection strategy for participants increases the performance. This increment of performance is higher when more agents are running (for instance, latencies of 13,9 seconds and 5,4 seconds respectively for default selection and presented selection, when running 21 agents).

5 Conclusions and Future Work

On the whole, a MAS is developed with the IDK tool from the specification to the testing for managing crisis-situations in a city, in order to illustrate the use of the IDK tool. Practitioners can download the IDK 2.8 distribution to experiment with both the IDK tool and the presented practical application.

A future direction is to adapt the presented MAS solution to other crisis-management scenarios, such as the scenarios considered in *RoboCup Rescue* [3].

Acknowledgements. This work is supported by the project "Agent-based Modelling and Simulation of Complex Social Systems (SiCoSSys)", funded by Spanish Council for Science and Innovation, with grant TIN2008-06464-C03-01.

References

1. AgentLink. Multi-agent Systems in Crisis Management: The Combined Systems. Case Study (December 2005)
2. Gómez-Sanz, J.J., Fuentes-Fernández, R., Pavón, J., García-Magariño, I.: INGENIAS Development Kit: a visual multi-agent system development environment (Awarded as Best Academic Demo of AAMAS 2008). In: The Seventh International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, Estoril Portuga, May 12-16, 2008, pp. 1675–1676. ACM, New York (2008)
3. Kitano, H.: Robocup rescue: A grand challenge for multi-agent systems. In: Proceedings of Fourth International Conference on MultiAgent Systems, 2000, pp. 5–12 (2000)

The Delphi Process Applied to African Traditional Medicine

Ghislain Atemezing, Iván García-Magariño, and Juan Pavón

Dept. Software Engineering and Artificial Intelligence
Facultad de Informática
Universidad Complutense de Madrid, Spain
atemezing@yahoo.com, ivan_gmg@fdi.ucm.es, jpavon@fdi.ucm.es

Abstract. The African Traditional Medicine (ATM) has been applied over the years in the ethnics of the African continent. The experts of this area have conserved this knowledge by oral transmission. This knowledge varies from one ethnic to other, thus it is distributed among these ethnics. In this context, this work proposes to use a multi-agent system to manage this distributed knowledge. The presented approach uses the Delphi process, in which, there are several agents that represent ATM healers and participate in several rounds of questionnaires in order to reach a consensus for providing a heal to a patient.

Keywords: Traditional Medicine, Case-Based Reasoning,language, multi-agent system, Delphi Process.

1 Introduction

African Traditional Medicine (ATM) [14] is the result of diverse experience, mixing customs and knowledge about nature, which has been transmitted by oral tradition along the history. Nowadays, the availability of computers and networks in an increasing number of places around the African continent opens the possibility to consider the support of knowledge systems for new practitioners, who can take benefit of ATM knowledge. ATM uses many symbols and rites in its way to cure. It distinguishes symptoms from illness. When a patient goes to a traditional-practitioner, he suffers from something (an illness) that has a name in the set of the diseases known in modern medicine. But the traditional-practitioner seems to deal with a "social unrest" expressed by the tensions (revealed or not) that could exist in the patient's environment. Also, this medicine faces a wide range of factors, causes of diseases and medicinal plants. However, this knowledge did not have until now, a formal codification for a better understanding and management of concepts and the relations between them. The work in [2] presents an ontology that best feeds with the ATM domain with the particularity of being extendible to other types of traditional medicine.

This work gathers the ATM knowledge in order to provide a prescription for a patient, by means of a Multi-agent System (MAS) that uses Delphi process. In this MAS, some agents represent ATM healers and each of these agents have

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 545–552, 2009.
© Springer-Verlag Berlin Heidelberg 2009

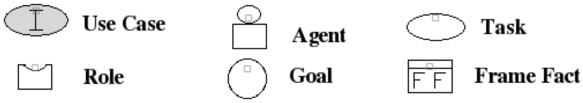


Fig. 1. The Most Relevant INGENIAS Notation

a piece of ATM knowledge. This MAS uses the Delphi [5] process to reach consensus among agents and provide a prescription to the patient. In the Delphi process, a monitor creates several rounds of questionnaires, in which the ATM healers fill the questionnaires, and questionnaires include some information of the previous rounds. In these manner, the system usually evolves to reach a consensus. Before this work, the Delphi process obtained good results for a MAS [10] that evaluates document relevance. This work chooses the INGENIAS [17] methodology for the development and its tool support, the INGENIAS Development Kit (IDK) [11], because they can generate fully functional MASs from specifications. The notation of the most relevant concepts of this methodology is presented in Figure 1.

The next section describes the background concepts, which are the ATM and the Delphi Method. Section 3 describes the application of the Delphi process to the ATM, by means of a MAS. Finally, Section 4 mentions the conclusions and future work.

2 Background Concepts

2.1 African Traditional Medicine

African Traditional Medicine (ATM) [14] is a system of cure in which disease is considered as a social illness, which is necessary to eradicate from the root. There is intervention of several actors from several domains, which turns complex and diversified exchanges and treated knowledge. Several actors can be identified in ATM, with specific roles and functions: the *healer*, who knows the names of plants, animals and rocks; the *fetishist*, who predicts important events and is consulted to find the cause of a disease; the *soothsayer*, who predicts and is seeing as the intermediary with the divinity; and the *magician*, who throws lots and makes use of black arts.

In the treatment process , a healer tries to reconcile the patient in all its integrity, as well physical as psychic, by using symbols that are a part of the universe and the life of every day of the patient during its interventions. In ATM, it is also necessary to distinguish symptoms from disease. When a patient meets a traditional doctor, he suffers from the evil of which one can attribute a name in human disease of the modern medicine. But for the traditional doctor, this patient is seen as a person who possesses a symptom, a sign of a social illness. Social illness expresses tensions (hidden or revealed) that could exist in the circle of acquaintances of the patient. Some anthropologists, such as [18], introduce the concept of traditional model to express all that is lived in the traditional vision.

The body in this model consists of two entities: a visible part and an invisible part. The global steps in the traditional model are : *interpretation* of the cause of the bad physical appearance (sort of diagnosis); phase of *divination* to know how to treat the patient; and *prescriptions* according to the cause of the disease and follow-up of the patient evolution in the process of cure depending of the case [1].

2.2 Delphi Method

The Delphi method [8] plans several rounds of questionnaires which are sent to the different involved experts. The results collected can be included partially in a new round of questionnaires, in order to re-orient the initial problem.

The Delphi approach has been applied in several areas: Roth [19] used the Delphi approach for acquiring knowledge from multiple experts. Recently, Bryant [6] applied the Delphi method for estimating the risk factors of the terrestrial chemical spill; Hayes [12] did a Delphi study of the future of marketing of the higher education; and Miro [16] applied the Delphi method to reach consensus among professionals with interest in chronic pain among children and adolescents.

The automatization of Delphi [21] was considered first as a set of computers and software assisting human experts in the process. Moreover, Holsapple [13] describes some possibilities to integrate software systems in a Delphi organization. However, the Delphi method was not completely computerized until our previous work [9], which concerns document relevance.

3 Delphi Process for ATM

The Delphi method can make a MAS reach a consensus [9]. The current work applies this method in the African traditional medicine. Within this section, Subsection 3.1 describes the general overview; and Sub-sections 3.2 and 3.3 respectively indicate how the questionnaires are replied and managed.

3.1 General Overview

In this application of Delphi process, the experts are the ATM healers, and they are modeled with a collection of cases. In this approach, the main use cases (see Figure 2) are the following:

- *beCuredUC*: The MAS indicate the way a patient can be cured.
- *delphiUC*: In this use case, the agents, representing the healers of ATM, cooperate to obtain the best cure for a patient.

The MAS uses the following agents for this application of Delphi process:

- *Patient* agent: This agent is the interface for the patient. The patient indicates its illness to this agent. This agent asks a *Monitor* agent to obtain a cure for its patient.

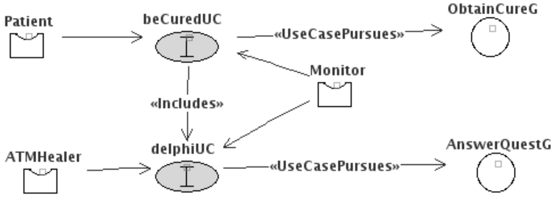


Fig. 2. Main use cases considered in the development of the Delphi process for ATM

- *ATMHealer* agent: Each of these agents has a particular piece of knowledge of the ATM. It can provide a prescription for each illness. This agent can also be flexible and consider other opinions and slightly alter its original prescription.
- *Monitor* agent: This agent can organize a Delphi process, in which it makes a group of different ATMHealer agents to reach a consensus.

The workflow (see Figure 3) of this ATM MAS follows the Delphi process. Firstly, in task *FillSymptoms*, the patient fills a form with its symptoms and the Patient agent inquires a prescription to the *Monitor* agent. Then, the form is received by the *Monitor* agent and a customized questionnaire is elaborated with task *InitQuestT*. The questionnaire is answered by ATMHealer agents by means of a task *AnswerQuestT*. The answer is processed by the *Monitor* with a task *ProcessAnswerT*. As a result of this task, another round can be derived or not. If a new round occurs, the task *CreateOtherQuestT* should be executed. This would force another elaboration of questionnaires and a new answer delivered by experts. If no more rounds occur, then the *Monitor* delivers the result to the *Patient* agent, which processes the prescription with task *ResultObtainedT* and presents it to the patient.

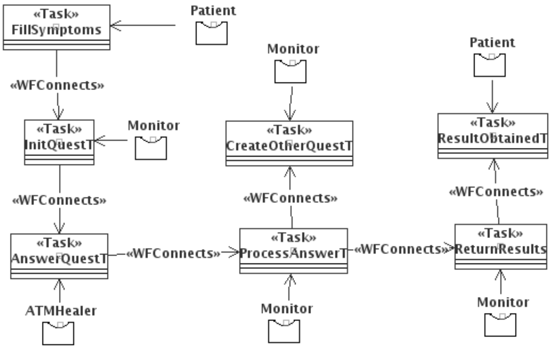


Fig. 3. Overview of the workflow used to implement the Delphi process for ATM

3.2 Replying the Questionnaires

The ATMHealer agents reply the questionnaires. In the first round, each ATMHealer agent receives input from the symptoms and indicates a prescription. Each ATM healer agent has a piece of knowledge, which is represented with a collection of African-medicine cases. These ATMHealer agents use the *Case Base Reasoning* (CBR) [22] to reply questionnaires. A case here is constituted with: the *symptoms, the illness, the healer experience of the case* and the *prescription.*

The symptoms consist of the following attributes: *Family (polygamic, monogamic, large); weight (constant or no); fever(yes or no); vomiting (frequently, si or no); human part (one of the sixteen division we have adopted from [15]); pain type (serious or no), patient's category (child, adolescent, adult and old); and sex (male or female)..*

The names of the illness in ATM changes from one region to another region. Thus, a brief ontology is necessary to narrow the diversity of terminology. The ATM healer experience indicates the reliability of the case. Both the illness and experience are taken into account for considering the answers from other ATM healers.

The prescription consists of the following attributes:

- *Operatory mode*: that describes how the remedy is obtain or the technics used to obtain the remedy.
- *Posology*: which describes how to take the remedy.
- *Medicine Group*: which is one of the fourteen medicine groups which, among others, can be respectively: drinked, breathed, purge, ointed, gargled and licked.
- *Requirements*: which are some previous conditions to make use of plants or organic minerals in a composition or/and posology.
- *Compositions*: which are plants names or minerals that enter in the composition of the potion or remedy.

For replying the questionnaires, the symptoms are used to retrieve the most similar case. The *similarity* function uses the cosine of the similarity for each attribute. The similarity of the most symptoms attributes is calculated by means of the ATM ontology and some table of values that indicates the similarity between two values. These ontology and similarity values are the results of a collaborative work with ten human ATM healers, who currently lives in Cameroon (a Central African country). The questionnaires are filled with both the illness and the prescription of the retrieved case.

Since the agent should save and learn from cases, this work uses the *JColibri* [4] CBR tool. Each agent has casebase in a plain text with a mapping connector to the CBR, where the columns are the relevant attributes of the forms questionnaires filled by the patient agent. The use of the CBR tool allows one to make use of the similarities functions already existed and implemented within the tool.

3.3 Managing the Replies for Reaching Consensus

The *Monitor agent* creates questionnaires considering the Replies of the previous Rounds. Then, ATM healer agents consider the replies of other ATM healer agents when these questionnaires are replied. This consideration is based on the *illness* and the *expertise* retrieved for the collections of cases.

Each Agent has his experience of treatment, that is all the cases of his different practices. This work uses the *Reputation Score* (see its definition in Equation 1) to balance this experience for each disease case (E_C). The ATMHealer agents communicates this reputation score, which is within the [0,1] interval, in order to justify its point of view.

$$R_S = \frac{\alpha * R_M + \beta * E_C}{\alpha + \beta} \tag{1}$$

where
- R_M is the Reputation of the *Master* [1] of the healer who is the source of the ATMHealer agent.
- E_C is the Experience in the Case of the human-healer who is the source of the ATMHealer agent.
- R_M and E_C are within the [0,1] interval.
- α, β are constant parameter values.

In the rounds of this process, different situations could be raised, which the Monitor agent manages with the following rules:

Rule 1. *if there are more than one agent with the same prescription and a high reputation score, the solution of these agents is adopted for the case.*

Rule 2. *if there is one agent with a reputation score greater than 0.9 and all the others with a score less than 0.5, the solution is the one proposed by that agent.*

Rule 3. *Otherwise, the Monitor starts a new round of questionnaires. The ATM healer agents are expected to smoothly change their replies. For this reason, the agents that strongly change their reply are considered as non-reliable, and they are omitted in the next round.*

Rule 4. *If there is no one who knows the case, the case must be treated by a human ATM healer. Each ATM healer agent saves the case in his KB and learn from it.*

3.4 CBR Integration Approach

During the last years, many successful approaches of integrating CBR to MAS architecture have been proposed for health system [7],[3]. In many occasions, CBR are modeled as services linked to agents different from a traditional point

---

[1] The *master* is the person from whom the healer has learnt. The reputation of the master is obtained by asking the patients of the master.

of view where the CBR was part of the agents; with the consequence of increasing the system's overall performance[20]. However, we agree with that posture, but in our system, we integrate the CBR into the ATMHealer agents just because it makes use of the first two parts of the traditional four phases of a CBR cycle; that means the retrieve and reuse phases. We are convinced that there is no need in this case to model CBR here as services.

4 Conclusions and Future Work

In conclusion, this work proposes a MAS that combines the Delphi process with the Case Base Reasoning (CBR) for African Traditional Medicine (ATM). The Delphi method regards on the domain, and it was necessary that several experts of ATM provide the necessary information to the engineers. The proposal is under validation with a prototype. For future work, the presented MAS can be tested with patients, in order to validate the system with further empirical results. In addition, this work pursues another future goal, which is to prove that the Delphi method is a reliable mechanism to reach consensus in MAS. The first step was to apply the Delphi method in document relevance. In this second work, the Delphi method is applied in African traditional medicine.

Acknowledgments. We acknowledge support from the project "Agent-based Modelling and Simulation of Complex Social Systems (SiCoSSys)", supported by Spanish Council for Science and Innovation, with grant TIN2008-06464-C03-01

References

1. Atemezing, G.: Modisation multi agent massif d'un syste d'aidela dision en miecine traditionnelle. Annales de la Facultides Arts, Lettres et Sciences Humaines del'Universitde Ngaound, pp. 199–217 (2006) ISSN :1026-3225
2. Atemezing, G., Pavon, J.: An Ontology for African Traditional Medicine. In: International Symposium on Distributed Computing and Artificial Intelligence 2008 (Dcaia 2008), p. 329. Springer, Heidelberg (2008)
3. De Paz, Y., Bajo, J., De Paz, J.F., Corchado, J.M.: Integrating Case Planning and RPTW Neuronal Networks to Construct an Intelligent Environment for Health Care. Expert Systems with Applications 36, 5844–5858 (2009)
4. Bello-Tomas, J.J., Gonzalez-Calero, P.A., Diaz-Agudo, B.: JColibri: An Object-Oriented Framework for Building CBR Systems. Lecture Note. In: Computer Science, pp. 32–46 (2004)
5. Brown, B.B., Rand Corporation: Delphi Process: A Methodology Used for the Elicitation of Opinions of Experts. Rand Corp. (1968)
6. Bryant, D.L., Abkowitz, M.D.: Estimation of terrestrial chemical spill risk factors using a modified delphi approach. Journal of Environmental Management 85, 112–120 (2007)
7. de Paz, Y., Dante, T., Corchado, J.M., Bajo, J.: Intelligent Environment for Monotoring Alzheimer Patients, Agent Technology for Health Care. Decision Support Systems 44(2), 2382–2396 (2008)

8. Cuhls, K.: Delphi method. Technical report, Fraunhofer Institute for Systems and Innovation Research (2003)
9. García-Magariño, I., Gómez-Sanz, J.J., Pérez-Agüera, J.R.: A complete-computerised delphi process with a multi-agent system. In: Sixth international Workshop on Programming Multi-Agent Systems, ProMAS 2008, Estoril Portugal, May 13, 2008, pp. 187–202 (2008)
10. García-Magariño, I., Gómez-Sanz, J.J., Pérez-Agüera, J.R.: A multi-agent based implementation of a delphi process. In: The Seventh International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, Estoril Portugal, May 12-16, 2008, pp. 1543–1546.
11. Gómez-Sanz, J.J., Fuentes-Fernández, R., Pavón, J., García-Magariño, I.: INGENIAS Development Kit: a visual multi-agent system development environment (Awarded as Best Academic Demo of AAMAS 2008). In: The Seventh International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, Estoril Portugal, May 12-16, 2008, pp. 1675–1676 (2008)
12. Hayes, T.: Delphi study of the future of marketing of higher education. Journal of Business Research 60, 927–931 (2007)
13. Holsapple, C.W., Joshi, K.D.: Knowledge manipulation activities: results of a delphi study. Information and Management 39, 477–490 (2002)
14. Sindzingre, N., Dozon, J.P.: Pluralisme thérapeutique et médecine traditionnelle en afrique contemporaire; Fonds Documentaire ORSTOM, 43–51 (1986)
15. Kingue Kwa, N., et al.: Les cahiers du male ma makom. Duala (1994)
16. Mir, J., Huguet, A., Nieto, R.: Predictive factors of chronic pediatric pain and disability: A delphi poll. The Journal of Pain 8(10), 774–792 (2007)
17. Pavón, J., Gómez-Sanz, J.J.: Agent Oriented Software Engineering with INGENIAS. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) CEEMAS 2003. LNCS, vol. 2691, pp. 394–403. Springer, Heidelberg (2003)
18. Rosny, E.: L'Afrique des guérisons. Karthala, Paris (1992)
19. Roth, R.M.: A Delphi approach to acquiring knowledge from single and multiple experts. In: Proceedings of the 1990 ACM SIGBDP conference on Trends and directions in expert systems, pp. 301–324 (1990)
20. Rodriguez, S., Bajo, J., Corchado, J.M., Dante, T., De Paz, J.F.: Developing a Service Oriented Alternative for Distributed Mullti-Agent Systems. In: ZOCO 2008 / JISBD, pp. 277–285 (2008)
21. Turoff, M., Hiltz, S.R.: Computer Based Delphi Processes. Gazing into the Oracle. In: The Delphi Method and its Application to Social Policy and Public Health, pp. 56–85. Jessica Kingsley Publishers, London (1996)
22. Watson, I., Marir, F.: Case-Based Reasoning: A Review. The Knowledge Engineering Review 9(4), 327–354 (1994)

Composing and Ensuring Time-Bounded Agent Services

Martí Navarro, Elena del Val, Miguel Rebollo, and Vicente Julián

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia (Spain)
{mnavarro,edelval,mrebollo,vinglada}@dsic.upv.es

Abstract. There are situations where an agent needs to compose several services together to achieve its goals. Moreover, if these goals should be fulfilled before a deadline, the problem of service composition become more complex. In this paper a multi-agent framework is presented to deal with service composition considering service execution time taking into account the availability and the workload of the agent that offers the service.

1 Introduction

Multi-Agent Systems technology makes possible to cover a broad area of problems. Typical problems are systems in which there are entities (Requesters) which may require one or more services from other different entities (Providers). As an example, in the area of manufacturing systems, Requesters would be clients with new work orders and Bidders would be factories or factory clusters which provide the needed resources to obtain the required product. Obviously, the development of these types of systems is complex and, therefore, it is necessary to analyze in detail the intrinsic characteristics of these application environments.

One of the main problems in this kind of systems is how to create added-value services dynamically by composing elemental services. Services can be seen as elemental components and they are commonly used by human developers to create bigger systems. Semantic annotations help machines to deal with services, but service discovery and composition are complex tasks that need extra intelligence doses to achieve proper results, specially in open and dynamic environments where services are not always available.

Another important problem to solve in this kind of systems is to attend requesters requirements when they ask for having the answer available in a bounded time. The late fulfillment of some services could reduce the quality of the response offered by the multi-agent system. Therefore, before performing a complex service, someone must analyze if the service can be provided on time.

This paper presents a multi-agent framework which allows to compose time-bounded services and to ensure their fulfillment. The multi-agent framework considers the current workload and the availability of the provider agents of

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 553–560, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the services which can be atomic or complex ones formed by a composition of elemental services.

The paper is structured as follows: In section 2, a general description of the framework is presented. Section 3 presents how services are adapted to be composed using planning techniques. Furthermore, a temporal extension for OWL-S descriptions are shown. After that, in section 4, the agent in charge of consulting the execution time of the services is described. Section 5 presents Real-Time Agents, which execute time-bounded services. Finally, conclusions and final remarks are presented in section 6.

2 SAES Framework

SAES (Search And Execution of Services) is an multi-agent framework which allows a client to search complex services that satisfy its requirements and to guarantee their execution before a maximum time (deadline) established by the client. The framework is composed of three types by agents,

1. **Service Composer Agent:** The Service Composer Agent (SCA) has as objective getting service compositions which fulfill a set of constrains.
2. **Commitment Manager Agent:** The Commitment Manager Agent (CMA) checks if the service composition is feasible and it can be executed on time. Moreover, this agent monitors the service execution in order to take it into account the observed performance for future similar situations.
3. **Real-Time Agents:** In our system, a Real-Time Agent (RTA) offers temporal-bounded services. These agents are trained to provide services whose execution needs to be controlled temporarily.

Internally, the SAES works as follow: (i) Initially a client sends a query to determine if there is a set of services which could satisfy a goal before a deadline established by the client. (ii) The request is received by the SCA, which starts the service search process. The aim of this process is to find individual services or compositions are fulfill clients goal taking into account the temporal restrictions established by the client. (iii) When a service composition is obtained, the SCA sends it to the CMA. (iv) After that, the CMA queries each RTA involved in the service composition asking for the agent's availability to provide the service on time. (v) Each RTA analyzes the service execution time according to its current schedule and returns it to the CMA. (vi) The CMA establishes pre-commitments with the involved RTAs to reserve the slack for the estimated service during a period of time. The CMA sends the resulting service to the client with a success probability. (vii) If the client agrees with the service composition the CMA confirms commitments with the involved RTA. Otherwise, the client sends a message rejecting the service and the CMA breaks the established pre-commitments. (viii) Once the service execution starts, the CMA monitors the fulfillment of the committed execution time. When the service ends its execution, the CMA stores the execution time to take it into account in future queries. In the case that a RTA does not fulfill its commitment, it will be penalized.

3 Service Composer Agent

If the client requirement cannot be solve only with one service, then a possible solution is to compose automatically several services in a sequence. However, dynamic service composition is a complex problem and it is not entirely clear which techniques serve service composition best. There are several proposals that use techniques such as hypergraphs [2][7], modelchecking [6][11] or planning [10][4]to deal with this problem.

Service composition as planning describes a service as a process in terms of inputs, outputs, preconditions and effects. Using the metaphor of an action, composition can be viewed as a planning problem. An important benefit of the planning approach is the exploitation of knowledge that has been accumulated over years of research on the field of planning. Therefore, well know planning algorithms, techniques and tools can be used to advantage of efficient and seamless service composition.

In SAES framework, the Service Composer Agent (SCA) provides a service composition (or a set of service compositions) which satisfies the goal and the deadline established by the client. To do that, the SCA considers the execution time of the services and uses AI planning techniques to automate this process. The SCA is formed by three components:

- **Service Translator:** It is responsible for translating OWL-S service descriptions extended with a duration non-functional parameter into PDDL 2.1 durative actions [5].
- **Problem Translator:** It is responsible for translating the client query into a PDDL 2.1 problem description.
- **Composer:** It takes as input a PDDL 2.1 problem and provides a set of plans which represent service compositions that satisfy the client goals and temporal constraints.

3.1 Temporal Extension in OWL-S Service Descriptions

Agent architectures use the service abstraction to describe its functionality, but service descriptions are somehow limited (for example, DF descriptions in FIPA-like architectures). A more expressive language has to be used to describe services so agents can compose and provide added-value services.

A common drawback to all standard service description languages is the impossibility of modeling how time pass in the system. Some services could be completely useless if they are not provided on time. Or providers could predict future bottlenecks and solve them with a proper scheduling of the pending tasks. For doing that, time is an important factor for services and might be taken into account.

The execution time of a service offered by an agent should be expressed in its service description. Service descriptions in SAES architecture are expressed using OWL-S, which has been extended to include temporal information. This extension consists on a new non-functional parameter which represents service time execution. Besides that, preconditions and effects are temporarily annotated.

Duration. Service duration is represented by a non-functional parameter called duration which represents the service time execution. This parameter is a PDDXML expression that contains a value assignment to the variable duration. PDDXML [8] is a XML dialect of PDDL that simplifies parsing, reading, and communication PDDL descriptions using SOAP.

Preconditions. The annotation of a precondition makes explicit whether the associated proposition must hold: (i) *at the start* of the interval (the point at which the service is applied), (ii) *at the end* of the interval (the point at which the final effects of the service are asserted), (iii) *over* the interval from the start to the end (invariant over the duration of the service).

The preconditions in the OWL-S document are also PDDXML expressions.

Effects. The annotation of an effect makes explicit whether the effect is immediate (it happens at the beginning of the interval) or delayed (it happens at the end of the interval). No other time points are accessible, so all discrete activity takes place at the identified start and end points of the service. The service effects are PDDXML expressions in the OWL-S document.

3.2 Composing Temporal Services

Basically, the SCA works as follows. When a service is registered in the system, the SCA gets the OWL-S service description and sends it to the Service Translator. The Service Translator (ST) is responsible for translating OWL-S descriptions into PDDL 2.1 durative actions. Once the SCA has the services modeled as actions, it generates a PDDL domain file which contains the definition of the actions structured as a planning problem. This process is an extension of a converter presented by Klusch and Gerber [8] which is limited to deal with services that are not temporally annotated.

When a query arrives at the SCA the composition process starts. The client query is an OWL document that contains the information related with the inputs of the desired service and the goals (outputs) to be achieved. This file is sent to the Problem Translator (PT) to be translated. into a valid initial state which is stored in a PDDL problem file.

After that, the domain and problem files in PDDL 2.1 are sent to the Composer. The Composer can include any planner which deals with PDDL 2.1 language and considers time as a parameter to optimize the plans. This planner obtains a plan composed of actions (services) with their initially estimated duration and the total estimated time of the plan. The plan represents a service composition sequence that satisfies the goal considering temporal annotations. The SCA will continue searching plans until the CMA sends to the SCA a message to finish the search process because a previous plan has been accepted.

4 Commitment Manager Agent

The CMA has three main functions: (i) To check if the set of services offered as a solution by the SCA will be available to fulfill the request. (ii) To establish a commitment relationship with the RTAs that offer the services (associated to the services selected for the composition). (iii) A RTA controls that the services will be executed correctly (achieve its commitments).

To fulfill the first function, the CMA must communicate with all RTAs that offer the services involved in the composed service. Each RTA analyzes when it can complete the service for the CMA and it returns the result to CMA. The result consists in a tuple $< T_{start}, T_{duration}, SP >$ where T_{start} indicates the moment when the service can begin its execution, $T_{duration}$ indicates the necessary time to complete the service and SP is the probability of a successful execution. Moreover, a pre-commitment between the RTA and the CMA is established.

When all RTAs have responded to the CMA, it must calculate the success probability associated to the whole service composition. For doing that, the CMA uses the success probability sent for all RTAs weighted with the information of previous executions of similar services. The service composition success probability is calculated as follows:

$$SP_{composition} = \prod_{i=0}^{N} SP_i * \omega_i \tag{1}$$

Where $\omega_i \in [0,1]$ is the weight associated to the service i. This weight is related to the previously fulfilled commitments; A RTA who has many unfulfilled commitments will have a low weight.

Once the CMA calculates the service composition success probability, it sends to the client the composed service and its probability $SP_{composition}$. The client analyzes the result and determines if the composition is suitable for it. If the client is agree with the composition, it communicates to CMA that can start the execution of all services. In that case, the CMA confirms the pre-commitments for a formal commitment with the RTAs that offer the services. If the client is not agree with the service composition, the CMA breaks the pre-commmitments freeing the slack reserved by the RTAs.

The CMA is also in charge of ensuring that the acquired commitments are fulfilled. In the case that a commitment cannot be fulfilled, the CMA penalizes the RTA who provides the service. This penalty is captured through the weights applied when the CMA calculates the service composition success probability.

5 Real-Time Agent

The RTA that offers a service is in charge of deciding if it can commit to execute the service without exceeding the maximum time proposed by the client.

To determine whether a service can be executed on time, it is necessary to know the execution time for each service. In some cases, the execution time of the service is known and limited. In these cases, to determine the necessary tasks to fulfill the service and the maximum time needed to perform it is relatively easy using well-known schedulability techniques [9][3].

There are services for which to calculate the needed execution time is not possible. In this type of services, a time estimation is the unique measure that can be made. In order to do this estimation, each agent that offers a service incorporates a module, called Temporal Constraint Manager (TCM).

The Temporal Constraint Manager must decide if a RTA can commit itself to perform a specific service. A possible way of performing such decision-making functionality is to use a Case-Based Reasoning (CBR) approach, which adapt previous problem solving cases to cope with current similar problems [1]. Therefore, in the absence of unpredictable circumstances, we assume that an agent can commit itself to perform a service within certain time if it has already succeeded in doing so in a similar situation. To carry out the decision-making about contracting or not a commitment, the TCM has been enhanced with a RT-CBR, following a soft Real-Time approach. This RT-CBR estimates the time that the performance of a service could entail by using the time spent in performing similar services.

The classical CBR cycle consists of four steps: *Retrieve*, *Reuse*, *Revise* and *Retain*. *Retrieve* similar experiences from a *case-base*, *reuse* the knowledge acquired in them, *revise* such knowledge to fit the current situation and, finally, *retain* the knowledge learnt from this problem-solving process. In our framework, the CBR phases must observe soft real-time constraints and thus, its execution time must be bounded. Otherwise, the RT-CBR could provide the TCM with useless time estimations about services whose deadline have already expired. In this paper is presented a generalization of the RT-CBR [12]. Each agent that offers a service must have a specific implementation of its RT-CBR taking into account its application domain.

To bound the response time of the TCM, the RT-CBR case-base must have an structure that eases the case retrieval. Anyway, independently of the choice made about the indexation, the temporal cost of most retrieval (and reuse) algorithms depend on size of the case-base. This entails to specify a maximum number of cases that can be stored in the case-base and to perform a constant maintenance and updating of the information stored.

Figure 1a shows the execution phases of the TCM. The module is launched when the agent begins its execution. At the beginning, the TCM controls if a new service request has arrived (Figure 1b). If the new request is a service request where the service execution time is not known, the TCM must estimate the time required to execute that service. It is necessary to determine if the service can be

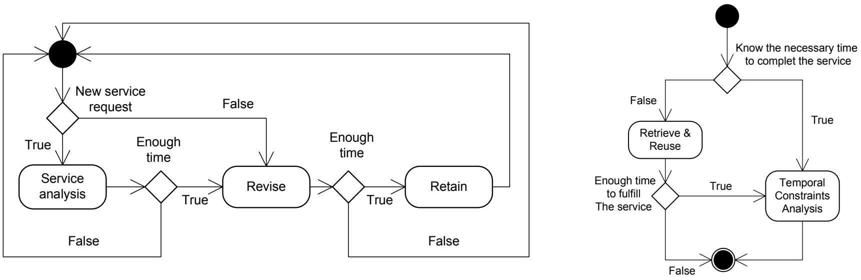


Fig. 1. a) Temporal Constraints Manager Algorithm. b) Service Analysis state.

completed before the deadline specified in the request. When the estimated time is obtained and the agent confirms that it is possible to execute the service, the necessary tasks to perform the service are analyzed at low-level using a real-time scheduler. The worst-case execution time of each phase of the TCM is known and, therefore, the phases are temporally bounded. This feature is crucial to allow the TCM to have a precise time control of each execution phase. As it can be seen in Figure 1a, the TCM execution is cyclical. When there is no request, the manager can use its idle time to perform the revision and retention phases.

The cases of the RT-CBR module have the following general structure:

$$C =< S, \{T\} > \tag{2}$$

where S represents the features that characterize a service that the RTA performed in the past and T is the time that the RTA spent in the execution of that previous service. In addition, since the same task could be executed several times with different durations, this value T can also be a serie of temporal values. Therefore, the RT-CBR estimates the duration of new services by means of a function $t : T \rightarrow f(T)$ computed over the temporal values that last similar previous services. The expected time T_s to perform a service that consists of a set of tasks is the aggregation of the estimated time for each of its tasks:

$$T_s = \sum_{i=0}^{I} t_i \tag{3}$$

Finally, the series of observed execution times also allows the RT-CBR module to estimate a *success probability* $P(T_s)$ for a service to be performed within an specified time. This is an interesting data for RTAs, which could use this probability to make strategic decisions about their potential commitments. Setting a *confidence factor* (CF) that represents a minimum threshold for the success probability, RTAs would commit themselves to fulfilling a service if

$$\exists T_s / P(T_s) \geq CF \wedge Ts \leq deadline \tag{4}$$

Thus, RTAs with more risky strategies could undertake commitments with lower confidence values than more cautious agents. Once an RTA determines if it can commit itself to fulfill the service, the agent will sent a message to the CMA with the necessary information. This information is: the moment when the agent starts the service execution, the duration of the service execution, and the success probability. The CMA can calculate if the service composition is feasible when all RTAs had answered.

6 Conclusions

In service composition, it is important to consider service time execution, mainly in environments where there are temporal constraints. This time in many situations is established by the providers and could not be realistic due to it does not consider workload or service availability in that moment. It is necessary to contact with provider agents and to query their availability and workload at that moment. With this information, a more realistic estimation can be obtained.

In this paper, a multi-agent system framework to deal with this problem has been presented. In this framework, a planning technique is used to get a service composition considering service execution time. To refine this service execution time, provider agents are consulted. These agents use real-time and CBR techniques to calculate the execution time taking the current situation of agents into account. In this way, the client can make a decision with more reliability in the information about services execution time and probability of success.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning; foundational issues, methodological variations, and system approaches. AI Comm. 7(1), 39–59 (1994)
2. Benatallah, B., Hacid, M.S., Rey, C., Toumani, F.: Request rewriting-based web service discovery. In: International Semantic Web Conference, pp. 242–257 (2003)
3. Burns, A., Wellings, A.: Advanced fixed priority scheduling. In: Mathai, J. (ed.) Real-Time Systems, pp. 32–65 (1996)
4. Carman, M., Serafini, L.: PaoloTraverso: Web service composition as planning. In: CAPS 2003 Workshop on Planning for Web Services (2003)
5. Fox, M., Long, D.: Pddl2.1: An extension to pddl for expressing temporal planning domains. J. Artif. Intell. Res (JAIR) 20, 61–124 (2003)
6. Gao, C., Liu, R., Song, Y., Chen, H.: A model checking tool embedded into services composition environment. In: Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC 2006), Washington, DC, USA, pp. 355–362. IEEE Computer Society Press, Los Alamitos (2006)
7. Hashemian, S., Mavaddat, F.: A graph-based approach to web services composition. IEEE Computer Society Press, Los Alamitos (2005)
8. Klusch, M., Gerber, A.: Semantic web service composition planning with owls-xplan. In: Proceedings of the 1st Int. AAAI Fall Symposium on Agents and the Semantic Web, pp. 55–62 (2005)
9. Liu, C.L., Layland, J.W.: Scheduling algorithms for multiprogramming in a hard-real-time environment. ACM 20(1), 46–61 (1973)
10. Vukovic, M., Robinson, P.: Adaptive, planning based, web service composition for context awareness. In: Advances in Pervasive Computing, pp. 247–252 (2004)
11. Walton, C.: Model checking multi-agent web services. In: Proceedings of the 2004 Spring Symposium on Semantic Web Services, Stanford, CA, USA (2004)
12. Navarro, M., Heras, S., Julián, V.: Ensuring Time in Real-Time Commitments. In: Proceedings of the 2008 Iberamia, pp. 183–192 (2008)

An Organisation-Based Multiagent System for Medical Emergency Assistance*

Roberto Centeno[1], Moser Fagundes[1], Holger Billhardt[1], Sascha Ossowski[1],
Juan Manuel Corchado[2], Vicente Julian[3], and Alberto Fernandez[1]

[1] CETINIA, University Rey Juan Carlos, Madrid
[2] Universidad de Salamanca
[3] Universidad Politecnica de Valencia, Spain

Abstract. In this paper we present a demonstrator application for a real-world m-Health scenario: mobile medical emergency management in an urban area. Medical emergencies have a high priority given the potential life risk to the patients. And the use of advanced applications that support the different actors involved can improve the usage of appropriate resources within an acceptable response time. The demonstrator is implemented using the THOMAS approach to open multiagent system based on an organisational metaphor. This metaphor is very suitable for capturing the nature and the complexity of the mobile health domain and, thus, it provides an appropriate mechanism for developing next-generation m-Health applications.

1 Introduction

In the healthcare domain the management of medical emergencies has a huge social impact given the immediate threat to a patient's life or long term health. Such extreme circumstances demand the usage of the appropriate resources within a limited response time in order to provide an efficient assistance.

The employment of advances on computer science to improve the provision of healthcare services is an area of significant economic and social potential. Recently, the notion of m-health has become prominent, focusing on applications that provide healthcare to people anywhere, anytime using broadband and wireless communications, as well as mobile computing devices [2].

In this work we apply an organisational approach to develop a prototype application for a next-generation m-health scenario: the coordination of the different actors (*patient*, *emergency centre*, *ambulances*, *hospitals*, etc.) involved in medical emergency management based on wireless mobile technology. The application is constructed on top of an abstract architecture specified in the THOMAS project (MeTHods, Techniques and Tools for Open Multi-Agent Systems) [7]. The THOMAS platform uses the notion of organisation as a key abstraction to

---

* The present work has been partially funded by the Spanish Ministry of Science and Innovation, grants TIN2006-14630-C03-02 (FPI grants program) and CSD2007-00022 (CONSOLIDER-INGENIO 2010).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 561–568, 2009.
© Springer-Verlag Berlin Heidelberg 2009

build MAS in open environments. It defines an abstract architecture, as well as a software platform, where agents interact according to norms and rules put forward by the organisational structure. Furthermore, the THOMAS platform allows for the integration of two well established research fields: Service-Oriented computing and Multiagent Systems. Thus, it provides a way to build Service-Oriented Multi-Agent Systems (SOMAS) – an approach proposed recently [5].

Our system is based on real-world data provided by the Medical Emergency Service of the Region of Madrid (SUMMA112). This service handles the medical emergencies, including the transportation of patients to hospitals. It counts with approximately 100 call operators, 36 medical doctors, 36 coordination technicians and 83 vehicles (ambulances and helicopters). Its emergency centre receives more than 1.200.000 emergency calls per year almost one call each 30 seconds, and around 60.000 of them are classified as situations of life risk.

The paper is organised as follows. Section 2 describes how the medical emergencies can be modelled as an organisation. Section 3 presents the application prototype designed to deal with these problems. Section 4 explains a case study chosen to illustrate how the application works. Finally, section 5 concludes and points out some future lines of work.

2 mHealth as an Organisation

A typical medical emergency assistance starts when a patient calls SUMMA112, asking for assistance. The call is received by an Operator, who gathers the initial data from the patient. Then she forwards the call to a Medical Doctor, who assigns the resources to attend the request according to its evaluation, however it does not mobilise the resource. Technicians do this task by assigning mobile units taking into account their availability, distance and time to reach the patient, type of resource (ambulances with different capabilities and available resources from other organisations, such as *Red Cross*, *Civil Protection*). Finally, according to the patient condition, she is transported to a hospital.

Analysing this typical scenario, we have determined basically the following actors which are involved along the whole process: *patients*, *coordinator staff*, *physicians*, *ambulances* and *hospitals*. These participants collaborate with each other in order to obtain their goals - to assist patients as fast as possible. Most of the time they act following some protocols/norms which have been determined by some authority. The quality of the whole system depends strongly on the efficiency and effectiveness of this collaboration, that is, on the mechanisms used to organise and coordinate the interactions among the participants.

In order to address both, the nature of the application domain and its high complexity, we propose to model the medical emergency assistance system as a multiagent organisation. In particular, following the framework proposed in [3], we have modelled a mHealth organisation as a particular type of organised multiagent system which is endowed with an organisational mechanism based on norms and roles. The participants in a medical emergency event will be represented by agents that join the organisation playing a particular role. Regarding intervention protocols, they will be modelled as norms that are shared by all participants – agents – within the organisation. We assume that norm compliance is effectively assured through the organisational mechanism. That is, in the realm of the framework in [3], the organisational mechanism used can be classified as a regulative coercive organisational mechanism.

We have identified the following basic roles within medical emergency assistance processes:

- **Patient:** agents playing role patient represent potential users of medical emergency assistance.
- **Hospital:** this role represents a hospital within the organisation.
- **Ambulance:** agents joining the organisation with the role ambulance are representing an ambulance vehicle together with the human resources assigned to it. In our current prototype, we do not differentiate between ambulance vehicles of different types – all emergency vehicles are assumed to offer the same services.
- **Emergency-Coordinator:** this agent is the main piece in the mHealth organisation. An agent playing this role is able to receive emergency calls from patients, and to find the "best" ambulance for each case, thus performing the high-level management of the emergency assistance procedure.

Whereas the roles *Patient, Hospital*, and *Ambulance* can be joined by more than one agent, the role *Emergency-Coordinator* can only be joined by one agent within the mHealth organisation.

Regarding the functional structure of the organisation, we have specified a set of norms that define the possible action patterns that can take place. We have used a very simple set of norms defining: *i) permissions*, and *ii) obligations*. Some examples for the used norms are the following:

- $\mathcal{N}_1$: OBLIGATION $isPlaying(a_i, Ambulance) \wedge isPlaying(a_j, Patient)$ EXECUTE $takePatient(a_i, a_j)$ WHEN $assignMission(a_i, a_j)$
- $\mathcal{N}_2$: OBLIGATION $isPlaying(a_i, Hospital) \wedge isPlaying(a_j, Patient)$ EXECUTE $beAdmited(a_j, a_i)$ WHEN $hasMedicalEmergency(a_j)$
- $\mathcal{N}_3$: PERMISSION $isPlaying(a_i, Patient) \wedge isPlaying(a_j, Emergency\text{-}Coordinator)$ EXECUTE $emergencyCall(a_i, a_j)$

Finally, the identified entities in the process of medical emergency assistance can be classified into two classes: *i)* active entities whose actions are driven from objectives and that possess a certain degree of freedom in their actions, and *ii)* entities that actually act as providers of certain services (e.g. find an ambulance, an hospital, etc.). The former - active entities - will be matched as agents populating the organisation, playing the defined roles. Non-active entities will be considered as resources which can be accessed by agents within the organisation.

3 System Architecture

The aim of our work is to build a system that provides decision support and value added services to the participants who are involved in medical emergency assistances - physicians, patients, hospitals and coordination staff. Following the design proposed in section 2, we have implemented a prototype of the mHealth organisation. In that prototype, the mHealth organisation has been populated with one type of agent for each possible role in the organisation. Thus, we have implemented a Patient agent, Hospital agent, Ambulance agent and a SUMMA agent. The latter plays the role *emergency-coordinator*. All agents have been implement as norm-compliant, that is, they always obey the existing norms.

Regarding the non-active entities, we have modelled them as web services because they only provide a static functionality without following intentions or objectives. We have designed a new role and a new type of agent – *Services-Provider* – which is in charge of providing the web services that implement the non-active entities. Agents which play this role have the obligation to answer when they receives a request of a web service call.

The web services used to implement the resources are the following:

- *Emergency Centre Finder:* this service finds the responsible emergency coordinator for a given location. The result of the invocation of this service will be the identifier of an agent which is playing the *emergency-coordinator* role – in our system this role will be played by the SUMMA112 agent.
- *Medical Record Store:* it allows agents to store and retrieve patients' medical history information trough an user name and password.
- *Ambulance Finder:* This service is an internal service for the emergency-coordinator centre, thus, it can be accessible only by the SUMMA112 agent. The service is able to find an ambulance for a particular patient, taking into account information such as his position, his symptoms, etc.
- *Hospital Finder:* this service is similar to ambulance finder service. It is only accessible for agents which are playing the ambulance role because of they have the obligation to find a hospital when a patient needs to be admitted. The service is able to find a hospital taking into account the patient's position, symptoms, etc.

The main advantage of implementing the non-active entity as web services, specially the ambulance and hospital finder, is that we can easily compare different strategies to assign ambulances and hospitals to patients, simply providing a new implementation of these web services. In the current version both, ambulance finder and hospital finder, return the closest ambulance and the closest hospital (able to treat the diagnosed disease) to the patient's location.

We have developed our prototype on top of the *THOMAS* platform [1]. This platform has been chosen because, on one hand, it uses the organizational paradigm and allows to define the multiagent system based on roles and norms that regulate the agents' behaviours. On the other hand it allows the integration of open multiagent systems and web services. The web services have been specified using the standards OWL-S and WSDL. This makes it possible to use standard techniques for service discovery, composition, etc.

3.1 THOMAS Platform

The THOMAS platform implements an abstract architecture for open multiagent systems based on the organisational paradigm. The agents have access to the functionality offered by THOMAS through a range of services included in several modules. The main components of THOMAS are the following:

- *Service Facilitator (SF):* this component provides simple and complex services to the active agents and organisations. Basically, its functionality is like a yellow page service and a service descriptor in charge of providing a green page service.
- *Organisation Manager Service (OMS):* it is mainly responsible of the management of the organisations and their entities. It allows to create and to manage organisations.
- *Platform Kernel (PK):* it maintains basic management services for an agent platform; it is in charge of communication between agents, etc.

Besides the possibility to structure an application through the concept of organisation, THOMAS allows for a seamless integration of multiagent systems with web services. Web services can be included and can be registered in the service facilitator by means of the services provider agents.

3.2 Spatial Environment Simulator

In order to evaluate our system we have created a spatial environment simulation tool in which the mHealth organisation can be embedded. The environment simulator is an independent module that captures key features of our problem domain. It has two fundamental functions: *i)* it allows to set up agents (and thus, organisations), and *ii)* it recompiles information about the actions that take place in the organisation.

The agents interact with the simulator through a request protocol, where the agents request the execution of an action. The simulator executes that action if the agent plays the required role. The THOMAS platform provides the information about the agents and their roles. In our case, three roles are represented in the physical environment: *patient*, *hospital* and *ambulance*. Once the simulation starts, the environment is constantly updated. Basically, the simulator controls the location of each agent along the time as well as their displacements.

Figure 1 shows the set of available actions for each role as well as the simulator design. *Patients* have only one available action which consists of informing their position in the map. The *hospitals* are able to accept or refuse patients brought by the ambulances, as well as to release the patients. Finally, the *ambulances* are able to catch patients that ask for help, release patients at the hospitals, and move around the physical environment by specifying a destination position (complex routes can be composed by specifying a sequence of positions). The simulation tool is composed of a control layer which handles the agent interactions, and a graphical interface that presents the current state of

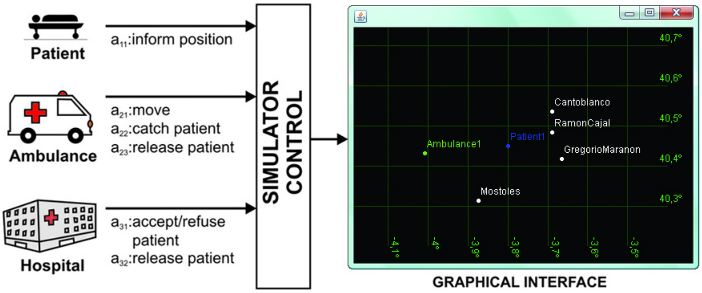


Fig. 1. Simulator design and set of available actions for each role

the system. In Figure 1 the graphical interface shows a simulation with one ambulance (*Ambulance1*), one patient (*Patient1*), and four hospitals (*Cantoblanco*, *GregorioMaranon*, *RamonCajal*, *Mostoles*).

4 Case Study

In this section we present an example of a concrete use case that has been simulated using the mHealth organisation.

In this scenario, Bob is a British tourist visiting Madrid. He suddenly feels a strong pain in his chest and as it is his first time in Madrid, he does not know what he should do. He has his personal agent which joins the mHealth organisation as patient. Immediately, the patient looks for a service that provides the functionality of looking for a medical emergency centre. The services provider agent, using its web service, returns the *SUMMA112's* identifier and a form for specifying personal data and symptoms.

Once *Bob* has selected his symptoms the form is sent to the *SUMMA112* agent. It uses its classification method to make a decision about *Bob*'s case. The preliminary diagnosis says that Bob may be suffering a hearth attack. Therefore, he is classified as a patient that needs an ambulance urgently. The *SUMMA112* agent uses its ambulance finder service to select the most adequate ambulance to take *Bob*. In that case, the closest ambulance to *Bob* is *ambulance3* so the *SUMMA112* agent assigns a new mission to take *Bob*. The ambulance agents have the obligation to take a patient when he needs a medical emergency assistance. Thus, *ambulance3* accepts the mission and goes to take *Bob* immediately.

While *ambulance3* is going to take *Bob*, the *ambulance* agent uses the service "medical record store" to retrieve *Bob*'s medical history. This is an important characteristic because it helps the physicians in the ambulances to make a preliminary decision before actually seeing the patient.

When *ambulance3* arrives at *Bob*'s location, the physician performs an "in situ" diagnosis and decides to which hospital *Bob* should be taken. The *ambulance* agent in *ambulance3* supports this decision by using the *hospital finder service*. Taking into account Bob's location, symptoms and diagnosis, as well as

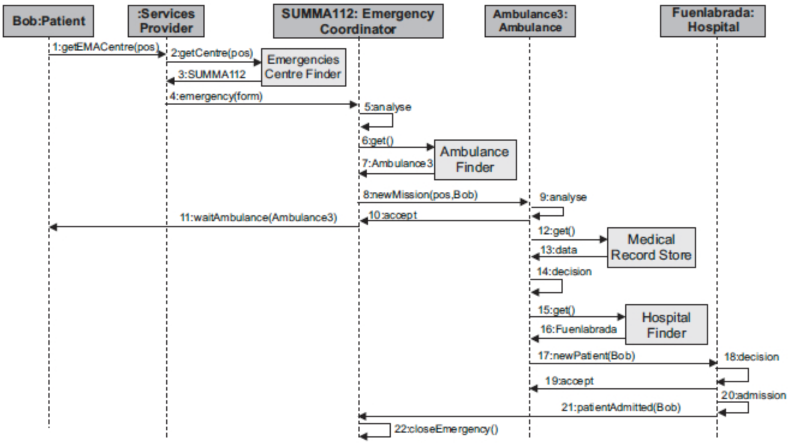


Fig. 2. Case study sequence diagram

the available units in the medical centres, the *Fuenlabrada hospital* is selected because is the nearest hospital which is able to treat heart attacks. Since hospital agents have the obligation to admit patients if they have the capacity, *Bob* is admitted at the *Fuenlabrada hospital*. The hospital agent informs the *SUMMA112* agent about this fact and the *SUMMA112* agent closes the case.

In this scenario we have populated the organisation with 31 agents playing the hospital role, 60 agents playing the role ambulance and one SUMMA112 agent which is playing the emergency-coordinator role. The information of the hospitals (location, bed capacities, etc.) correspond to the hospitals located in the Region of Madrid. Figure 2 shows the sequence of interactions that take place within the organisation in this scenario.

5 Conclusions

In this work we have presented an application of a multiagent system based on organisational principles in the healthcare domain: a prototype application for the support the processes involved in medical emergency assistance. The application provides value added services and decision support to the different actors (patient, emergency centre, ambulances, hospitals, etc.) involved in emergency assistance processes. The organisational paradigm provides a straightforward approach to model the functioning of complex application domains by means of roles and norms. We developed our application as an instantiation of the THOMAS abstract architecture, which explicitly supports multiagent organisations. Furthermore, the THOMAS platform facilitates the integration of non-active entities as web services, following a SOMAS approach. In order to show the system dynamics, we have implemented a case study based on real data provided by the Madrid Medical Emergency Service *SUMMA112*.

Related work, such as [6], has been developed to support medical emergencies processes. In this paper, the authors deal with the problem of selecting the best ambulance to be assigned to an emergency call. In contrast to this work, our approach concentrates on the whole assistance process and tends to provide support to all involved actors. Other related work deals with coordination issues in disaster management in general ([8]), where different teams such as polices, fire brigades, ambulances, etc. have to coordinate each other to assist emergencies. In our work we focus on medical emergencies, in a similar way to the work presented in [4], taking into account just hospitals and ambulances.

In the future we plan to work on mechanisms that may optimise the global utility of the system (e.g., response time and cost) by studying different mechanisms to select the ambulances and hospitals. Finally, we also plan to extend the functionality of the application in order to support other processes of the medical emergency service (e.g., inter-hospital transportation of patients, management of bed capacities of hospitals, etc.).

References

1. Argente, E., Julian, V., Botti, V.: Mas modelling based on organizations. In: 9th Int. Workshop on Agent Oriented Software Engineering, pp. 1–12 (2008)
2. Cáceres, C., Fernández, A., Ossowski, S.: Cascom - context-aware health-care service coordination in mobile computing environments. ERCIM News (60), 77–78 (2005)
3. Centeno, R., Billhardt, H., Hermoso, R., Ossowski, S.: Organising mas: A formal model based on organisational mechanisms. In: 24th Annual ACM Symposium on Applied Computing (SAC 2009), Hawaii, USA, March 8-12 (to appear, 2009)
4. Ciampolini, A., Mello, P., Storari, S.: A multi-agent system for medical services synergy and coordination. In: Nealon, J., Cortes, U., Fox, J., Moreno, A. (eds.) International ECAI 2004 workshop on Agents applied in health care, pp. 38–46 (2004)
5. Huhns, M.N.: A research agenda for agent-based service-oriented architectures. In: Klusch, M., Rovatsos, M., Payne, T.R. (eds.) CIA 2006. LNCS, vol. 4149, pp. 8–22. Springer, Heidelberg (2006)
6. López, B., Innocenti, B., Busquets, D.: A multiagent system to support ambulance coordination of urgent medical transportation. IEEE Intelligent Systems 23(5), 50–57 (2008)
7. Ossowski, S., Julián, V., Bajo, J., Billhardt, H., Botti, V., Corchado, J.M.: Open mas for real world applications: An abstract architecture proposal. In: Proc. XII Conference of the Spanish Association for Artificial Intelligence (CAEPIA), vol. 2(1), pp. 151–160 (2007)
8. Robocup Rescue, `http://www.robocup.rescue`

TEMMAS:
The Electricity Market Multi-Agent Simulator

Paulo Trigo[1], Paulo Marques[1], and Helder Coelho[2]

[1]LabMAg, GuIAA; DEETC, ISEL - Instituto Superior de Eng. de Lisboa, Portugal
ptrigo@deetc.isel.ipl.pt, 28562@alunos.isel.ipl.pt
[2]LabMAg; DI, FCUL - Faculdade de Ciências da Universidade de Lisboa, Portugal
hcoelho@di.fc.ul.pt

Abstract. This paper presents the multi-agent TEMMAS simulator of an artificial electric power market populated with learning agents. The simulator facilitates the integration of two modeling constructs: i) the specification of the environmental physical market properties, and ii) the modeling of the decision-making (deliberative) and reactive agents.

1 Introduction

The start-up of nation-wide electric markets, along with its recent expansion to intercountry markets, aims at providing competitive electricity service to consumers. The new market-based power industry calls for human decision-making in order to settle the energy assets' trading strategies. The interactions and influences among the market participants are usually described by game theoretic approaches which are based on the determination of equilibrium points to which compare the actual market performance [1,2]. However, those approaches find it difficult to incorporate the ability of market participants to repeatedly probe markets and adapt their strategies. Usually, the problem of finding the equilibria strategies is relaxed (simplified) both in terms of: i) the human agents' bidding policies, and ii) the technical and economical operation of the power system. As an alternative to the equilibrium approaches, the multi-agent based simulation (MABS) comes forth as being particulary well fitted to analyze dynamic and adaptive systems with complex interactions among constituents [3,4].

In this paper we describe TEMMAS, a MABS approach to the electricity market, aimed at simulating the interactions of agents and to study the macro--scale effects of those interactions. TEMMAS agents exhibit bounded rationality, i.e., they make decisions based on local information (partial knowledge) of the system and of other agents while learning and adapting their strategies during a simulation in order to reveal and assist to understand the complex and aggregate system behaviors that emerge from the interactions of the market agents.

2 TEMMAS Modeling Framework

We describe the structural TEMMAS constituents by means of two concepts: i) the *environmental entity*, which owns a distinct existence in the real environment,

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 569–576, 2009.
© Springer-Verlag Berlin Heidelberg 2009

e.g. a resource such as an electricity producer, or a decision-making agent such as a market bidder generator company, and ii) the *environmental property*, which is a measurable aspect of the real environment, e.g. the price of a bid or the demand for electricity. Hence, we define the *environmental entity* set, $\mathcal{E}_{\mathcal{T}} = \{ e_1, \ldots, e_n \}$, and the *environmental property* set, $\mathcal{E}_{\mathcal{Y}} = \{ p_1, \ldots, p_n \}$. The whole environment is the union of its entities and properties: $\mathcal{E} = \mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{Y}}$.

The environmental entities, $\mathcal{E}_{\mathcal{T}}$, are often clustered in different classes, or types, thus partitioning $\mathcal{E}_{\mathcal{T}}$ into a set, $\mathcal{P}_{\mathcal{E}_{\mathcal{T}}}$, of disjoints subsets, $\mathcal{P}^i_{\mathcal{E}_{\mathcal{T}}}$, each containing entities that belong to the same class. Formally, $\mathcal{P}_{\mathcal{E}_{\mathcal{T}}} = \{ \mathcal{P}^1_{\mathcal{E}_{\mathcal{T}}}, \ldots, \mathcal{P}^k_{\mathcal{E}_{\mathcal{T}}} \}$ defines a full partition of $\mathcal{E}_{\mathcal{T}}$, such that $\mathcal{P}^i_{\mathcal{E}_{\mathcal{T}}} \subseteq \mathcal{E}_{\mathcal{T}}$ and $\mathcal{P}_{\mathcal{E}_{\mathcal{T}}} = \cup_{i=1 \ldots k} \mathcal{P}^i_{\mathcal{E}_{\mathcal{T}}}$ and $\mathcal{P}^i_{\mathcal{E}_{\mathcal{T}}} \cap \mathcal{P}^j_{\mathcal{E}_{\mathcal{T}}} = \emptyset \; \forall i \neq j$. The partitioning may be used to distinguish between decision-making agents and available resources, e.g. a company that decides the biding strategy to pursue or a plant that provides the demanded power.

The environmental properties, $\mathcal{E}_{\mathcal{Y}}$, can also be clustered, in a similar way as for the environmental entities, thus grouping properties that are related. The partitioning may be used to express distinct categories, e.g. economical, electrical, ecological or social aspects. Another, more technical usage, is to separate constant parameters from dynamic state variables.

The factored state space representation. The state of the simulated environment is implicitly defined by the state of all its environmental entities and properties. We follow a factored representation, that describes the state space as a set, $\mathcal{V}$, of discrete state variables [5]. Each state variable, $v_i \in \mathcal{V}$, takes on values in its domain $\mathcal{D}(v_i)$ and the global (i.e., over $\mathcal{E}$) state space, $\mathcal{S} \subseteq \times_{v_i \in \mathcal{V}} \mathcal{D}(v_i)$, is a subset of the Cartesian product of the sate variable domains. A state $s \in \mathcal{S}$ is an assignment of values to the set of state variables $\mathcal{V}$. We define $f_{\mathcal{C}}$, $\mathcal{C} \subseteq \mathcal{V}$, as a projection such that if s is an assignment to $\mathcal{V}$, $f_{\mathcal{C}}(s)$ is the assignment of s to $\mathcal{C}$; we define a *context* c as an assignment to the subset $\mathcal{C} \subseteq \mathcal{V}$; the initial state variables of each entity and property are defined, respectively, by the functions $init_{\mathcal{E}_{\mathcal{T}}} : \mathcal{E}_{\mathcal{T}} \rightarrow \mathcal{C}$ and $init_{\mathcal{E}_{\mathcal{Y}}} : \mathcal{E}_{\mathcal{Y}} \rightarrow \mathcal{C}$.

The decision-making approach. Each agent perceives (the market) and acts (sells or buys) and there are two main approaches to develop the reasoning and decision-making capabilities: i) the qualitative mental-state based reasoning, such as the belief-desire-intention (BDI) architecture [6], which is founded on logic theories, and ii) the quantitative, decision-theoretic, evaluation of causal effects, such as the Markov decision process (MDP) support for sequential decision--making in stochastic environments. There are also hybrid approaches that combine the qualitative and quantitative formulations [7,8].

The qualitative mental-state approaches capture the relation between high level components (e.g. beliefs, desires, intentions) and tend to follow heuristic (or rule-based) decision-making strategies, thus being better fitted to tackle large-scale problems and worst fitted to deal with stochastic environments.

The quantitative decision-theoretic approaches deal with low level components (e.g., primitive actions and immediate rewards) and searches for long-term policies that maximize some utility function, thus being worst fitted to tackle large-scale problems and better fitted to deal with stochastic environments.

The electric power market is a stochastic environment and we currently formulate medium-scale problems that can fit a decision-theoretic agent model. Therefore, TEMMAS adaptive agents (e.g., market bidders) follow a MDP based approach and resort to experience (sampled sequences of states, actions and rewards from simulated interaction) to search for optimal, or near-optimal, policies using reinforcement learning methods such as Q-learning [9] or SARSA [10].

3 TEMMAS Design

Within the current design model of TEMMAS the electricity asset is traded through a spot market (no bilateral agreements), which is operated via a *Pool* institutional power entity. Each generator company, *GenCo*, submits (to *Pool*) how much energy, each of its generating unit, $GenUnit_{GenCo}$, is willing to produce and at what price. Thus, we have: i) the power supply system comprises a set, $\mathcal{E}_{GenCo}$, of generator companies, ii) each generator company, *GenCo*, contains its own set, $\mathcal{E}_{GenUnit_{GenCo}}$, of generating units, iii) each generating unit, $GenUnit_{GenCo}$, of a *GenCo*, has constant marginal costs, and iv) the market operator, *Pool*, trades all the *GenCo*s' submitted energy.

The bidding procedure conforms to the so-called "block bids" approach [11], where a block represents a quantity of energy being bided for a certain price; also, *GenCo*s are not allowed to bid higher than a predefined price ceiling. Thus, the market supply essential measurable aspects are the energy price, quantity and production cost. The consumer side of the market is mainly described by the quantity of demanded energy; we assume that there is no price elasticity of demand (i.e., no demand-side market bidding).

Therefore, we have: $\mathcal{E}_{\mathcal{T}} = \{ Pool \} \cup \mathcal{E}_{GenCo} \cup_{g \in \mathcal{E}_{GenCo}} \mathcal{E}_{GenUnit_g}$ where $\mathcal{E}_{\mathcal{Y}} = \{ quantity, price, productionCost \}$. The *quantity* refers both to the supply and demand sides of the market. The *price* referes both to the supply bided values and to the market settled (by *Pool*) value.

The $\mathcal{E}_{GenCo}$ contains the decision-making agents. The *Pool* is a reactive agent that always applies the same predefined auction rules in order to determine the market price and hence the block bids that clear the market. Each $\mathcal{E}_{GenUnit_{GenCo}}$ represents the *GenCo*'s set of available resources.

The resources' specification. Each generating unit, $GenUnit_{GenCo}$, defines its marginal costs and constructs the block bids according to the strategy indicated by its generator company, *GenCo*. Each $GenUnit_{GenCo}$ calculates its marginal costs according to, either the "WithHeatRate" [12]) or the "WithCO_2" [13] formulation.

The "WithHeatRate" formulation estimates the marginal cost, *MC*, by combining the variable operations and maintenance costs, *vO&M*, the number of heat rate intervals, *nPat*, each interval's capacity, cap_i and the corresponding heat rate value, hr_i, and the price of the fuel, $fPrice$, being used; the marginal cost for a given $i \in [1, nPat]$ interval is given by,

$$MC_{i+1} = vO\&M + \frac{(cap_{i+1} \times hr_{i+1}) - (cap_i \times hr_i)}{blockCap_{i+1}} \times fPrice \tag{1}$$

where each block's capacity is given by: $blockCap_{i+1} = cap_{i+1} - cap_i$.

The "WithCO_2" marginal cost, MC, combines the variable operations and maintenance costs, $vO\&M$, the price of the fuel, $fPrice$, the CO_2 cost, CO_2cost, and the unit's productivity, η, through the expression,

$$MC = \frac{fPrice}{\eta} \times K + CO_2cost + vO\&M \tag{2}$$

where K is a fuel-dependent constant factor, and CO_2cost is given by,

$$CO_2cost = CO_2price \times \frac{CO_2emit}{\eta} \times K \tag{3}$$

where CO_2emit is the CO_2 fuel's emissions. Here all blocks have the same capacity; given a unit's maximum capacity, $maxCap$, and a number of blocks, $nBlocks$, to sell, each block's capacity is given by: $blockCap = \frac{maxCap}{nBlocks}$.

The decision-making strategies. Each generator company defines the bidding strategy for each of its generating units. We designed two types of strategies: a) the *basic-adjustment*, that chooses among a set of basic rigid options, and b) the *heuristic-adjustment*, that selects and follows a predefined well-known heuristic. There are several *basic-adjustment* strategies already defined in TEMMAS. Here we outline seven of those strategies, $sttg_i$ where $i \in \{1, \ldots, 7\}$, available for a $GenCo$ to apply: i) $sttg_1$, bid according to the marginal production cost of each $GenUnit_{GenCo}$ (follow heat rate curves, e.g., cf. Table 2), ii) $sttg_2$, make a "small" increment in the prices of all the previous-day's block bids, iii) $sttg_3$, similar to $sttg_2$, but makes a "large" increment, iv) $sttg_4$, make a "small" decrement in the prices of all the previous-day's block bids, v) $sttg_5$, similar to $sttg_4$, but makes a "large" decrement, vi) $sttg_6$, hold the prices of all previous-day's block bids, vii) $sttg_7$ set the price to zero. There are two *heuristic-adjustment* defined strategies: a) the "Fixed Increment Price Probing" (FIPP) that uses a percentage to increment the price of last day's transacted energy blocks and to decrement the non-transacted blocks, and b) "Physical Withholding based on System Reserve" (PWSR) that reduces the block's capacity, as to decrement the next day's estimated system reserve (difference between total capacity and total demand), and then bids the remaining energy at the maximum market price.

The agents' decision process. The above strategies correspond to the $GenCo$ agent's primary actions. The $GenCo$ has a set, $\mathcal{E}_{GenUnit_{GenCo}}$, of generating units and, at each decision-epoch, it decides the strategy to apply to each generating unit, thus choosing a vector of strategies, $\overrightarrow{sttg}$, where the i^{th} vector's component refers to the $GenUnit^i_{GenCo}$ generating unit; thus, its action space is given by:

$\mathcal{A} = \times_{i=1}^{|\mathcal{E}_{GenUnit_{GenCo}}|} \{ sttg_1, \ldots, sttg_7 \}_i \cup \{ \text{FIPP}, \text{PWSR} \}$. The *GenCo*'s perceived market share, *mShare*, is used to characterize the agent internal memory so its state space is given by $mShare \in [0..100]$. Each *GenCo* is a MDP decision-making agent such that the decision process *period* represents a daily market. At each decision-epoch each agent computes its daily profit (that is regarded as an internal reward function) and the the *Pool* agent receives all the *GenCo*s's block bids for the 24 daily hours and settles the hourly market price by matching offers in a classic supply and demand equilibrium price (we assume a hourly constant demand).

TEMMAS *architecture and construction.* The TEMMAS agents along with the major inter-agent communication paths are represented in the bottom region

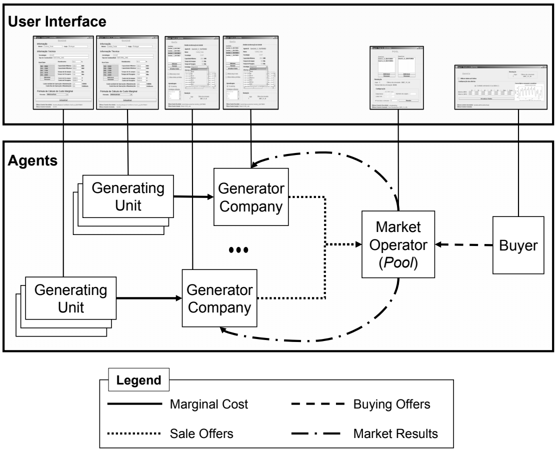


Fig. 1. The TEMMAS architecture and the configurable parameters

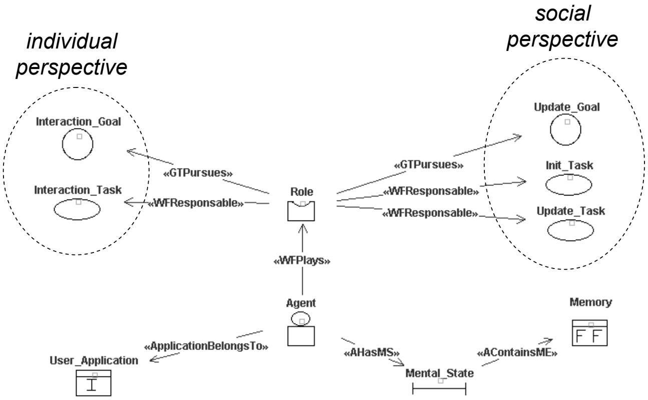


Fig. 2. TEMMAS agent's view using INGENIAS framework

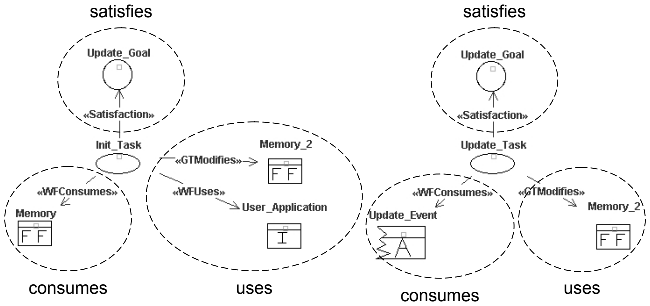


Fig. 3. TEMMAS tasks and goals specification using INGENIAS framework

of Figure 1; the top region represents the user interface that enables to specify the each of the resources' and agents' configurable parameters. The implementation of the TEMMAS architecture followed the INGENIAS [14] methodology and used its supporting development platform. Figure 2 presents the general "agent's perspective", where the tasks and the goals are clustered into individual and social perspectives. Figure 3 gives additional detail on the construction of tasks and goals using INGENIAS.

4 TEMMAS Illustrative Setup

We used TEMMAS to build an electric market simulation model. Our experiments have two main purposes: i) illustrate the TEMMAS functionality, and ii) analyze the agents' resulting behavior, e.g. the learnt bidding policies, in light of the market specific dynamics; cf. [15] for our extended experimental setup.

We considered three types of generating units: i) one base load coal plant, CO, ii) one combined cycle plant, CC, to cover intermediate load, and iii) one gas turbine, GT, peaking unit. Table 1 shows the essential properties of each plant type and Table 2 shows the heat rate curves used to define the bidding blocks. The marginal cost was computed using expression (1); the bidding block's quantity is the capacity increment, e.g. for CO, the 11.9 marginal cost bidding block's quantity is $350 - 250 = 100$ MW (cf. Table 2, CO, top lines 2 and 1).

Table 1. Properties of generating units; the units' types are coal (CO), combined cycle (CC) and gas turbine (GT); the O&M indicates "operation and maintenance" cost

Property	*unit*	**Type of generating unit**		
		CO	**CC**	**GT**
Fuel	—	Coal (BIT)	Nat. Gas	Nat. Gas
Capacity	MW	500	250	125
Fuel price	€/MMBtu	1.5	5	5
Variable O&M	€/MWh	1.75	2.8	8

Table 2. Each generating unit's capacity block (MW) and heat rate (Btu/kWh) and the corresponding marginal cost (€/MWh)

CO generating unit			**CC generating unit**			**GT generating unit**		
Cap.	Heat rate	Marg. cost	Cap.	Heat rate	Marg. cost	Cap.	Heat rate	Marg. cost
250	12000	—	100	9000	—	50	14000	—
350	10500	11.9	150	7800	29.8	100	10600	44.0
400	10080	12.5	200	7200	29.8	110	10330	46.2
450	9770	12.7	225	7010	30.3	120	10150	48.9
500	9550	13.1	250	6880	31.4	125	10100	52.5

Table 3. The experiment's *GenCo*s and *GenUnit*s

GenCo		
name	Prod. Capac.	***GenUnits***
GenCo_major	2000	2×CO & 4×CC
GenCo_minor&active	875	3×CC & 1×GT

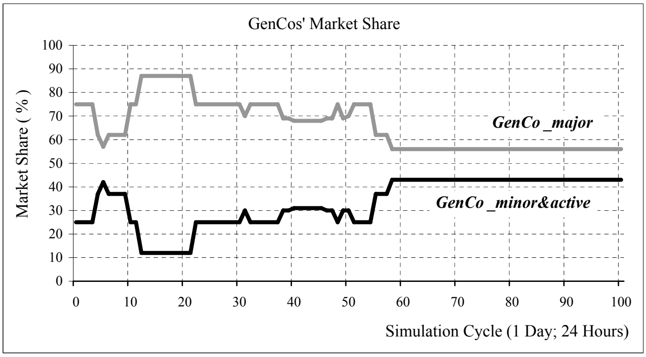


Fig. 4. Market share evolution induced by *GenCo_minor&active*. [*Exp. #2*]

We designed a simple experimental scenario and Table 3 shows the *GenCo*'s name and its production capacity according to the respective *GenUnit*s (cf. Table 1). The "*active*" suffix (cf. Table 3, *name* column) means that *GenCo* searches for its *GenUnit*s best bidding strategies; i.e. "*active*" is a policy learning agent.

We considered a constant, 2000 MW, hourly demand for electricity. Figure 4 shows the market share evolution while *GenCo_minor&active* learns to play in the market with *GenCo_major*, which is a larger company with a fixed strategy: "bid each block 5€ higher than its marginal cost". We see that *GenCo_minor&active* gets around 18% $(75 - 57)$ of market from *GenCo_major*. To earn that market the *GenCo_minor&active* learnt to lower its prices in order to exploit the "5€ space" offered by *GenCo_major* fixed strategy.

5 Conclusions and Future Work

This paper describes our preliminary work in the construction of the TEMMAS agent-based electricity market simulator. Our contribution includes a comprehensive formulation of the simulated electric power market environment along with the inhabiting decision-making and learning agents. Our initial results reveal an emerging and coherent market behavior, thus inciting us to extend TEMMAS with additional bidding strategies and to incorporate specific market rules, such as congestion management and pricing regulation mechanisms.

References

1. Berry, C., Hobbs, B., Meroney, W., O'Neill, R., Jr, W.S.: Understanding how market power can arise in network competition: a game theoretic approach. Utilities Policy 8(3), 139–158 (1999)
2. Gabriel, S., Zhuang, J., Kiet, S.: A Nash-Cournot model for the north american natural gas market. In: Proceedings of the 6th IAEE European Conference: Modelling in Energy Economics and Policy (September 2-4, 2004)
3. Schuster, S., Gilbert, N.: Simulating online business models. In: Proceedings of the 5th Workshop on Agent-Based Simulation (ABS 2004), May 3-5, 2004, pp. 55–61 (2004)
4. Helleboogh, A., Vizzari, G., Uhrmacher, A., Michel, F.: Modeling dynamic environments in multi-agent simulation. JAAMAS 14(1), 87–116 (2007)
5. Boutilier, C., Dearden, R., Goldszmidt, M.: Exploiting structure in policy construction. In: Proceedings of the IJCAI 1995, pp. 1104–1111 (1995)
6. Rao, A., Georgeff, M.: BDI agents: From theory to practice. In: Proceedings of the First International Conference on Multiagent Systems, S, pp. 312–319 (1995)
7. Simari, G., Parsons, S.: On the relationship between MDPs and the BDI architecture. In: Proceedings of the AAMAS 2006, May 8–12, pp. 1041–1048 (2006)
8. Trigo, P., Coelho, H.: Decision making with hybrid models: the case of collective and individual motivations. International Journal of Reasoning Based Intelligent Systems (IJRIS); Inderscience Publishers (2009)
9. Watkins, C., Dayan, P.: Q-learning. Mach. Learning 8, 279–292 (1992)
10. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
11. OMIP - The Iberian Electricity Market Operator, `http://www.omip.pt`
12. Botterud, A., Thimmapuram, P., Yamakado, M.: Simulating GenCo bidding strategies in electricity markets with an agent-based model. In: Proceedings of the 7th Annual IAEE European Energy Conference (IAEE 2005), August 28–30 (2005)
13. Sousa, J.: How market players aadjusted their strategic behaviour taking into account the CO2 emission costs - an application to the spanish electricity market. In: Proceedings of the 4th International Conference on the European Electricity Market (EEM 2007), Cracow, Poland, May 23–27 (2007)
14. Gómez-Sanz, J., Fuentes-Fernández, R., Pavón, J., García-Magariño, I.: INGENIAS development kit: a visual multi-agent system development environment (BEST ACADEMIC DEMO OF AAMAS 2008). In: Proceedings of the Seventh AAMAS, Estoril, Portugal, May 12-16, 2008, pp. 1675–1676 (2008)
15. Trigo, P., Marques, P.: The electricity market as a multi-agent system. In: Proceedings of the 5th International Conference on the European Electricity Market, Lisbon, Portugal, May 28–30, 2008. IEEE Xplore (2008)

Two Steps Reinforcement Learning in Continuous Reinforcement Learning Tasks

Iván López-Bueno, Javier García, and Fernando Fernández

Universidad Carlos III de Madrid
Avda. de la Universidad 30, Madrid, Spain
ivan.lopez-bueno@alumnos.uc3m.es, fjaviergp@gmail.com,
ffernand@inf.uc3m.es

Abstract. Two steps reinforcement learning is a technique that combines an iterative refinement of a Q function estimator that can be used to obtains a state space discretization with classical reinforcement learning algorithms like Q-learning or Sarsa. However, the method requires a discrete reward function that permits learning an approximation of the Q function using classification algorithms. However, many domains have continuous reward functions that could only be tackled by discretizing the rewards. In this paper we propose solutions to this problem using discretization and regression methods. We demonstrate the usefulness of the resulting approach to improve the learning process in the Keepaway domain. We compare the obtained results with other techniques like VQQL and CMAC.

1 Introduction

Two Steps Reinforcement Learning (2SRL) is a method to compute the action-value function in model free Reinforcement Learning [5]. The method assumes a reduced set of actions and finite trials, where positive and discrete rewards can be obtained only when a goal area is achieved. It is based on finding discretizations of the state space that are adapted to the value function being learned, trying to keep the convergence properties of the discretized methods using non-uniform discretizations [11]. In this case, we learn the action value function, $Q(s, a)$, instead of the state value function, $V(s)$, learned in Dynamic Programming [9]. The method is based on two learning phases. The first one is a model free version of the Smooth Value Iteration algorithm [2], that is called Iterative Smooth Q-Learning (ISQL). This algorithm, that executes an iterative supervised learning of the Q function, can be used to obtain a state space discretization too. This new discretization is used in a second learning phase, that is called Multiple Discretization Q-Learning (MDQL), to obtain an improved policy. Performing both phases demonstrated more accurate results in different domain [5].

However, the method presented a main drawback: it requires a discrete reward function. Using such discrete reward function permits ISQL to learn a function approximation of the Q function applying classification algorithms such as J48,

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 577–584, 2009.
© Springer-Verlag Berlin Heidelberg 2009

an algorithm to learn decision trees [10]. However, many domains, like Keepaway, have continuous reward functions. In this case, 2SRL needs to discretize the reward space by hand, and to test different discretizations to obtain an accurate one.

To apply the same ideas of 2SRL in domains with continuous rewards requires that the function approximation used in the ISQL phase be a regression approach. As before, the approximation method should generate a discretization of the state space, so such discretization can be used in the second learning phase. Fortunately, there are different approaches in the literature for regression that generate state space discretizations of the input space. Classical ones are M5 [10] or PART [4], which generate regression trees and regression rules respectively.

In this work we adapt the original 2SRL algorithm to deal with domains with continuous rewards. Specifically, we use Keepaway [8], a sub-task of the RoboCup simulator, to perform the evaluation. We compare the new approach with the original algorithm (using discretized rewards), and also with previous approaches like VQQL [3] and CMAC [8], obtaining very compiling results.

This paper is organized as follows. Section 2 describes the algorithm 2SRL in domains with continuous rewards. Section 3 shows the evaluation of the new approach. Last, Section 4 describes the main conclusions.

2 Two Steps Reinforcement Learning in Domains with Continuous Rewards

The use of 2SRL requires, mainly, the adaptation of the Iterative Smooth Q-Learning algorithm (ISQL). ISQL is derived of the Discrete Value Iteration [1], where a function approximator is used instead of the tabular representation of the value function, so the algorithm can be used in the continuous state space case. The algorithm is described in Figure 1. The update equation of the Q function used is the stochastic Q-Learning update equation. The figure shows the adapted version of the original ISQL algorithm to permit continuous rewards.

The algorithm assumes a discrete set of L actions, and hence, it will generate L function approximators, $Q_{a_i}(s)$. It requires a collection of experience tuples, T. Different methods can be applied to perform this exploration phase, from random exploration to human driven exploration [7]. In each iteration, from the initial set of tuples, T, and using the approximators $\hat{Q}_{a_i}^{iter-1}(s)$, $i = 1, \ldots, L$, generated in the previous iteration, the Q-learning update rule for deterministic domains[1] can be used to obtain L training sets, $T_{a_i}^{iter}$, $i = 1, \ldots, L$, with entries of the kind $< s_j, c_j >$ where c_j is the resulting value of applying the Q update function to the training tuple j, whose state is s_j.

In the first iteration, $\hat{Q}_{a_i}^{0}(s)$ are initialized to 0, for $i = 1, \ldots, L$, and all $s \in S$. Thus, when the respective c_j are computed, they depend only on the possible values of the immediate reward, r.

---

[1] The Q-learning update rule for stochastic domains can be used too.

```
Iterative Smooth Q-Learning with Regression

– Inputs:
    1. A state space $X$
    2. A discrete set of L actions, $A = \{a_1, \ldots, a_L\}$
    3. A collection $T$ of $N$ experience tuples of the kind $< s, a_i, s', r >$, where $s \in X$
       is a state where action $a_i$ is executed, $s' \in X$ is the next state visited and $r$ is
       the immediate reward received
– Generate $L$ initial approximators of the action-value function $\hat{Q}^0_{a_i} : X \rightarrow \mathcal{R}$, and
  initialize them to return 0
– $iter = 1$
– Repeat
    • For all $a_i \in A$, initialize the learning sets $T^{iter}_{a_i} = \emptyset$
    • For j=1 to N, using the $j^{th}$ tuple $< s_j, a_j, s'_j, r_j >$ do
        * $c_j = \alpha c_j + (1 - \alpha) \max_{a_r \in A} \gamma \hat{Q}^{iter-1}_{a_r}(s'_j)$
        * $T^{iter}_{a_j} = T^{iter}_{a_j} \cup \{< s_j, c_j >\}$
    • For each $a_i \in A$, train $\hat{Q}^{iter}_{a_i}$ to approximate the learning set $T^{iter}_{a_i}$
    • $iter = iter + 1$
  Until $r_{max}$ is propagated to the whole domain
– Return $\hat{Q}^{iter-1}_{a_i}$, $\forall a_i \in A$
```

Fig. 1. *Iterative Smooth Q-Learning* algorithm for domains with continuous rewards

The discretizations obtained in this first phase can be used to tune, in a second phase, the action value function obtained in the previous phase, following a multiple (one per action) discretization based approach. This second phase was called Multiple Discretization Q-Learning (MDQL).

The figure 2 shows how to move from the first scheme to the second one, when an approximation like M5 Rules has been used, and L rule sets have been obtained (one for each action). Then, we transform the *RuleSet* obtained by a M5 model tree/rules into a M5 regression tree/rules called *RuleSet'*. We apply this transformation because the M5 model tree/rules uses a linear regression model on the right part of the rule and we want to obtain a numerical value. If we use the M5 regression tree/rules we get a number of the right part of the rules. That could be more useful in the MDQL phase. The left part of the rules of the approximator *RuleSet' i* will be used as the discretization $D_i(s)$, and the right part of the rules of *RuleSet' i* will be located in column i of the generated Q table, providing an initialization of the Q table in the second learning phase[2]. Each RuleSet' may have a different number of rules, so the number of rows of the table is given by the maximum number of rules of the L approximators, and zero values can be used to complete the columns with less rules.

Once the translation from the first scheme to the second one is done, a new learning phase can be executed using the Q-learning update function shown in equation 1.

$$Q(s_t, a_t) \rightarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \qquad (1)$$

---

[2] We also can initialize to 0 and not to use the right part of the rules.

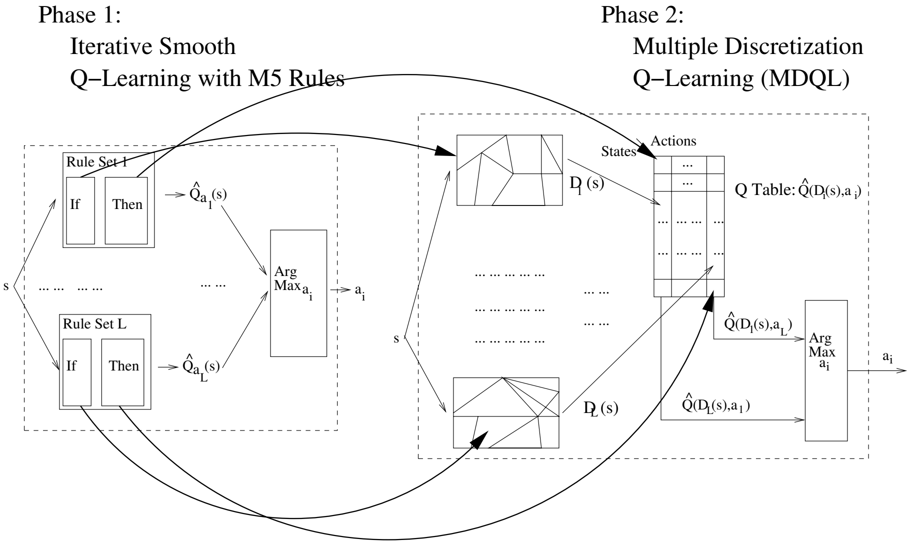


Fig. 2. The two steps of the 2SRL algorithm with Regression

The second learning phase can be executed with new experiences or with the same ones used in the first phase. In the experimentation performed next, the first approach is applied using new experiences. This second phase can be executed exactly as it was defined in the original 2SRL algorithm, so additional explanation of this phase, and how to connect both phases, can be found in the literature [5].

3 Evaluation

To evaluate the new approach, we use the Keepaway subtask of the RoboCup domain [8].

3.1 Keepaway Domain

In this subtask, the keepers try to maintain the possession of the ball within a region, while the takers try to gain the possession. An episode ends when the takers take the possession or the ball leaves the playing region. When an episode ends, the players are reset for another episode. In our experiments each keeper learns independently. From a point of view of a keeper, an episode consists of a sequence of states, actions and rewards:

$$s_0, a_0, r_1, s_1, ..., s_i, a_i, r_{i+1} \tag{2}$$

where a_i is selected based on some perception of s_i. We reward the keepers for each time step which they keep possession, so we set the reward r_i to the number of time steps that elapsed while following action $a_{i-1} : r_i = t_i - t_{i-1}$. The keepers' goal is to select an action that the remainder of the episode will be as long as possible, and thus maximize the total reward.

The state space is composed of several features that can be considered continuous. These features use information derived from distances and angles between the keepers, takers and the center of the playing area. The number of features used for each state depends on the number of players. In 3vs2 there are 13 features, in 4vs3 there are 19 and in 5vs4 there are 25 features. In the experiments, we use VQ, CMAC and 2SRL as methods for state space generalization in order to improve the learning process. The action space is composed by two different macro-actions, *HoldBall()* and $PassBall(k_i)$ where k_i is the teammate i.

3.2 Results of Two Steps Reinforcement Learning in RoboCup Keepaway

We use both M5 and J48 (C4.5) algorithms to make an approximation of the Q function for each iteration in the ISQL phase. In the case of J48 [10] we have discretized the reward function by hand, and we have applied the original 2SRL algorithm. For M5 [10], we use the adapted version proposed in this paper. The approximator obtained in *ith* iteration is used as a state space generalization in MDQL and the reward tuples are updated with the stochastic Q-Learning update function. The first tuple set was obtained with a random policy working in the keepaway domain.

The parameter setting for 2SRL is the following: $\gamma = 1$, $\alpha = 0.125$. An $\epsilon - greedy$ strategy is followed, increasing the value of epsilon from 0 (random behaviour) to 1 (fully greedy behaviour) by 0.0001 in each episode. The size of the playing region is 25×25. The Q table is initialized to 0. In 3vs2, we obtain the results shown in Figure 3.

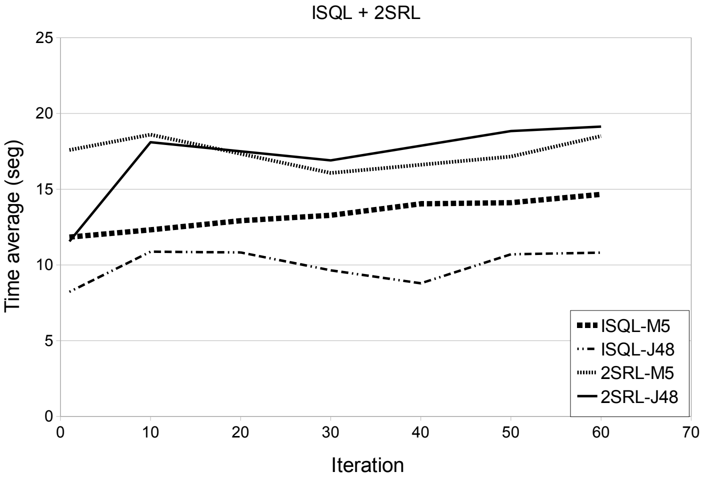


Fig. 3. The two steps of the 2SRL algorithm with M5 and J48

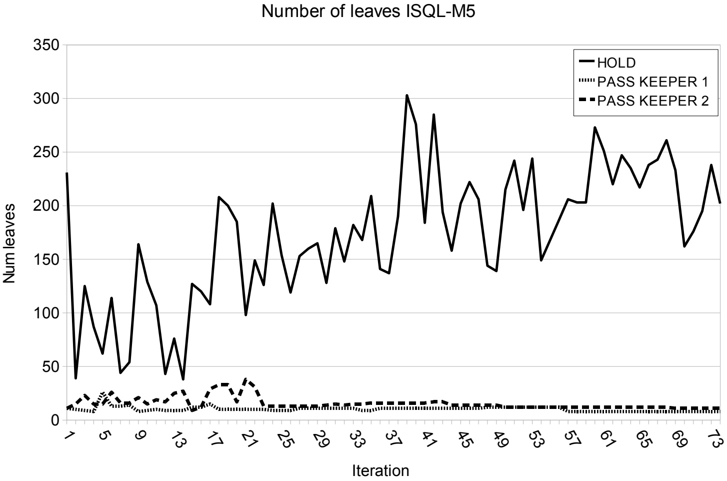


Fig. 4. Evolution of the number of leaves for ISQL-M5

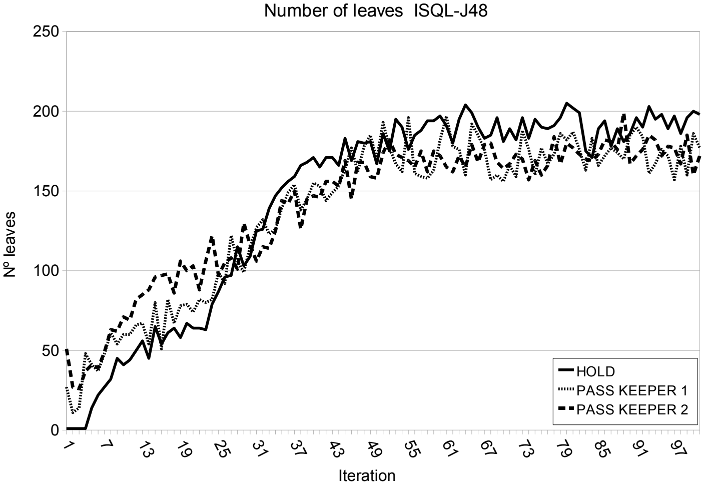


Fig. 5. Evolution of the number of leaves for ISQL-J48

In the graphs, there are four learning curves using ISQL-M5, ISQL-J48, 2SRL-M5 and 2SRL-J48. The *y-axis* is the average episode length when a stable policy is obtained and the *x-axis* is the ISQL iteration. The ISQL-M5 curve was generated using M5 model trees obtaining a very good improvement. The ISQL-J48 curve was generated using the best discretization of the reward space obtained with 88 classes and the performance is worse than M5 model. However, M5 model tree generates a very small state space discretization (Figure 4). For this reason, we transform the *RuleSet* into *RuleSet'* like we explain in section 2.

The trees generated with J48 algorithm have too many leaves (≈ 12000 per action) and the tree leaves are used as space state discretization. We need a smaller discretization more useful for the second phase of 2SRL. We apply different pruning methods to curb the tree growth and to stabilize the space state discretization (Figure 5).

When we use tuned M5 and J48 (88 classes) trees in MDQL we obtain better results than M5 and J48 (88 classes) without tuning. We can see in Figure 3 that the trees obtained from the last iterations of ISQL are more useful in MDQL than the first ones. In the experiments, M5 obtains better or similar results than the best configuration of J48 for ISQL and 2SRL respectively.

3.3 Comparison with Different Approaches

In this section we compare 2SRL with other techniques: the best configurations obtained by hand for VQQL [3] and CMAC [8].

In the experiments presented in this report, 2SRL and VQQL obtain similar results in spite of to be very different techniques. The best VQQL experiment has 64 states and 2SRL has 274(M5) and 553(J48) states. In complex domains with a large state space, the choice of the discretization method may have dramatic effect on the results that we obtain. For this reason, the state representation is chosen with great care. In our experiments we have used ISQL-2SRL to tune the state space discretization obtained from M5 and J48 algorithms. CMAC results [8] are better than 2SRL and VQ results in 3vs2 (Figure 6).

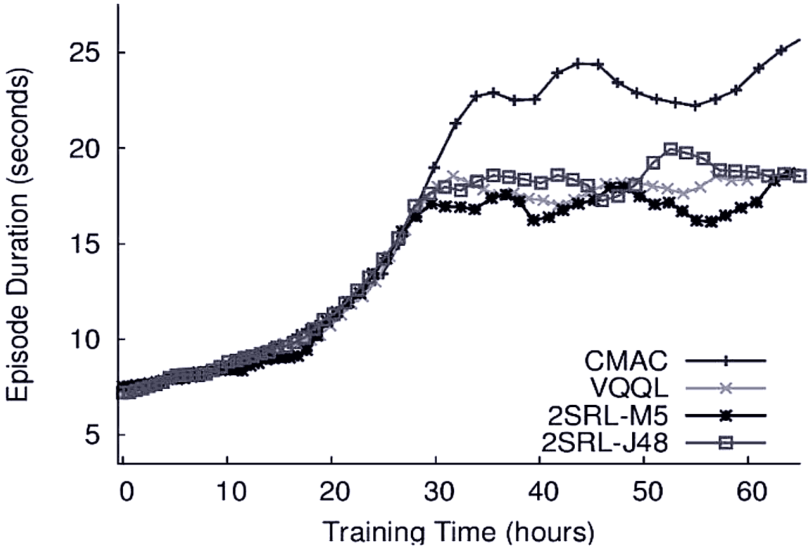


Fig. 6. CMAC, VQQL, 2SRL algorithm with M5 and J48

4 Conclusions

In this paper we have applied two steps reinforcement learning in Keepaway domain. Keepaway domain has continuous rewards and we have adapted the original 2SRL algorithm to deal with domains with continuous rewards. We use M5 and J48 to approximate the Q function in the first phase and we use the same approximators as discretization of the continuous keepaway state space, obtaining very successful results. In the experiments, we present the result of the best J48 discretization obtained after an exhaustive experimentation evaluating different discretizations of the reward function. M5 does not need to discretize by hand the reward space and it obtains better or similar results than J48. We have shown that 2SRL can be used in complex domains with continuous rewards obtaining better results than using only the first phase of the algorithm.

Acknowledgements

This work has been partially sponsored by a regional project CCG08-UC3M/TIC-4141 of the *Comunidad de Madrid*, and a national project TIN2008-06701-C03-03 of the *Ministerio de Ciencia e Innovación* of Spain.

References

1. Bellman, R.: Dynamic programming. Princeton University Press, Princeton (1957)
2. Boyan, J.A., Moore, A.W.: Generalization in reinforcement learning: Safely approximating the value function. Adva. Neural Inform. Process. Syst. 7 (1995)
3. Fernández, F., Borrajo, D.: VQQL. Applying vector quantization to reinforcement learning. In: Veloso, M.M., Pagello, E., Kitano, H. (eds.) RoboCup 1999. LNCS, vol. 1856, pp. 292–303. Springer, Heidelberg (2000)
4. Frank, E., Witten, I.H.: Generating Accurate Rule Sets Without Global Optimization. In: Fifteenth International Conference on Machine Learning, pp. 144–151 (1998)
5. Fernández, F., Borrajo, D.: Two steps reinforcement learning. Internation Journal of Intelligent Systems 23, 213–245 (2008)
6. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement Learning: A Survay (1996)
7. Smart, W.D.: Making reinforcement learning work on real robots. PhD thesis, Department of Computer Science at Brown University, Providence, RI (May 2002)
8. Stone, P., Sutton, R., Kuhlmann, G.: Reinforcement Learning for RoboCup-Soccer Keepaway. Adaptive Behaviour 13(3) (2005)
9. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
10. Frank, E., Witten, I.H.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
11. Munos, R., Moore, A.: Variable Resolution Discretization in Optimal Control. In: Machine Learning, pp. 291–323. Kluwer Academic Publishers, Dordrecht (2002)

Multi-Agent System Theory for Modelling a Home Automation System

G. Morganti, A.M. Perdon, G. Conte, and D. Scaradozzi

Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione
Università Politecnica delle Marche
Via Brecce Bianche, 12
60131 Ancona, Italy
morganti@diiga.univpm.it, {perdon,gconte,d.scaradozzi}@univpm.it

Abstract. A paradigm for modelling and analysing Home Automation Systems is introduced, based on Multi-Agent System theory. A rich and versatile environment for Home Automation System simulation is constructed for investigating the performances of the system. The exploitation of limited resources (electricity, gas, hot water) depends on behavioural parameters of the individual appliances. In order to deal with the problem of developing systematic design and validation procedures for control strategies, global performance indices for the system are introduced. Different strategies for allocating resources and establishing priorities in their use can therefore be tested and compared.

1 Introduction

The aim of this paper is to present an approach to the study of Home Automation systems which, based on the Multi-Agent System (MAS) theory allows to introduce a rigorous formalisation and to define practical tools for the analysis of performances.

Home Automation Systems (HAS) consist of a number of appliances (e.g. washing machine, dishwasher, oven, refrigerator, boiler) and other devices or subsystems (e.g. for heating, air conditioning, entertainment), which are connected by a communication network of some kind. The various elements of the system share common resources (mainly electricity, but also water) to accomplish their specific tasks, according to user requirements. Since resources are limited (due to costs, contract rules or provider directives), competition between the elements of the system generates internal conflicts. Distributed or centralised control strategies have to be implemented within the system, so to allocate limited resources while maximising performances, using the information dispatched over the communication network and the computational resources the devices, or a dedicated unit of them, may have [1].

From a control theory point of view, Home Automation Systems are very complex objects, characterised by the presence of components and constraints of different kinds, a distributed structure and a hybrid event/time-driven behaviour. Although the market offers devices and solutions that allow to assemble more or

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 585–593, 2009.
© Springer-Verlag Berlin Heidelberg 2009

less sophisticated Home Automation Systems, efficient design techniques are far from being developed. In particular, general methods for evaluating the performances and for tuning the control strategies are missing. On the other hand, the demand for methods and tools that facilitate the systematic design of efficient Home Automation Systems is made more and more urgent by the increase of the quantity of limited resources that are consumed for domestic use and by the growing competition in the market of appliances and home devices [2].

Here, we present and discuss a specific contribution to the construction of tools for the analysis and design of Home Automation Systems. We start by recalling the notion of Home Automation System based on the Multi-Agent System theory, that was introduced and discussed in [3,4,5,6]. On the formal basis that notion provides, we construct an environment for the simulation and emulation of Home Automation Systems. At the same time, we define indices of performances for paradigmatic control strategies, characterised by suitable parameters, that, by solving conflicts between individual components of the system, allocate limited resources. The outcome of this work is a rich and versatile framework in which one can analyse the effects caused on the performance indices by variations in the parameter that characterise the control strategies. In other term, control strategies for Home Automation Systems can be validated and tuned by exploiting the results of simulation, so to increase user satisfaction and system efficiency.

2 Multi-Agent System and Home Automation System Definition

Multi-Agent System theory has been employed to model and study several situations, characterised by the presence of autonomous agents which interact and possibly compete in various way, in many areas of computer science, automation and robotics (see e.g. [3,4,7] and the references therein).

Another application is the IHome Environment [8], an example of intelligent home that uses the Multi-Agent System technology as a way to control the behaviour of house appliances from the resource consumption and coordination viewpoint. The simulated IHome environment is controlled by intelligent agents that are associated with particular appliances. In the MavHome project [9] the smart home is seen as an intelligent agent that perceives its environment through the use of sensors, and can act upon the environment through the use of actuators. The home has certain overall goals, such as minimising the cost of maintaining the home and maximising the comfort of its inhabitants. C@sa [10] is a Multi-Agent system aiming at modelling, controlling and simulating house behaviour according to user and context features, using several "Operator Agent" and a "Supervisor Agent". Another notable environment is Adaptive House [11], that focuses on the development of a home that programs itself by observing the lifestyle and desires of the inhabitants, and learns to anticipate and accommodate their needs. Although Multi-Agent System theory is not used in this work, this system is interesting since the control is handled by neural network reinforcement learning and prediction techniques.

Anyway, in all these works, the resource management strategies can neither be fine-tuned nor evaluated by means of well-defined performance indices. Moreover, the constraints on resource consumption are quite generic and not as “hard” as in the present work. Our approach is based on the use of very special agents, that are not to be seen as computer science agents; we provide an appropriate definition of “Domotic Agent”, that stems from the more general definition of agent as an entity who is capable of action.

2.1 Home Automation Systems as Multi-Agent Systems

As we have already mentioned, Home Automation Systems consist of appliances and other devices or subsystems, connected by a communication network. Communication may happen through the power line, by means of suitable devices called nodes.

From the point of view we have adopted, we can consider the human user himself as an agent of the Home Automation System. Competing for the resources, his needs will conflict with those of other agents in the same structured environment and the solution of conflicts will follow the same general rules. Privileges and priorities granted to the human user do not represent external disturbances which may interfere with system behaviour and degrade its performances, but are integrated in the laws that govern its operations.

In a real Home Automation System, the agents must have some individual features that facilitate their integration into a larger system, without reducing their ability to work as stand-alone devices and to satisfy the user. Summarising, the features can be informally described as follows:

- *Autonomy:* the capability to perform their own tasks, when required, without support from the Home Automation System or, alternatively, the capability to negotiate resources in interacting with other agents
- *Collaborative behaviour:* the capability to cooperate with other agents in order to share resources and to reach common goals
- *Adaptability:* the capability to modify their behaviour according to the outcome of interaction processes
- *Reactivity:* the capability to react, to some degree, to system actions.

The above features assure the possibility to work sharing common resources, and suitable strategies for optimising individual performances are applied [6].

Multi-Agent approach is very suited to deal with such a situation and allows to model the structure of a general Home Automation System at least at two levels: a global level concerns the overall behaviour and performances of the system, and a local level that concerns the way in which single components and devices are integrated into the system and work. In other words, in a Home Automation System we have a collection of components, which can be generically called agents, and an overall architecture, which defines the environment in which the agents interact and the modalities of interaction. From this point of view, a Home Automation System qualifies as a Multi-Agent System in the sense of

Sycara [3], in which, in particular, there is no centralised, global controller or supervisor; data and knowledge are decentralised; computing power is distributed and asynchronous; each agent has incomplete information or capabilities and, as a consequence, limited knowledge and awareness of the overall situation.

In order to allow the system to operate, the architecture of the system and the way in which information flows in it must guarantee that each agent may declare itself, may detect and possibly recognise other agents and may interact with other agents according to given rules (system rules).

2.2 Domotic Agent

Following the Multi-Agent System theory point of view, an agent, namely the basic component of a larger system, can be defined by characterising its capabilities. Keeping in mind the application we are interested in, a suitable way to characterise an agent is provided by the following definitions.

Definition 1. An *agent* is a virtual or physical entity that:

1. is able to perform specific actions in a given environment;
2. is able to perceive elements of the environment;
3. is able to construct (partial) models of the environment
4. is able to use personal resources and environmental resources;
5. is able to direct its actions toward specific goals;
6. is able to communicate directly with other agents in the environment;
7. is able to offer services to other agents;
8. is able to govern its action according to its possibilities and limitations, to its goals, to its knowledge of the environment, to the resources available.

Remark 1. This definition is formally analogous to the definition of agent in Ferber [7]. It should be noted, as stated in Russell and Norvig [12], that in general the definition of agent "is meant to be a tool for analyzing systems, not an absolute characterization". For a more extensive discussion of the issues involved in the general definition of agent, the reader is referred to [13].

Now, we can give the notion of *domotic object* and of *domotic agent*, the elementary component of a Home Automation System, by specialising Def. 1.

Definition 2. A *domotic object* is a generic agent in the sense of Definition 1 that has at least the general capacities 1, 4, 5 and 8 and, concerning capacity 6, it is able to communicate to other agents in the environments at least its requirements about environmental resources.

Definition 3. A *domotic agent* is a domotic object that, in addition, has at least the general capacity 2. A *cognitive domotic agent* has also capacity 3.

Remark 2. Note that in this framework "cognitive" is not meant to define faculty of cognition in general, but with respect to the object of our studies.

The above definitions are quite abstract, but, as described in [5], they fit well with the appliances and other devices found in a domestic environment.

2.3 Home Automation System

After having defined the main elements, namely the domotic object and agents, that will form the overall system, we can give the following Definition.

Definition 4. A *Home Automation System (HAS)* consists of:

1. a set GR of Global Resources;
2. a set DO of Domotic Objects;
3. a set DA of Domotic Agents, subset of DO;
4. one Information Network IN, that connects domotic objects and agents;
5. a set R of Rules that govern the individual behaviour and the concurrent operation of domotic objects concerning use and transformation of external resources, communication, perception and understanding;
6. a set L of operators, called Laws, that describe the time evolution of the global system according to the individual behaviour of objects and of agents.

The notions described must be viewed as conceptual instruments that facilitate modelling and analysis of concrete examples, help in understanding their features and in identifying their critical parameters. Remark that, in facts, the time evolution of a Home Automation System formally described on the basis of Def. 4 is completely determined by L and it depends, in particular, on the Rules which form R. Then, developing simulation tools and procedures as described in the next Section, it is possible to investigate the effects of different choices of the Rules on the global evolution and behaviour of the system and to evaluate the resulting performances in terms of indices that represent user satisfaction.

3 Simulation Environment

In the virtual environment we construct, agents are characterised in a realistic way, by exploiting the experience in studying and modelling white goods derived from the collaboration with Italian manufacturers. Besides making simulation more effective, this choice is crucial to allow the simulation engine to interface with real appliances. Anyway, since the duration of an appliance cycle may be quite long, one of the important feature of the simulator is that it can work only with virtual agents, whose execution speed can be increased setting a specific parameter, so to shorten the duration of simulation.

The behaviour of each agent in the simulator is modelled as a sequence of transitions from state to state. The sequence of state transitions during normal operation of each agent is time-driven, except in case a required resource is or becomes unavailable. In such case, this event may cause a transition.

Basically, the simulator consists of a software environment, developed using LabVIEW, where programs (virtual instruments) that represent the single agents are executed simultaneously; each program implements the sequence of state transitions that characterises an agent. Agents exchange information by sharing global variables. To run a simulation, one has to define the virtual environment by indicating the agents and the related global variables and by specifying the values of a number of global and individual (i.e. concerning a single agent) parameters. During the simulation, agents are put into action automatically, according to a chosen schedule, so to simulate normal operations in a domestic environment. The set of agents in the Home Automation System is composed of the following: washing machine, dishwasher, boiler, human user and metering devices. Other agents can easily be added, if necessary, such as oven, vacuum cleaner, etc.

3.1 Control Strategies

Allocation of limited resources is the key problem that the Home Automation System we consider has to deal with, by solving conflict between competing agents. Here, we assume that limitation of resources means that the actual consumption must not exceed a fixed threshold for a limited time T_{lim}. The need of a control strategy for electricity consumption is particularly perceived in typical Italian home installations, where the contract with the provider determines expressly a 3 kW threshold and T_{lim} may vary around a few minutes; consumption over the threshold for a time greater than T_{lim} causes a blackout. In this case the user has to switch off some appliance to go back under the threshold and put the master switch on again; it is undoubtedly an unwanted event, not least because sometimes the master switch is placed outside home. Other possible goals could be minimise the cost, if for instance different rates for resources apply depending on peak or off-peak hours.

Denoting by overload the condition in which the threshold is surpassed and analysing available resources and their quantitative limits, we have found convenient to classify the resources in two groups:

1. resources for which overload can persist, without causing degradation in quality, for a (short) time $T_{lim} > 0$, after which, if overload persists, its use becomes unacceptably expensive, or, alternatively, the resource is cut off, making necessary to intervene for restarting the system;
2. resources for which overload could persist indefinitely, without causing cut off and the necessity of restarting the system, but whose persistence in time degrades relevantly the quality, so that it is appropriate to chose $T_{lim} = 0$.

Two paradigmatic examples are that of electricity, for which T_{lim} is greater than 0 and that of hot water, for which T_{lim} is ideally 0, since overloads would cause an abrupt decrease of temperature and/or flow.

Resources belonging to different groups have to be managed with different strategies. Overall control strategies for managing first group resources can be designed following the lines of the Power Levelling strategy introduced and discussed in [1,5]. This means that the action of each home automation agent is determined by two parameters, respectively the *Overload Time* t_o and the *Suspension Time* t_s. The Overload Time t_o represents the time the agent waits before stopping its action in case an overload occurs. The Suspension Time t_s represents the time the agent waits before restarting its action after having stopped it because of an overload.

Overall control strategies for managing second group resources can be designed following the lines of the Water Levelling strategy. In case overload occurs when more than one agent at a time tries to use the resource, planning and evaluation turn out to be particularly simple. To do so, in fact, an agent has only to be able to check, before using the resource, if others are already doing so and, in case, to renounce. This kind of strategy assures the highest priority to the agent which first gets the resource and forces the others to wait.

3.2 Performance Indices

In order to compare the results of different simulation scenarios, determining thereby which strategies and policies are the best, we need one or possibly more indices of performance. We have defined two indices, that point at important aspects: relative delay in completion of tasks and number of overloads. Other choices are possible if different problems have to be investigated.

Relative Delay. Let the duration of task i, when all the required resources are available, be indicated by TN_i and call it nominal duration. In case of conflict and competition, during a simulation, the time needed to complete the task may take a higher value $TA_i \geq TN_i$. Then, we consider the relative delay

$$\Delta_i = \frac{TA_i - TN_i}{TN_i} \tag{1}$$

and, letting N be the number of tasks in the considered scenario, we define the global percent relative delay $\Delta\%$ as

$$\Delta\% = 100 \cdot \frac{\sum_{i=1}^{N} p_i \Delta_i}{N} \tag{2}$$

where the p_i are weights that can be chosen according to the characteristics of the task. Clearly, user satisfaction increases when this index approaches 0. Nominal durations have been determined using real appliances.

Number of Overloads. This index simply counts the occurrences of overloads during the simulation and it is denoted by OL. Overloads stress the system and put it at risk or cause a great increase of costs, so the Home Automation System is performing better if OL is kept small.

3.3 Parameter Optimisation

To optimise the choice of parameters in order to achieve better performances is the object of a relatively new research area, the simulation optimisation.

The capabilities of the simulator have been tested on several experiments on a simple Home Automation System, using the Power Levelling strategy with different parameters, and different parameter optimisation procedures have been implemented: a Tabu Search and a Genetic Algorithm [14,15].

As an example, we can mention the simulation results obtained using the Tabu Search approach, without entering into the optimisation method details, for which the reader is referred to [14].

The parameters that we want to optimise are the Overload Time and the Suspension Time for both the washing machine (WM) and the dishwasher (DW). As first starting solution, we decided to choose a situation of equilibrium with the same parameter values for both the appliances, i.e. $t_o = 50$ s and $t_s = 1000$ s. Running the simulation with these parameters resulted in 13 overloads and 287.1% as relative delay; clearly it was a very bad solution: the great number of overloads stresses the whole system and the accomplishment of the tasks requires a duration that is almost four times as the expected one.

Applying the optimisation method, the best solution (WM: $t_o = 30$ s, $t_s = 1100$ s; DW: $t_o = 90$ s, $t_s = 700$ s) was found after 14 iterations and it resulted in 2 overloads and 22.6% of relative delay.

The performance of this solution is significantly better than the starting solution, so we can say that our framework has been very effective in finding a good set of parameters starting from a set chosen with no knowledge or prediction about the relationship between parameters and performance index, which was a very poor solution.

4 Conclusions and Future Directions

Starting from the formal definitions of basic concepts in home automation we have shown how a Multi-Agent approach has been used to develop a coherent, general framework for modelling, simulating and analysing Home Automation Systems. From a conceptual point of view, a key feature of this approach is the possibility of defining indices of performance, that turn out to be functional in designing and optimising global control strategies and behavioural policies.

The simulator has proven to be a useful tool that allows to adopt the most effective strategy for each scenario; when appliances are added or discarded from the house or the restrictions on the resources change, a new simulation optimisation can be easily performed.

Actually we are investigating metaheuristic methods, such as genetic algorithms, for parameter optimisation as well as other simulation optimisation methods in order to design control strategies for energy saving and optimal resource exploitation. Another promising direction could be the use of Artificial Neural Networks to determine a set of resource management parameters that optimise the performance indices.

References

1. Cipriani, M.: Bringing new services and benefits in a changing home environment. In: LonWorld (2001)
2. Vishwanathan, V., McCalley, J., Honavar, V.: A multi agent system infrastructure and negotiation framework for electric power systems. In: Power Tech Proceedings (2001)

3. Sycara, K.: Multiagent systems. AI Magazine 10(2), 79–93 (1998)
4. Flores-Mendez, R.A.: Towards a standardization of multi-agent system framework. Crossroads 5(4), 18–24 (1999)
5. Conte, G., Scaradozzi, D.: Viewing home automation systems as multiple agents systems. In: Multi-agent system for industrial and service robotics applications, RoboCUP 2003 (2003)
6. Scaradozzi, D.: Methodologies and Techniques for Analysis and Design of Home Automation Systems. PhD thesis, Università Politecnica delle Marche (2005)
7. Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
8. Lesser, V., Atighetchi, M., Benyo, B., Horling, B., Raja, A., Vincent, R., Wagner, T., Ping, X., Zhang, S.X.: The intelligent home testbed. In: Proceedings of the Autonomy Control Software Workshop (Autonomous Agent Workshop), Seattle (January 1999)
9. Rao, S.P., Cook, D.J.: Predicting inhabitant actions using action and task models with application to smart homes. International Journal of Artificial Intelligence Tools 13, 81–100 (2004)
10. De Carolis, B., Cozzolongo, G., Pizzutilo, S., Plantamura, V.L.: Agent-based home simulation and control. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS, vol. 3488, pp. 404–412. Springer, Heidelberg (2005)
11. Mozer, M.C.: Lessons from an adaptive house. In: Cook, D., Das, S. (eds.) Smart environments: Technologies, protocols, and applications. J. Wiley & Sons, Chichester (2004)
12. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs (1995)
13. Franklin, S., Graesser, A.: Is it an agent, or just a program?: A taxonomy for autonomous agents. In: Jennings, N.R., Wooldridge, M.J., Müller, J.P. (eds.) ECAI-WS 1996 and ATAL 1996. LNCS, vol. 1193. Springer, Heidelberg (1997)
14. Conte, G., Scaradozzi, D., Perdon, A., Morganti, G.: Parameter tuning in distributed home automation systems: towards a tabu search approach. In: MED 2008: 16th IEEE Mediterranean Conference on Control and Automation, Ajaccio, France (2008)
15. Morganti, G., Perdon, A., Conte, G., Scaradozzi, D., Brintrup, A.: Optimising home automation systems: a comparative study on tabu search and an evolutionary multi-objective algorithm. In: MED 2009: 17th IEEE Mediterranean Conference on Control and Automation, Thessaloniki, Greece (2009)

THOMAS-MALL: A Multiagent System for Shopping and Guidance in Malls

S. Rodríguez[1], A. Fernández[2], V. Julián[3], J.M. Corchado[1], S. Ossowski[2],
and V. Botti[3]

[1] Universidad de Salamanca
[2] Universidad Rey Juan Carlos de Madrid
[3] Universidad Politécnica de Valencia, (Spain)

Abstract. This article presents a case study in which the THOMAS architecture is applied in order to obtain a multi-agent system (MAS) that can provide recommendations and guidance in a shopping mall. THOMAS is made up of a group of related modules that are well-suited for developing systems in other highly volatile environments similar to a shopping mall. Because the development of this type of system is complex, it is essential to thoroughly analyze the intrinsic characteristics of typical environment applications, and to design all of the system components at a very high level of abstraction.

1 Introduction

This article presents a dependable solution for using a novel architecture in designing and building a system for guiding and advising users in a shopping mall. A shopping mall can be considered a dynamic and volatile environment in which shops change, promotions appear and disappear, and the products that are offered are continually changing. As such, a high level design with an abstract architecture is essential.

The architecture we used is THOMAS (*MeTHods, techniques and tools for Open Multi-Agent Systems*) [6][7]. THOMAS is a new architecture for open MAS and is made up of a group of related modules that are well-suited for developing systems in other highly volatile environments similar to a shopping mall. This design will use a high level of abstraction to determine which components are necessary for addressing all of the needs and characteristics of a shopping mall guidance system.

Artificial intelligence techniques have given way to new studies that allow, among other things, modeling the problem of a shopping mall in terms of agents and MAS. The shopping mall is turned into an intelligent environment where users are surrounded by these techniques, but do not need to adapt to them. One of the objectives of MAS is to build systems capable of autonomous and flexible decision-making, and that will cooperate with other systems within a "society" [5]. This "society" must consider characteristics such as distribution, continual evolution and flexibility, all of which allow the members (agents) of the society to enter and exit, to maintain a proper structural organization, and to be executed on different types of devices. All of these characteristics can be incorporated via the open MAS and virtual organization paradigm, which was conceived as a solution for the management, coordination and control of agent performance [8]. The organizations not only find

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 594–601, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the structural composition of agents (i.e., functions, relationships between roles) and their functional behavior (i.e., agent tasks, plans or services), but they also describe the performance rules for the agents, the dynamic entrance and exit of components, and the dynamic formation of groups of agents[3].

The goal of this study is to present a case study in which the THOMAS architecture is used to build an open MAS for guiding users through a shopping mall. We will propose an application for this architecture and will evaluate its appropriateness for developing an open MAS in a real environment. The first step of this research involves designing the components needed for addressing all the needs and characteristics of a shopping mall system. The design is based on the GORMAS (Guidelines for Organization-based Multi-Agent Systems) [1] methodology, which is specifically geared towards organizations.

This article is organized as follows: section 2 presents the principle characteristics of the architecture and methodologies used; section 3 indicates the MAS that was developed for the actual case study (the shopping mall), and highlights the characteristics provided by the type of architecture used for its development; and the final section presents some of the conclusions obtained by this research.

2 THOMAS Outline

THOMAS [6][7] is the name given to an abstract architecture for large scale, open multi-agent systems. It is based on a services oriented approach and primarily focuses on the design of virtual organizations.

The architecture is basically formed by a set of services that are modularly structured. THOMAS uses the FIPA[1] architecture, expanding its capabilities with respect to the design of the organization, while also expanding the services capacity. THOMAS has a module with the sole objective of managing organizations that have been introduced into the architecture, and incorporates a new definition of the FIPA Directory Facilitator that is capable of handling services in a much more elaborate way, following the service-oriented architecture directives.

THOMAS consists of three principle components: *Service Facilitator (SF), Organization Manager Service (OMS)* and *Platform Kernel (PK).*

The SF primarily provides a place where autonomous entities can register service descriptions as directory entries. The OMS component is primarily responsible for specifying and administrating its structural components (role, units and norms) and its execution components (participating agents and the roles they play, units that are active at each moment). In order to manage these components, OMS handles the following lists: *UnitList*: maintains the relationship between existing units and the immediately superior units (SuperUnit), objectives and types; *RoleList*: maintains the relationships between existing roles in each unit, which roles the unit inherits and what their attributes are (accessibility, position); *NormList*: maintains the relationship between the system rules; *EntityPlayList*: maintains the relationship between the units that register each agent as a member, as well as the role that they play in the unit. Each virtual unit in THOMAS is defined to represent the “world” for the system in

---

[1] http://www.fipa.org (*Foundation for Intelligent Physical Agents*)

which the agents participate by default. Additionally, the roles are defined in each unit. The roles represent the functionality that is necessary for obtaining the objective of each unit. The PK component directs the basic services on a multi-agent platform and incorporates mechanisms for transporting messages that facilitate the interaction among the various entities.

From a global perspective, the THOMAS architecture offers a total integration enabling agents to transparently offer and request services from other agents or entities, at the same time allowing external entities to interact with agents in the architecture by using the services provided.

The development of MAS is typically based on a design that focuses on each agent independently, and is geared towards each agent's structure and performance. This research presents a new focus in which the design is directed at the organizational aspects of the agents, establishing two descriptive levels: the organization and the agent [4]. The system we developed used the GORMAS [1] organizational methodology.

3 Case of Study: Tormes Shopping Mall

The case study application facilitates the interaction between the users (clients in the shopping mall), the store or sales information, and recreational activities (entertainment, events, attractions, etc.), and defines the services that can be requested or offered. We developed a wireless system capable of incorporating agents that provide orientation and recommendation functionalities to the user, and that can be applied not only in a shopping mall, but also in other similar environments such as a supermarket, an educational facility, medical or health care center, etc.[2]. The clients use the agents via their mobile devices and RFID (Radio Frequency Identification) [9]technology in order to consult the store directory, receive special offers and personalized promotions, and ask for recommendations to navigate through the mall or locate other clients. Clients can also use the mechanisms available to them to plan a particular route that allows them to better spend their time in the mall and receive personalized notices.

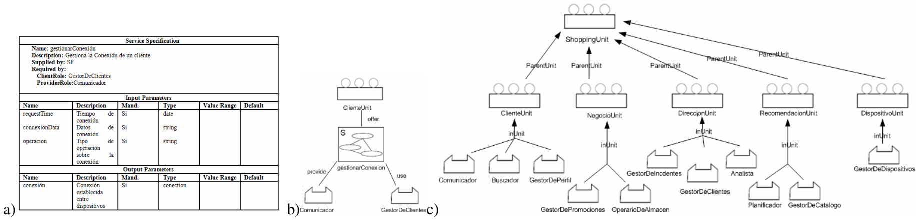
There are different types of agents that come into play: (i) the *User* agent, which is in charge of managing client communication, finding and identifying other user devices, and maintaining the user's profile; (ii) the *Shop* agent, which is in charge of maintaining the warehouse (i.e., product database) and the promotions that can be offered to the clients; (iii) and the *Guiding* agent, which is charge of managing user profiles, controlling communications, analyzing the promotions, managing all incidents, and most importantly, planning the best route for each user according to the available resources and the user profile. The *Guiding* agent can be considered the heart of the system, as it receives the most current information from each of the mall's stores, and interacts directly with the clients to offer personalized services.

The first step in analyzing and designing the problem is to define the following roles that will exist within the architecture: *Communicator*: in charge of managing the connections that each user makes. *Finder*: in charge of finding users with similar tastes. *Profile Manager*: in charge of creating and defining the client profile. *Promotions Manager*: in charge of suggesting promotions and offers. *Warehouse*

Operator: in charge of managing all inquiries made on the warehouse, managing updates and monitoring product shortages. *Analyst:* in charge of auditing sales information and the degree of client satisfaction. *Planner*: offers recommendation and guidance services to the shopping mall clients. This role is able to dynamically plan and replan in execution time. It suggests routes that clients might want to take through the mall, according to their individual preferences. *Client Manager*: in charge of managing the connections between the mall clients, managing the profiles for clients that are visiting the mall, monitoring the state of the clients, and managing the notification service for the mall. *Incident Manager*: manages and resolves incidents, offers a client location service, and manages an alarm system. *Directory Manager*: responsible for managing the mall's store directory, including businesses, products, promotions and offers. *Device Manager*: makes it possible for the interactive elements within the environment to interact. It deals with devices that use technologies such as RFID, etc.

We have also designed an organizational structure. We will first analyze its dimensions, and then proceed to identify the structure that is best suited to apply to the system [1]. Our case study is modeled as a conglomerate (*ShoppingUnit*) made up of five units, each one dedicated to one type of functionality within the setting. The five units are: (i)*ClientUnit*, contains the roles associated with the client: Communicator, Finder, and Profile Manager; (ii)*BusinessUnit*, contains the roles associated with a business: Promotions Manager, Warehouse Operator;(iii) *ManagingUnit*, contains the roles assigned with global management tasks for the mall: Incident Manager, Client Manager, and Analyst;(iv)*RecommendationUnit*, contains the roles dealing with recommendations or suggestions made to the client: Planner and Directory Manager;(v)*DeviceUnit*, which contains the roles associated with the management of devices: Device Manager.

The diagram in Figure 1c provides a structural view of the organizational model, which is adapted according to a conglomerate pattern. Different services are provided within each unit of the organization. The following section defines the services offered by the units, and uses an example to detail each one and how it has been modeled and described in the architecture. The type of role, the inputs and outputs, and a summary of the functionality for each unit are all explained. Figure 1b shows the internal model of the *ClientUnit*. The internal structure for *ShoppingUnit* and the remaining units was modeled in the same way.



Service Specification

Name: gestionarConexión
Description: Gestiona la Conexión de un cliente
Supplied by: SF
Required by:
ClientRole: GestorDeClientes
ProviderRole: Comunicador

Input Parameters

Name	**Description**	**Mand.**	**Type**	**Value Range**	**Default**
requestTime	Tiempo de conexión	Si	date		
connexionData	Datos de conexión	Si	string		
operacion	Tipo de operación sobre la conexión	Si	string		

Output Parameters

Name	**Description**	**Mand.**	**Type**	**Value Range**	**Default**
conexión	Conexión establecida entre dispositivos	Si	conection		

Fig. 1. a) *ManageConnection* service in *ClientUnit* b) Diagram of organization model: functional view of *ClientUnit* c) Diagram of organization model: structural view

One side of the diagram models the functional views of the units, which allows us to identify the services specific to each domain, while the other side precisely details the behavior of the organization services, how they interact with the environment, which interactions are established between the system entities, and how they approach the aspects of an open system. The next step is to define the rules in order to establish the control and management of the services. For example, the basic service provided by *ClientUnit* will be *ManageConnection*, which is provided by the agents that take on the role of Communicator. The functionalities offered by this service will allow the clients to control their connection to the system.

Similarly, within the *BusinessUnit* there are roles associated with a particular business and as a result, the services offered will be related to the corresponding promotions, products and sales (e.g., *SendPromotion* or *RetrievePromotion*). The services related to *ManagingUnit* involve the overall management tasks within a shopping mall (e.g., system incidents, data analysis, surveys, client management, notices, etc.). *RecommendationUnit* is comprised of services that request recommendations or suggestions based on user preferences and certain restrictions (time, money, etc.). It also includes planning and replanning the route that the user will follow based on the suggested recommendations, and determines the validity and value of the proposed routes. The *DeviceUnit* services deal with the sensors embedded in the physical system (RFID).

The type of services offered is controlled by the system according to the established norms [7]. The internal functionality of the services is responsible for the agents that are offered, but the system is what specifies the agent profiles, as well as the rules to follow for ordering requests or offering results. In this way, when faced with illicit or improper client performance, the system can act to impose sanctions. The OMS will internally save the list of norms that define the role involved, the content of the norms, and the roles in charge of ensuring that the norm is met. We have defined a set of norms in our system for controlling the performance within each unit. This way, for example, an agent within *ClientUnit* that acts like *Communicator* is required to register a service as *manageConnection*. If it does not abide by these norms, it will be punished and expelled from the unit. The punishment is logical given that if the agent does not establish a connection within the allocated time, it cannot perform any of the other system tasks. *OBLIGED Communicator REGISTER manageConnection(?requestTime, ?connectionData, ?operation) BEFORE deadline SANCTION (OBLIGED OMS SERVE Expulse (?agentID Comunicator ClienteUnit))*

Similarly, we have defined a complete set of norms that will control all of the system performances.

3.1 Example of Service Planning with THOMAS

The system considers the client objectives, the available time, and financial limitations, and proposes the optimal route according to the client's profile. The planning model we propose was integrated within a previously developed MAS [2]. We will see the series of steps that are taken within the system when a planning route is requested, and how THOMAS generates the system configuration that will give way to the plan. The first thing is to define the structural components of the organization, that is, the units that will be involved (which are initially empty), the system roles and norms. The indicated service requirements will be registered in the SF. To do so, either the basic OMS services for registering structural components

Table 1. SF: Basic services

Service Facilitator					
Entity	**Action**	**Service**	**ClientRole**	**ProvRole**	**Profile**
ClientUnit	Requires	manageConnection	ClientManager	Communicator	ClienteSP
DeviceUnit	Requires	locate	Communicator/IncidentManager	DeviceManager	DispositivoSP
…	…	…	…	…	…

will be used, or the API will directly execute the same functionality. This way, a community type *ShoppingUnit* will be created, representing the organization, whose purpose is to control the shopping mall. It has five internal unit planes: *ClientUnit*, *BusinessUnit*, *ManagingUnit*, *RecommentationUnit*, and *DeviceUnit*, each of which is dedicated to the functionalities we have previously seen. Each unit defines the existing roles, indicating their attributes (visibility, position, etc) and who they inherit them from.

The SF will announce basic services that are required for the overall system functionality. The basic services indicate which services are required (according to the defined norms) when creating the units.

From this moment on, the external agents can request the list of existing services and decide whether or not to enter and form part of the organization and with which roles. In our case we have clients that use their mobile device to send a request to the system so that it can inform them on the optimal route to take within the shopping mall. In order to carry out this function, we have, for example *Co1*, *Pe1* and *Pl1* acting as agents that will carry out the roles of *Communicator*, *ProfileManager* and *Planner* respectively. Agents *C1* and *C2* represent the clients that would like to receive a planning route.

Initially, all the agents head towards the THOMAS platform and are associated with the virtual "world" organization. As such, the OMS will play the *member* role in the "world" organization. When SF is asked about existing services in the system, the following response is obtained: `ClientUnit Requires manageConnection ClientRole =ClientManager;ProvRole=Comunicator;`

Because the service doesn't have an assigned *grounding*, it cannot be requested. But a functionality can be added, thus obtaining the *Communicator* role.

The *Co1* agent wants to offer that functionality, for which it requests receiving the *Communicator* role for the *ClientUnit*: `AcquireRole(ClientUnit, Communicator)`

If all goes well, the OMS will register *Co1* in the role of *Communicator* in *ClientUnit* within the *Entity Play List*. This list shows the roles that the different agents assume within THOMAS.

The *Co1* agent has carried out all of the regular steps for acquiring a role within THOMAS. This process is illustrated in Figure 2a where once Co1 has been registered as a member of the THOMAS platform, it asks SF which defined services have a profile similar to its own "communicator information service". This request is carried out using the SF SearchService (message 1), in which Communicator InformationServiceProfile corresponds to the profile of the *manage Connection* service implemented by *Co1*. The SF returns service identifiers that satisfy these search requirements together with a ranking value for each service (message 2). Ranking value indicates the degree of suitability between a service and a specified service purpose. Then *Co1* executes GetProfile (message 3) in order to obtain detailed information about the *manageConnection* service. Service outputs are "service goal" and "profile" (message 4). The *manageConnection* profile specifies that service providers have to play a *Communicator* role within *ClientUnit*. Thus, *Co1* requests the AcquireRole service from the OMS in order to acquire this provider role (message 5). AcquireRole service is carried out successfully (message 6), because *ClientUnit* is accessible from the virtual organization, thus *Co1* is registered as a *Communicator*.

There will be another inquiry regarding which services exist within the units. *DirectionUnit*, *RecommendationUnit* and *DeviceUnit* will return the services that are necessary for planning. The SF will again return a list (similar to Table1).

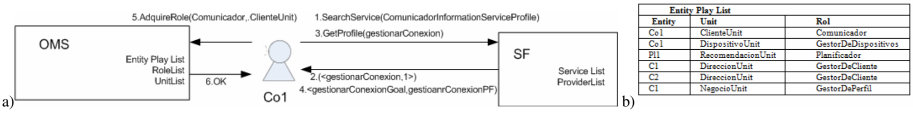
Based on the profiles, we will determine that *Co1* is interested in acquiring the role of *DeviceManager* since in this case it wants to interact with the elements within the environment. *Co1* will use this role to act as intermediary to process the signals that come from the client devices and make them comprehensible to the system. It will allow the order requested by the client from a mobile device to be understood and executed by the specific device that is the object of the request. (`AcquireRole(DeviceUnit, DeviceManager)).`

The agent will now be registered as a member of *DeviceUnit* with the role of *DeviceManager*. This role will require the agent to register the Locate service, associating it with the *process* and *grounding* that it considers to most useful. If this is not done within the allocated time, the agent will be expelled. The actual norm is as follows: `OBLIGED DeviceManager REGISTER Locate(?route) BEFORE deadline SANCTION (OBLIGED OMS SERVE Expulse (?agentID DeviceManager DeviceUnit))`

The agent will be informed of the norm upon carrying out the "AcquireRole", so that it can take it into consideration if it is a normative agent (otherwise ignore it). To avoid external agents assuming the role of *DeviceManager*, the agent registers a new incompatibility norm in the system. This norm will make it impossible for other agents to take on the same role: `RegisterNorm ("norm1", "FORBIDDEN Member REQUEST AcquireRole Message(CONTENT(role 'DeviceManager))")`

The *Pe1* and *Pl1* agents will act in a similar fashion, registering at the end for the corresponding units *ProfileManager* and *Planner*. They too will be required to register the services as indicated by the defined norms. (*GenerateProfile*, *ConsultProfile*, *UpdateProfile*, *MSState, UpdateMSGState, Replan, ValidateRoute, ValueRoute, ShopListRecovery*) Each one is required for generating the optimal route for the user to follow. The *C1* and *C2* agents will request acquiring the *ClientManager* role in order to access the basic services: *FindClient, GenerateProfile, ConsultProfile, UpdateProfile, MSGState, and UpdateMSGState*.

The agents will also consider whether to acquire other system roles that might be necessary for the required functionality. *C1* can request existing services from the SF, and will receive a list with all the agents that offer their services. In this case, for example, *C1* could be interested in offering the *SendPromotion* service as a suggesting sent to the user. These services are offered from the *BusinessUnit*, for which it is necessary to acquire the role of *ProfileManager* `(AcquireRole, (BusinessUnit, ProfileManager))`. The *Entity Play List* would end up as shown in figure 2b.


a)
5.AdquireRole(Comunicador,.ClienteUnit)
1.SearchService(ComunicadorInformationServiceProfile)
3.GetProfile(gestionarConexion)
OMS
Entity Play List
RoleList
UnitList
6.OK
Co1
2.(<gestionarConexion,1>)
4.<gestionarConexionGoal,gestioanrConexionPF)
SF
Service List
ProviderList
b)

Entity Play List

Entity	Unit	Rol
Co1	ClienteUnit	Comunicador
Co1	DispositivoUnit	GestorDeDispositivos
Pl1	RecomendacionUnit	Planificador
C1	DireccionUnit	GestorDeCliente
C2	DireccionUnit	GestorDeCliente
C1	NegocioUnit	GestorDePerfil

Fig. 2. a)Agent Co1 registering,b)Entity Play List

4 Conclusions

An important issue in the development of real open multi-agent systems is to provide developers with methods, tools and appropriate architectures which support all of the requirements of these kinds of systems. Traditional MAS development methodologies are not suitable for developing open MAS because they assume a fixed number of agents that are specified during the system analysis phase. It then becomes necessary to have an infrastructure that can use the concept of agent technology in the development process, and apply decomposition, abstraction and organization methods. We propose a methodology that incorporates decomposition and abstraction via the THOMAS architecture for a dynamic MAS environment. This architecture has allowed us to directly model the organization of a shopping center according to a previous basic analysis, to dynamically and openly define the agent roles, functionalities and restrictions, and to obtain beforehand the service management capabilities (discovery, directory, etc.) within the platform. THOMAS provides us with the level of abstraction necessary for the development of our system, and the set of tools that facilitate its development. In THOMAS architecture, agents can transparently offer and invoke services from other agents, virtual organizations or entities. Additionally, external entities can interact with agents through the use of the services offered. A case-study example was employed as an illustration of not only the usage of THOMAS components and services, but also of the dynamics of the applications to be developed with this architecture. In this way, examples of THOMAS service calls have been shown through several scenarios, along with the evolution of different dynamic virtual organizations.

References

1. Argente, E., Julian, V., Botti, V.: MAS Modelling based on Organizations. In: 9th Int. Workshop on Agent Oriented Software Engineering, pp. 1–12 (2008)
2. Bajo, J., Corchado, J.M., De Paz, Y., De Paz, J.F., Rodríguez, S., Martín, Q., Abraham, A.: SHOMAS: Intelligent Guidance and Suggestions in Shopping Centres. Applied Soft Computing (2008) ISSN 1568-4946
3. Bernon, C., Cossentino, M., Pavón, J.: Agent Oriented Software Engineering. The Knowledge Engineering Review 20(2), 99–116 (2005)
4. Boissier, O.: Organization Oriented Programming, from closed to open organizations. In: Engineering Societies in the Agents World VI. Springer, Ireland (2007)
5. Carbó, J., Molina, J.M., Dávila, J.: Fuzzy Referral based Cooperation in Social Networks of Agents. AI Communications 18(1), 1–13 (2005)
6. Carrascosa, C., Giret, A., Julian, V., Rebollo, M., Argente, E., Botti, V.: Service Oriented MAS: An open architecture. In: Actas del AAMAS 2009 (in press, 2009)
7. GTI-IA. An Abstract Architecture for Virtual Organizations: The THOMAS project, `http://www.fipa.org/docs/THOMASarchitecture.pdf`
8. Dignum, V., Dignum, F.: A landscape of agent systems for the real world. Technical report 44-cs-2006- 061, Institute of Information and Computing Sciences, Utrecht (2006)
9. Shekar, S., Nair, P., Helal, A.: iGrocer- A Ubiquitous and Pervasive Smart Grocery Shopping System. In: Proc. ACM SAC, pp. 645–652 (2003)

Multiagent-Based Educational Environment for Dependents

Antonia Macarro, Alberto Pedrero, and Juan A. Fraile

Pontifical University of Salamanca, Compañía 5, 37002 Salamanca, Spain
`{mmacarroal,apedreroes,jafraileni}@upsa.es`

Abstract. This paper presents a multiagent architecture that facilitates active learning in educational environments for dependents. The multiagent architecture incorporates agents that can be executed on mobile devices and that facilitate language learning in an ubiquitous way. Moreover, the agents are specifically designed to provide advanced interfaces for elderly and dependent people. The architecture has been tested in a real environment and the preliminary results obtained are presented within this paper.

Keywords: Ambient Intelligence, language learning, Multiagent systems, elderly people.

1 Introduction

Ambient Intelligence evolves from the ubiquitous computing [4], and constitutes the most promising technological approach to meet the challenge of developing strategies for early detection and prevention of problems in dependency environments [17]. One of the segments of the population who will benefit with the advent of systems based on Ambient Intelligence will be the elderly and people with disabilities [8], contributing to improve their quality of life [10].

Different authors [3] [9] [18] consider that in the near future, the institutions for health care will be provided with intelligent Systems capable to easily interact with the humans. It is necessary to improve the supply of services, as well as the way to offer them. Trends situate the user surrounded of technology that facilitates new and personalized services. Multiagent systems and architectures based on mobile devices, have recently been explored as a system of interaction with the elderly and dependent [1]. These systems can provide support in the daily lives of dependent people [20], providing a cognitive and physical support for the assisted person [2]. They can also provide mechanisms for establishing new strategies for learning interaction, which greatly facilitates the teaching, particularly languages learning.

The aim of this work is to provide a methodology, on the basis of the Ambient Intelligence (AmI) [5] [15], for formal teaching of languages oriented to dependent people. We propose the combination of new technologies along with the traditional teaching. In this way it will be possible to combine the advantages of the face to face teaching with the advantages of distance learning. It will be necessary to upgrade the systems of evaluation/accreditation to assess the knowledge or skills acquired during

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 602–609, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the learning process. To achieve this objective, we propose the use artificial intelligence techniques, intelligent agents and wireless communications.

The rest of the paper is structured as follows: Next section introduces the problem that motivates most of this research. Section 3 presents the multiagent architecture proposed to resolve the problem. Section 4 describes a case study to test the proposal and, finally, Section 5 presents the results and conclusions obtained.

2 Ambient Intelligence in Dependent Educational Environments

There is an ever growing need to supply constant care and support to the disabled and elderly and the drive to find more effective ways to provide such care has become a major challenge for the scientific community [4]. 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 [4]. In the USA, people over 65 years old are the fastest growing segment of the population [1] and it is expected that in 2020 they will represent about 1 of 6 citizens totaling 69 million by 2030. Furthermore, over 20% of people over 85 years old have a limited capacity for independent living, requiring continuous monitoring and daily care [2]. Some estimations of the World Health Organization show that in 2025 there will be more than 1000 million people aged over 60 in the world, so if this trend continues, by 2050 will be double, with about the 80% concentrated in developed countries [18].

Education is the cornerstone of any society and it is the base of most of the values and characteristics of that society. The new knowledge society offers significant opportunities for AmI applications, especially in the fields of education and learning [17]. The new communication technologies propose a new paradigm focused on integrating learning techniques based on active learning (learning by doing things, exchange of information with other users and the sharing of resources), with techniques based on passive learning (learning by seeing and hearing, Montessori, etc.) [7]. While the traditional paradigm, based on a model focused on face to face education, sets as fundamental teaching method the role of the teachers and their knowledge, the paradigm based on a learning model highlights the role of the students. In this second paradigm the students play an active role, and build, according to a personalized action plan, their own knowledge. Moreover, they can establish their own work rhythm and style. The active methodology proposes learning with all senses (sight, hearing, touch, smell and taste), learn through all possible methods (school, networking, etc.), and have access to knowledge without space or time restrictions (anywhere and at any time).

There are different studies that have used the Ambient Intelligence to facilitate learning. In [4], Bomsdorf shows the need to adapt intelligent environments to changes depending on the educational context and the characteristics of users. Morken *et al.* [18] analyze the characteristics of intelligent environments for learning. They focus on the role of mobility in educational environments and the role that acquire the mobile devices. Naismith *et al.* [19] conducted a detailed study describing the role of mobile devices in education, analyzing the characteristics of the devices and their capacity for learning in educational environments. All these approaches are focused on the role of learning in Ambient Intelligence environments, but none of them is oriented on learning for dependents or elderly people. The following section presents a multiagent architecture that facilitates learning methodology using an active through mobile devices.

3 Multiagent-Based Intelligent Environment for Language Education

Information technologies can notably improve the learning process in educational environments, providing the users with new technological resources [7]. At present there are various tools to facilitate the active learning, such as forums, wikis, email, chat, virtual campuses, and so on. However, none of them is focused to language learning for elderly people or dependents. This paper presents a multiagent architecture specially designed to facilitate active learning in dependence environments. Agents can be characterized through their capacities in areas such as autonomy, reactivity, pro-activity, social abilities, reasoning, learning and mobility [12] [21], which make them particularly suitable for use in the design and development of intelligent environments for dependency. That way, each student can have a personal agent, able to adapt to his personal needs taking into account the characteristics of his profile. In addition, each teacher has a personal agent that facilitates his interaction with students. The outline of the proposed architecture is presented in Figure 1.

In Figure 1 it is possible to see the four types of agents defined within the architecture: a student agent, a professor agent, an evaluator agent and a manager agent. The student and professor agents can be run on mobile devices, while the evaluator and manager agents run on a central computer. In the following paragraph, the main characteristics of the agent types are presented:

- Student Agent. It is a type of interface agent that allows a student to interact with the system. The student agents adapt to the needs of the human students, taking into consideration their profiles and customizing their access to the system.
- Professor Agent. It is a type of interface agent that allows a teacher to interact with the multiagent system. Through this agent a teacher is able to establish tasks for the students, and keep track of the personal development of the students
- Evaluator Agent. This agent is responsible for evaluating the work done by the students and allows progress on the level of difficulty. It is the agent that evaluates the level of learning acquired by the students. Moreover, it is in charged of statistics and reports periodically delivered to the teachers.
- Manager Agent. This agent is responsible for all management tasks within the system. These tasks include communications between the other agents in the multiagent system, assignation of tasks to students, or the creation of reports to be delivered to the professors. The allocation of tasks is based on past experiences, using a case-based reasoning system [9], so that the agent is able to learn and perform a more efficient assignation of tasks to students. Case-based reasoning is a technique in which past experiences are used to

solve new problems. This technique has been successfully used for the assignation of tasks on previous works focused on planning the working day of nurses and doctors in geriatric residences [5] [10] [11], or planning guidance routes in shopping malls [4].

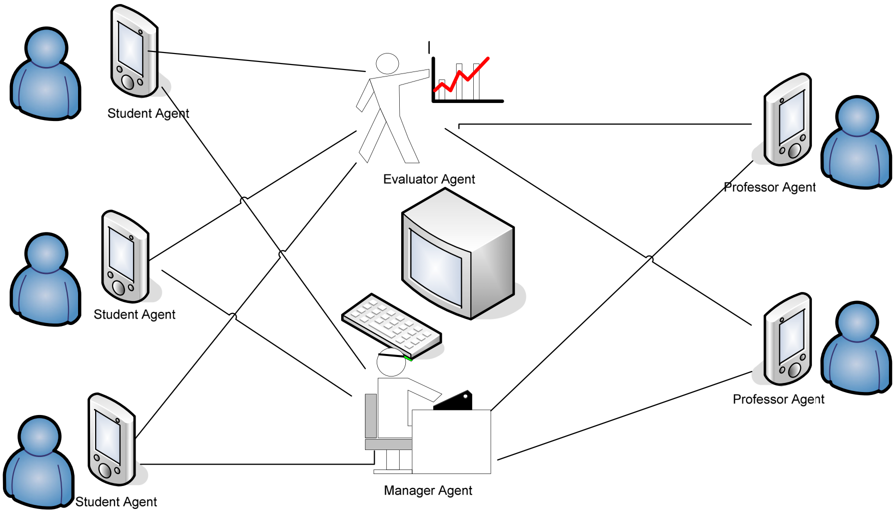


Fig. 1. Multiagent architecture for educational environments

The proposed architecture has great capacity for learning and adaptation to the user needs. Some works [14], which proposes an e-mail system adapted to elderly people without experience in the use of computers, show that not all users have the same difficulties. Even when an application was developed specifically for them, may require different versions for different users depending on their characteristics. The customization of the tasks and the interface can solve this problem because it allows different versions depending on the needs and particularities of each user [18].

One of the advantages of the architecture proposed in this paper is the ability of the agents to communicate through wireless channels, which facilitate independence of location or time constraints. The professor and student agents can be executed in mobile devices, and communicate using Wi-Fi, Bluetooth, SMS or MMS technologies. In this way, the students can use the learning system at their homes regardless of timetables. Access to services is independent of the terminal used and the use of services is simple and homogeneous, allowing a rapid assimilation by the user. In addition, users can receive fully customized services.

4 Case Study: Learning Languages at the Universidad de la Experiencia

Concern about the education of elderly people has led the universities to set up, under different names, studies for elderly people. These initiatives began to be put into practice in Spain in the early 90's, and nowadays all the Spanish universities have an educational program for elderly. In general, elderly people are inexperienced in the use of the Internet, finding problems to understand and use some of the conventions used in user interfaces: double-clicking with the mouse, movements between text fields to be filled in, bar displacement [16]. Moreover, the problems are common vision [15], cognitive problems [10], manual dexterity [20]. In order to benefit this group from the advantages of the digital environment, it is necessary to provide new methods and techniques. The usability is a key factor, given that a difficult-to-use system will be probably rejected by the users, increasing the fear and, perhaps, a sense of frustration in the use of these technologies. The work of Marqui and Todman [18] presents the difficulties and negative experiences of the elderly people in the use of computers (mainly the anxiety it causes).

Among the agencies that investigate new proposals to improve the daily life of elderly and dependent people, we can find the European Educational Programs Autonomous Agency, whose mission is to manage the Spanish participation in the Lifelong Learning Program (PAP) of the European Union. Within the adult education program is included the Grundtvig program. Encourage learning throughout life has become a political priority for all European countries. Language training for dependents is common and widely accepted. In this way, under the frame of the Grundtvig program, we defined a case study at the University of the Experience at the Pontifical University of Salamanca. To encourage the active learning and the autonomy of the students, we designed a series of tests that can be completed in a simple way through mobile phones. The case study, that involved 20 individuals, allowed us to evaluate the proposed methodology, and more concretely the improvement of the knowledge acquired in French language learning. Specifically, we developed questions with multiple options (A B C D) presented in groups of 10 to each student. If the success obtained exceeded the 80%, the student agreed to a higher level of difficulty. Moreover, at the same time the user chooses the option at the mobile phone, a microphone system records the phonetic. If the pronunciation is correct, the system recognizes the improvement and the user can advance to the next phase. In order to highly facilitate the user interaction with the multiagent system, we developed simple and intuitive interfaces. 5 tests were conducted at each of the students. Each test had a maximum duration of 60 minutes.

5 Results and Conclusions

This paper has presented a methodology for teaching languages to dependents, using techniques of Ambient Intelligence. The new multi-architecture presented in this paper provides an active learning through simple and intuitive interfaces, installed on mobile devices. This requires the integration of intelligent agents with innovative strategies of teaching languages and mobile devices. In this way we have obtained:

- An active method of teaching and learning of languages for dependents and elderly people.
- An interaction system based on the AmI paradigm for language education.

- The obtained language learning system was adapted to be executed on mobile devices, facilitating the adaptation to the needs of the dependent and elderly people.
- The multiagent architecture presented in this work has been tested by means of the case study presented in Section 4.

Table 1 shows the average number of hits in each of the five tests of the case study, and the average time to complete the 10 questions per test for each of the students at the University of Experience. As shown in Table 1, in the first test, the students had low rates of success and the time spent completing them was greater. As students learnt, the success rate increased and the time to complete tests decreased.

Figure 2 shows the results of the five tests performed in the case study presented in Section 4. Figure 2 shows the evolution of the successes of the students in the tests. As can be seen, the number of successes was lower in the early tests, and increased as new tests were performed. The horizontal axis represents the time (the tests), and the vertical axis the number of successes along the time. It is usual to find errors at the beginning of a new task. In the later stages the error rate decreased, but also the learned new subjects, reaching a plateau. The task of learning for the French language presented in the case study is steep at the beginning and then increasingly flat. This means that at the beginning made great progress, but after a while acquiring new knowledge is more difficult.

Table 1. Average success rate and time (in minutes) for the tests in the case study

	Success Rate	Time
Test 1	32	51
Test 2	43	47
Test 3	56	42
Test 4	77	38
Test 5	85	34

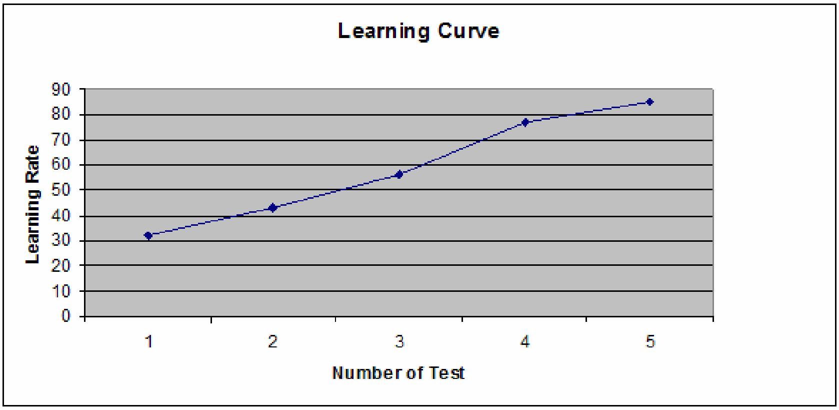


Fig. 2. Learning curve for the elderly students at the Universidad de la Experiencia. The curve shows the average learning rate for each of the five tests performed.

This paper has presented a multiagent architecture that facilitates languages learning to elderly people. The preliminary results obtained in a case study in a real scenario are promising, and let us conclude that our proposal can be useful in educational environments. However, there are still certain aspects to improve. Unless the users have remarked the usability of the system, it is necessary to design more efficient interfaces, facilitating the interaction to disabled people. Moreover, it is necessary to evaluate the architecture in different environments and for different users. Finally, it would be useful to take into account the possibility of creating agent based virtual organizations, facilitating the simulation of situations and enabling the detection of key aspects for optimum learning.

Acknowledgments. This work has been supported by the MOVENTE project, founded by Fundación Caja Duero.

References

1. Anastasopoulos, M., Niebuhr, D., Bartelt, C., Koch, J., Rausch, A.: Towards a Reference Middleware Architecture for Ambient Intelligence Systems. In: ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (2005)
2. Angulo, C., Tellez, R.: Distributed Intelligence for smart home appliances. Tendencias de la minería de datos en España. Red Española de Minería de Datos. Barcelona, España (2004)
3. Bajo, J., Molina, J.M., Corchado, J.M.: Ubiquitous computing for mobile environments. In: Issues in Multi-Agent Systems: The AgentCities.ES Experience. Whitestein Series in Software Agent Technologies and Autonomic Computing, pp. 33–58. Birkhäuser, Basel (2007)
4. Bajo, J., Corchado, J.M., de Paz, Y., de Paz, J.F., Rodríguez, S., Martín, A., Abraham, A.: SHOMAS: Intelligent Guidance and Suggestions in Shopping Centres. Applied Soft Computing 9(2), 851–862 (2009)
5. Bajo, J., de Paz, J.F., de Paz, Y., Corchado, J.M.: Integrating Case-based Planning and RPTW Neural Networks to Construct an Intelligent Environment for Health Care. Expert Systems with Applications 36(3) part 2, 5844–5858 (2009)
6. Bomsdorf, B.: Adaptation of Learning Spaces: Supporting Ubiquitous Learning in Higher Distance Education. In: Davies, N., Kirste, T., Schumann, H. (eds.) Dagstuhl Seminar Proceedings. Mobile Computing and Ambient Intelligence: The Challenge of Multimedia, Dagstuhl, Germany (2005)
7. Brown, T.H.: Beyond constructivism: Exploring future learning paradigms. In: Education Today, issue 2 of 2005, Aries Publishing Company, Thames, New Zealand (2005)
8. Carretero, N., Bermejo, A.B.: Inteligencia Ambiental. CEDITEC: Centro de Difusión de Tecnologías, Universidad Politécnica de Madrid, España (2005)
9. Corchado, J.M., Laza, R.: Constructing Deliberative Agents with Case-based Reasoning Technology. International Journal of Intelligent Systems 18(12), 1227–1241 (2003)
10. Corchado, J.M., Bajo, J., Abraham, A.: GERAmI: Improving the delivery of health care. IEEE Intelligent Systems 23(2), 19–25 (2008)
11. Corchado, J.M., Bajo, J., de Paz, Y., Tapia, D.: Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care. Decision Support Systems 34(2), 382–396 (2008)

12. Corchado, J.M., Glez-Bedia, J., de Paz, Y., Bajo, J., de Paz, J.F.: Replanning mechanism for deliberative agents in dynamic changing environments. Computational Intelligence 24(2), 77–107 (2008)
13. Czaja, S.J., Lee, C.C.: Designing computer systems for older adults. In: Jacko, J.A., Sears, J. (eds.) The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, pp. 413–427. Lawrence Erlbaum, Mahwah (2003)
14. Dickinson, A., Newell, A.F., Smith, M.J., Hill, R.L.: Introducing the Internet to the over-60s: developing an email system for older novice computer users. Interacting with Computers 17, 621–642 (2005)
15. Echt, K.V.: Designing web-based health information for older adults: visual considerations and design directives. In: Morrell, R.W. (ed.) Older adults, Health information and the World Wide Web, pp. 61–87. Lawrence Erlbaum Associates, Mahwah (2002)
16. Ellis, R.D., Kurniawan, S.H.: Increasing the usability of online information for older users: a case study in participatory design. International Journal of Human-Computer Interaction 2(12), 263–276 (2000)
17. Friedewald, M., Da Costa, O.: Science and Technology Roadmapping: Ambient Intelligence in Everyday Life (AmI@Life). Working Paper. Institute for Prospective Technology Studies IPTS, Seville (2003)
18. Kurniawan, S.H., King, A., Evans, D.G., Blenkhorn, P.L.: Personalising web page presentation for older people. Interacting with computers 18, 457–477 (2006)
19. Naismith, L., Lonsdale, P., Vavoula, G., Sharples, M.: Futurelab Literature Review in Mobile Technologies and Learning, Technical Report for Futurelab (2004), `http://www.futurelab.org.uk/research/reviews/reviews_11_and12/11_01.htm`
20. Ranganathan, V.K., Siemionow, V., Sahgal, V., Yue, G.H.: Effects of aging on hand function. Journal of the American Geriatrics Society 49, 1478–1484 (2001)
21. Wooldridge, M., Jennings, N.R.: Agent Theories, Architectures, and Languages: a Survey. In: Wooldridge, Jennings (eds.) Intelligent Agents, pp. 1–22. Springer, Heidelberg (1995)

Social and Cognitive System for Learning Negotiation Strategies with Incomplete Information

Amine Chohra, Arash Bahrammirzaee, and Kurosh Madani

Images, Signals, and Intelligent Systems Laboratory (LISSI / EA 3956), Paris-XII University,
Senart Institute of Technology, Avenue Pierre Point, 77127 Lieusaint, France
`{chohra,bahrammirzae,madani}@univ-paris12.fr`

Abstract. Finding adequate (*win-win* solutions for both parties) negotiation strategy with *incomplete* information for autonomous agents, even in one-to-one negotiation, is a complex problem. First part of this paper aims to develop negotiation strategies for autonomous agents with incomplete information, where negotiation behaviors, based on time-dependent behaviors, are suggested to be used in combination (inspired from empirical human negotiation research). Suggested combination allows agents to improve negotiation process in terms of agent utilities, round number to reach an agreement, and percentage of agreements. Second part aims to develop a social and cognitive system for learning negotiation strategies from interaction, where characters conciliatory, neutral, or aggressive, are suggested to be integrated in negotiation behaviors (inspired from research works aiming to analyze human behavior and those on social negotiation psychology). Suggested strategy displays ability to provide agents, through a basic buying strategy, with a first intelligence level in social and cognitive system to learn from interaction (human-agent or agent-agent).

1 Introduction

Negotiations have received wide attention from the distributed Artificial Intelligence (AI) community as a pervasive mechanism for distributed conflict resolution between intelligent computational agents [1]. In a context where agents must reach agreements (deals) on matters of mutual interest, *negotiation* techniques for reaching agreements (deals) are required. In general, any negotiation settings will have four different components [2]: - a negotiation set, - a protocol, - a collection of strategies, one for each agent, and - an agreement rule that determines the reach agreements stopping the negotiation. These four parameters lead to an extremely rich and complex environment for analysis. Another source of complexity in negotiation is agent number involved in process, and way in which these agents interact [2].

An interesting survey on negotiation models in AI field is given in [3], [4], [5]. Elsewhere, Lomuscio *et al.* [6] identified the main parameters on which any automated negotiation depends and provided a classification scheme for negotiation models. Instead of focusing on analyzing strategy equilibrium and historical information as in game theory, AI researchers are interested in designing flexible and sophisticated negotiation agents in complex environments (incomplete information).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 610–618, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Faratin *et al.* [7] devised a negotiation model that defines a range of strategies and behaviors for generating proposals based on time, resource, and behaviors of negotiators. Moreover, a type of combinations was earlier suggested in [7], and developed [8], [9], it concerns linear combinations of different behaviors. Then, several research works suggested negotiation strategies based on such type of combinations. Elsewhere, a second type of combination (inspired from empirical research on human negotiation) [10] consists of using different behaviors during a thread, i.e., a behavior during the first part and another one during the second part. In fact, a negotiator's demand level is the level of benefit to the self associated with the current offer. Concession rate is the speed at which demand level declines over time [11]. These studies support the conclusion [11], [12], [13]: negotiators who start with high demands and concede slowly often fail to reach agreement leading to inferior outcomes; those who start with low demands and concede rapidly usually reach agreement on the other party's terms yielding inferior outcomes; those between these extremes ordinarily achieve better outcomes.

By another way, in research works developed aiming to analyze and describe the human behavior in [14], twelve categories representing three major parts of the behavior have been defined: the positive socio-emotional part, a neutral task part, and the negative socio-emotional part. In another side, in research works on the social psychology of the negotiation of Rubin and Brown developed in [15], the interpersonal orientation of a person has an influence on his negotiating behavior.

Elsewhere, learning from interaction in negotiation is fundamental, from embodied cognitive science and understanding natural intelligence perspectives [16], [17], for understanding human behaviors [18]. Reinforcement learning is more focused on goal-directed learning from interaction than other approaches [19], [20], [21], offering advantages over classical dynamic programming [22]: they are *on-line* taking into account dynamics the real environment nature, and they can employ neural networks [23], [24], [25], to represent their knowledge, and to generalize.

In this paper, first, one-to-one bargaining process over single issue, is developed in Sect. 2. In this work, this process is based on time-dependent behaviors developed by Faratin *et al.* [7]. These behaviors, which can be used individually, are suggested to be used in *combination*. Afterwards, experimental environments and measures are detailed for these two strategy kinds in Sect. 3. Also, experimental results of different strategies, are presented and analyzed in Sect. 3. Second, a social and cognitive system is developed to interact with an integrated system of simulation for negotiation allowing to learn, from interaction, a negotiation strategy based on human personality behaviors in Sect. 4. Also, reinforcement learning (Q-learning) approach is developed in order to acquire the strategy negotiation behaviors in Sect. 4. Afterwards, a Fuzzy ArtMap Neural Network (FAMNN) is developed, in Sect. 4, to acquire this strategy.

2 One to One Negotiation

The negotiation set is the range of issues over which an agreement must be reached. Let i represents the negotiating agents, in bargaining negotiation $i \in \{\text{buyer}(b), \text{seller}(s)\}$, and j the issue under negotiation ($j = price$). For agent b, the proposal to offer or accept is within interval $[min^b, max^b]$, where max^b is the reservation price of b

in the negotiation thread, and min^b is lower bound of a valid offer. Similarly, for agent s, the proposal to offer or accept is within interval $[min^s, max^s]$, where min^s is the reservation price of s and max^s is upper bound of a valid offer. Thus, proposals $x^b[t]$ to be offered by b and $x^s[t]$ to be offered by s at time t, with $0 <= t <= t^i_{\max}$, are:

$$x^b[t] = min^b + \alpha^b(t)\,(max^b - min^b),\; x^s[t] = min^s + (1 - \alpha^s(t))\,(max^s - min^s),$$
$$\text{where } \alpha^b(t) \text{ and } \alpha^s(t) \text{ are ensuring that } 0 <= \alpha^i(t) <= 1, \tag{1}$$
$$\alpha^i(0) = K^i \text{ (positive constant) and } \alpha^i(t^i_{max}) = 1,$$

where $t^b_{\max}$ and $t^s_{\max}$ denote the negotiation deadline for agent b and agent s, respectively. Such $\alpha^i(t)$ functions can be defined in a wide range according to the way in which $\alpha^i(t)$ is computed by Eq. (2) defining: Boulware (B) with $\beta < 1$, Conceder (C) with $\beta > 1$, and Linear (L) with $\beta = 1$ [5].

$$\alpha^i(t) = K^i + (1 - K^i)\left(\frac{min(t, t^i_{max})}{t^i_{max}}\right)^{\frac{1}{\beta}}. \tag{2}$$

Concerning the strategies, this paper deals the work developed Lopes *et al.* [10] based on the second type of combination inspired from empirical human negotiation research as explained in Sect. 1. Indeed, Lopes *et al.* suggested a combination of opening negotiation behaviors (starting optimistic, realistic, or pessimistic) and concession behaviors, resulting in three strategy families (starting high and conceding slowly, starting reasonable and conceding moderately, and starting low and conceding rapidly). With regard to the work of Lopes *et al.* [10], in this paper, experiment results of strategies using behaviors individually are given in order to compare them with those of suggested strategies using behaviors in combination. More experimental measures are used as utility product and utility difference in order to finely evaluate the strategy results, and in order to perceive the improvement degree given using such combination. Thus, two kinds of strategies are considered for both agent b and agent s: - individual behaviors (one behavior for each strategy), where each strategy use individually the behaviors Boulware (B), Linear (L), or Conceder (C) during a negotiation *thread* ; and - behaviors in combination, where each strategy use two behaviors (among B, L, C) in combination during a negotiation *thread*, resulting in combinations: BL, BC, LB, LC, CB, and CL. Concerning agreement rule, agent b accepts an offer $x^s[t]$ from agent s at time t if it is not worse than the offer he would submit in the next step. Similarly, agent s accepts an offer $x^b[t]$ from agent b.

3 Experiments: Environments, Measures, and Results

Environments are defined in bargaining bilateral negotiation between buyer(b) and seller(s), in single issue j = *price*. The experimental environment is defined by the following variables $[t^b_{max}, t^s_{max}, T_{max}, K^b, K^s, min^b, max^b, min^s, max^s]$. The negotiation interval for *price* is defined using two variables as detailed in [7]: θ^i randomly selected between the ranges [10, 30] for both agents, and $\Phi = 0$.

Analysis and evaluation of negotiation strategies developed in [8], indicated that negotiation deadline significantly influences the performance. From this, the environment is defined, in short term deadlines (which the performance is more sensitive than long term deadlines), from random selection of the round number within [1, 5] corresponding to a random selection of t^i_{max} within [*2*, *10*]. Parameter β ranges are defined as: $\beta \in [0.01, 0.2]$ for B, $\beta = 1$ for L, and $\beta \in [20, 40]$ for C. Constants K^i are chosen as small positive $K^i = 0.1$, for both agents, in order to not constrain time-dependent behaviors. Also, initiator of an offer is randomly chosen [7].

To produce statistically meaningful results, for each experiment, the precise set of environments is sampled from the parameters specified in Sect. 3.1 and the number of environments used is $N = 200$, in each experiment. This ensures that the probability of the sampled mean deviating by more than *0.01* from the true mean is less than *0.05*.

To evaluate the strategy effectiveness the three following measures are considered:

- *Average Intrinsic Utility* for each agent is given by AU^i in Eq. (3), where N is the total number of environments in each experiment, U^i the utility of each agent, for each environment with deal, and N_D is the number of environments with deals. If there is no deal in an environment, then $U^b = U^s = 0$.

$$U^b = \frac{max^b - x}{max^b - min^b}\ ,\ U^s = \frac{x - max^s}{max^s - min^s}\ ,\ \text{and}\ AU^i = \frac{\sum_{n=1}^{N} U^i[n]}{N_D}\ . \tag{3}$$

- *Average Round Number AR* is given in Eq. (4), for each environment with deal:

$$AR = \frac{\sum_{n=1}^{N} R_D[n]}{N_D}\ ,\ \text{where } R_D \text{ is number of rounds.} \tag{4}$$

- *Percentage of Deals*, the percentage of deals *D(%)* is obtained, from the *Average Deal AD* given in Eq. (5), and also given in Eq. (5).

$$AD = \frac{N_D}{N}\ \text{ and }\ D(\%) = AD\,.100\,\%\ . \tag{5}$$

In order to analyze the performance of strategies, two measures are obtained [9]: - *utility product*, *UP*, of the utilities is computed as shown in Eq. (6) ; and - *utility difference*, *UD*, difference is computed as shown in Eq. (6).

$$UP = AU^b . AU^s \ \text{and}\ UD = \left| AU^b - AU^s \right|\ . \tag{6}$$

Results given in Table 1 are obtained from *9* experiments (B-B, B-L, B-C, L-B, L-L, L-C, C-B, C-L, C-C) where both agents b and s have same or different strategies (using behaviors individually) and where each experiment is an average of *N=200* environments. These results show that the adequate strategy when both agents have strategies using behaviors individually is L(buyer)-L(seller) case with the best (compromise of high *UP* and low *UD* and high *D(%)* and low *AR*) performance in term of utility product *UP*(high) about *0.2547* and utility difference *UD*(low) about *0.0451*, and percentage of deals *D(%)*(high)=*83%* and average round to reach an agreement *AR*(low)=*1.1807*.

Results given in Table 2, Table 3, and Table 4 are obtained from *36* experiments (BL-BL, BL-BC, BL-LB, BL-LC, BL-CB, BL-CL, and BC-BL, BC-BC, BC-LB, BC-LC, BC-CB, BC-CL, and LB-BL, LB-BC, LB-LB, LB-LC, LB-CB, LB-CL, and so on…) where both agents b and s have same or different strategies (using behaviors in combination) and where each experiment is an average of *N=200* environments.

Table 1. Results *AR*, *D(%)*, *UP*, *UD*, both agents have strategies using behaviors individually

	AR, D(%), UP, UD	*AR, D(%), UP, UD*	*AR, D(%), UP, UD*
Seller	B(Buyer)	L(Buyer)	C(Buyer)
B	1.4693, 49, 0.2508, 0.3826	1.2881, 59, 0.3193, 0.2390	0.5142, 70, 0.3364, 0.3482
L	1.3906, 64, 0.2333, 0.3474	1.1807, 83, 0.2547, 0.0451	0.5057, 87, 0.2571, 0.3915
C	0.3595, 89, 0.1511, 0.6861	0.3111, 90, 0.1684, 0.5844	0.2065, 92, 0.2520, 0.1390

Table 2. Results *AR*, *D(%)*, *UP*, and *UD*, agents with strategies using behaviors in combination

	AR, D(%)	*UP*	*UD*	*AR, D(%)*	*UP*	*UD*
Seller	BL(Buyer)			BC(Buyer)		
BL	2.71264, 87%	0.25436	0.00106	2.52747, 91%	0.26541	0.04873
BC	2.06382, 94%	0.27210	0.19270	2.07777, 90%	0.25150	0.21552
LB	2.43820, 89%	0.24643	0.06170	2.30681, 88%	0.25174	0.00780
LC	2.05154, 97%	0.23353	0.25296	1.94949, 99%	0.20627	0.40024
CB	0.54000, 100%	0.09391	0.79355	0.60000, 100%	0.10619	0.78095
CL	0.56000, 100%	0.08450	0.80261	0.55000, 100%	0.10600	0.77828

Table 3. Results *AR*, *D(%)*, *UP*, and *UD*, agents with strategies using behaviors in combination

	AR, D(%)	*UP*	*UD*	*AR, D(%)*	*UP*	*UD*
Seller	LB(Buyer)			LC(Buyer)		
BL	2.38823, 85%	0.28493	0.00764	2.31958, 97%	0.23666	0.14254
BC	2.23076, 91%	0.26701	0.01793	2.01075, 93%	0.24630	0.27879
LB	2.30232, 86%	0.25757	0.00986	2.01136, 88%	0.24515	0.12918
LC	1.94791, 96%	0.27118	0.09615	1.88888, 99%	0.25533	0.00050
CB	0.61000, 100%	0.17878	0.56262	0.59000, 100%	0.19029	0.54408
CL	0.48000, 100%	0.18538	0.57058	0.54000, 100%	0.18075	0.56877

Table 4. Results *AR*, *D(%)*, *UP*, and *UD*, agents with strategies using behaviors in combination

	AR, D(%)	*UP*	*UD*	*AR, D(%)*	*UP*	*UD*
Seller	CB(Buyer)			CL(Buyer)		
BL	1.12903, 93%	0.15593	0.61685	1.24175, 91%	0.16614	0.59882
BC	1.37078, 89%	0.17057	0.56780	1.29473, 95%	0.16629	0.61102
LB	0.93750, 96%	0.19787	0.48905	1.05208, 96%	0.21860	0.42831
LC	0.96907, 97%	0.20937	0.44588	0.96938, 98%	0.17765	0.48887
CB	0.27000, 100%	0.26951	0.05934	0.31000, 100%	0.24658	0.17449
CL	0.37000, 100%	0.22223	0.10801	0.24000, 100%	0.25171	0.13275

These results show that the adequate strategy when both agents have strategies using behaviors in combination is LC(buyer)-LC(seller) case with the best performance in term of *UP*(high)*=0.25533* and *UD*(low)*=0.00050*, and *D(%)(high)=99%*, and *AR*(low)*=1.88888*. From these results, since with incomplete information, an adequate strategy could be also here a strategy which imitates the other agent since: for *b*(BL) adequate strategy is *s*(BL) (*UP=0.25436*, *UD=0.00106*), for *b*(BC) adequate strategy is *s*(LB) (*UP=0.25174*, *UD=0.00780*), for *b*(LB) adequate strategy is *s*(BL), for *b*(LC) adequate strategy is *s*(LC), for *b*(CB) adequate strategy is *s*(CB), and for *b*(CL) adequate strategy is *s*(CL). From comparison of strategies using behaviors individually with those using them in combination, suggested combination allows agents to improve negotiation process in terms of agent utilities (utility product *UP* and utility difference *UD*), average round number to reach an agreement *AR*, and percentage of agreements *D(%)* as demonstrated by the following results: - from Table 1 using behaviors individually, L(buyer)-L(seller) case with the best performance in term of *UP=0.2547*, *UD=0.0451*, *D(%)=83%*, and *AR=1.1807* ; and - from Table 3 using behaviors in combination, LC(buyer)-LC(seller) case with best performance in term of *UP=0.25533*, *UD=0.00050*, *D(%)=99%*, and *AR=1.88888.*

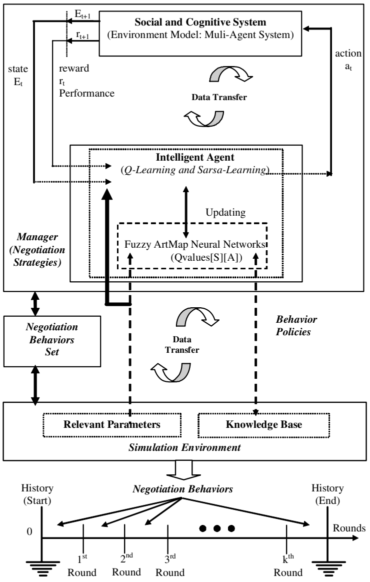
4 Social and Cognitive System for Learning Negotiation Strategies

The technology of multi-agent systems which facilitates the negotiation at operative level of the decision-making [26] is used. More, in order to satisfy their design objectives, agents are designed to be intelligent, i.e., capable of flexible behavior [2], [16], [17]: able to perceive their environment, and respond in a timely fashion to changes (reactivity), to exhibit goal-directed behavior (proactiveness), and capable of interacting with other agents and possibly humans (social ability) ; such abilities are guaranteed in this work by psychological aspects of the personality, multi-agents, and Q-Learning. In order to elaborate such negotiation strategy, social and cognitive system which is presented in Fig. 1, is developed. It is built of three main parts: a multi-agent system representing environment model, an intelligent agent, and a simulation environment. In this work, for an *intelligent* agent, given a state (round) and a given strategy (basic buying strategy), the agent reactions (choosing an Action in each state) is based on the updating of Qvalues (from reinforcement Q-Learning) corresponding to negotiation behaviors. The basic negotiation behaviors Conciliatory (Con), Neutral (Neu), and Aggressive (Agg) have been extended to nine such as:

- more Conciliatory (+Con), Conciliatory (Con), less Conciliatory (-Con),
- more Neutral (+Neu), Neutral (Neu), less Neutral (-Neu),
- less Aggressive (-Agg), Aggressive (Agg), more Aggresive (+Agg).

These behaviors correspond to the Actions such as: +Con (Action0), Con (Action1), -Con (Action2), +Neu (Action3), Neu (Action4), -Neu (Action5), -Agg (Action6), Agg (Action7), and +Agg (Action8). A strategy is then fixed using for instance the strategy rewards (for a basic buying strategy used for training) in Table 5. After learning, Action corresponding to character with the max Qvalue is chosen.

Fuzzy ArtMap Neural Network (FAMNN) implementation of reinforcement Q-learning offers the advantages of learning, generalization, and limited memory requirement for storing the knowledge [23], [24], [25]. Such learning can be achieved exploiting Adaptive Resonance Theory (ART) in combination with neural networks and fuzzy logic. Indeed, ART can be used to design hierarchical artmap neural



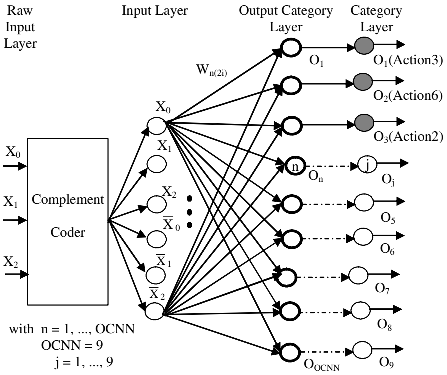


Fig. 1. Social and cognitive system

Fig. 2. Fuzzy ArtMap Neural Network architecture

Table 5. Strategy rewards (basic buying strategy)

	States From S0 to S8				States From S9 to S16				States From S17 to S24			
+ Con	- 60	- 60	…	- 60	- 60	- 60	…	- 60	- 2	- 2	…	- 2
Con	- 50	- 50	…	- 50	- 50	- 50	…	- 50	- 1	- 1	…	- 1
- Con	- 40	- 40	…	- 40	- 40	- 40	…	- 40	**+ 1**	**+ 1**	**…**	**+ 1**
+ Neu	**+ 1**	**+ 1**	**…**	**+ 1**	- 30	- 30	…	- 30	- 10	- 10	…	- 10
Neu	- 1	- 1	…	- 1	- 20	- 20	…	- 20	- 20	- 20	…	- 20
- Neu	- 2	- 2	…	- 2	- 10	- 10	…	- 10	- 30	- 30	…	- 30
- Agg	- 40	- 40	…	- 40	**+ 1**	**+ 1**	**…**	**+ 1**	- 40	- 40	…	- 40
Agg	- 50	- 50	…	- 50	- 1	- 1	…	- 1	- 50	- 50	…	- 50
+ Agg	- 60	- 60	…	- 60	- 2	- 2	…	- 2	- 60	- 60	…	- 60

networks that can *rapidly* self-organize *stable* categorical mappings between i-dimensional input vectors and n-dimensional output vectors [27]. FAMNN is then trained to acquire the suggested strategy, from learning algorithm detailed in [28], and from a training set of one hundred (*100*) examples. This classifier sprouted output category node number *OCNN=9* and yields convergence in well under cycle number cycle number *CN=1* with the learning rate $\eta=1$, the small positive constant $\lambda=0.000001$, and the vigilance baseline $\sigma=0.4$. After learning, FAMNN classifier is built of raw input layer (complement coder), input and output category, and category layers as illustrated in Fig. 2.

5 Discussion and Conclusion

In a first part, a *combination*, inspired from empirical human negotiation, of time-dependent behaviors has been suggested for autonomous agent systems with *incomplete* information intending to find adequate (*win-win* solutions for both parties) strategy, in one-to-one single issue negotiation. With regard to strategies using behaviors individually, obtained results demonstrate that the suggested combination allows agents to improve negotiation process in terms of agent utilities, round number to reach an agreement, and percentage of agreements. Another interesting conclusion is that, since with incomplete information, an adequate strategy could be a strategy which imitates the other agent (adequate strategies have been **BL-BL**, BC-LB, LB-BL, **LC-LC**, **CB-CB**, and **CL-CL**). With regard to work developed in [10]: - strategies of class starting reasonable and conceding moderately lead, on average, to superior outcomes (in our case: LB-BL and LC-LC) ; - strategies of class starting high and conceding slowly lead, on average, to fewer and slower agreements (in our case: BL-BL and BC-LB) ; and strategies of class starting low and conceding rapidly lead, on average, to more and faster agreements but with inferior outcomes (in our case: CB-CB and CL-CL). In a second part, a negotiation strategy using behaviors capturing psychological aspects of human personality has been suggested. Such negotiation behaviors are acquired by reinforcement Q-learning approach in order to guarantee agent reactivity, agent proactiveness, and agent social ability. The strategy is then acquired by FAMNN exploiting particularly their *fast* and *stable* learning capabilities (essential trait of intelligence). Thus, the suggested strategy displays ability to provide agents, through a basic buying strategy, with a first intelligence level in social and cognitive system for learning negotiation strategies from interaction (human-agent and agent-agent).

References

1. Rosenschein, J., Zlotkin, G.: Rules of Encounter. MIT Press, Cambridge (1994)
2. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons, England (2002)
3. Jennings, N.R., et al.: Automated negotiation: prospects, methods, and challenges. Int. J. of Group Decision and Negotiation 10(2), 199–215 (2001)
4. Gerding, E.H., van Bragt, D., Poutré, J.L.: Scientific Approaches and Techniques for Negotiation: A Game Theoretic and AI Perspective. Technical Report, SEN-R0005 (2000)
5. Li, C., Giampapa, J., Sycara, K.: Bilateral negotiation decisions with Uncertain Dynamic outside options. IEEE Trans. on Sys., Man, and Cyb. 36(1), 1–13 (2006)
6. Lomuscio, A.R., Wooldridge, M., Jennings, N.R.: A classification scheme for negotiation in electronic commerce. Int. J. of Group and Neg. 12(1), 31–56 (2003)
7. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Int. J. of Robotics and Autonomous Systems 24(3-4), 159–182 (1998)
8. Wang, K.-J., Chou, C.-H.: Evaluating NDF-based negotiation mechanism within an agent-based environment. Robotics and Autonomous Sys. 43, 1–27 (2003)
9. Ros, R., Sierra, C.: A negotiation meta strategy combining trade-off and concession moves. Auton Agent Multi-Agent Sys. 12, 163–181 (2006)
10. Lopes, F., et al.: A negotiation model for autonomous computational agents: Formal description and empirical evaluation. J. of Intellig. and Fuzzy Sys. 12, 195–212 (2002)
11. Pruitt, D.: Negotiation Behavior. Academic Press, London (1981)
12. Hamner, W.: Effects of bargaining strategy and pressure to reach agreement in a stalemated negotiation. Journal of Personality and Social Psychology 30(4), 458–467 (1974)

13. Carnevale, P., Pruitt, D.: Negotiation and Mediation. In: Rosenzweig, M., Porter, L. (eds.) Annual Review of Psychology, vol. 43, pp. 531–581. Annual Reviews Inc. (1992)
14. Bales, R.F.: Interaction Process Analysis: A Method for the Study of Small Groups. Add.-Wesley, Cambridge (1950)
15. Rubin, J.Z., Brown, B.R.: Social Psychology of Bargaining and Neg. Aca. Press, NY (1975)
16. Pfeifer, R., Scheier, C.: Understanding Intelligence. MIT Press, Cambridge (1999)
17. Chohra, A.: Embodied cognitive science, intelligent behavior control, machine learning, soft computing, and FPGA integration. Technical GMD Report, no. 136, Germany (June 2001)
18. Zeng, D., Sycara, K.: Benefits of learning in negotiation. In: Proc. of the 14th National Conference on Artificial Intelligence (AAAI 1997), Providence, RI (July 1997)
19. Watkins, C.J.C.H.: Learning from Delayed Rewards. PhD Thesis, King's College (1989)
20. Whitehead, S.D.: Reinforcement Learning for the Adaptive Control of Perception and Action. Technical Report 406, University of Rochester (February 1992)
21. Sutton, R.S., Barto, A.G.: Reinforcement Learning. MIT Press, Cambridge (1998)
22. Dietterich, T.G.: Hierarchical reinforcement learning with the MAXQ value function decomposition. Journal of Artificial Intelligence Research 13, 227–303 (2000)
23. Anderson, J.A.: An Introduction to Neural Networks. The MIT Press, England (1995)
24. Patterson, D.W.: Artificial Neural Networks: Theory and Applications. Prentice-Hall, Englewood Cliffs (1996)
25. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)
26. Sandholm, T.W.: Distributed Rational Decision Making, pp. 201–258. MIT Press, Cambridge (1999)
27. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: Fast stable learning and categorization of analog patterns by adaptive resonance system. NN 4, 759–771 (1991)
28. Azouaoui, O., Chohra, A.: Soft computing based classifiers for obstacle avoidance behavior of Intelligent Autonomous Vehicles. Int. J.of App. Intelligence. 16(3), 249–271 (2002)

Evaluation of Multi-Agent System Communication in INGENIAS

Celia Gutiérez, Iván García-Magariño, and Jorge J. Gómez-Sanz

Departamento Ingeniería de Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid
Ciudad Universitaria s/n,
28040, Madrid, Spain
`{cegutier,ivan_gmg}@fdi.ucm.es, jjgomez@sip.ucm.es`

Abstract. This paper proposes a corpus of metrics to evaluate the balance of communications in these systems. The hypothesis of this paper is that these metrics are strongly related with the quality of service of the MASs. In addition, some classification rules are provided to classify agents according to the metrics; thus, an origin of the low quality of service of MASs is detected. The detection of this origin is the first step for debugging and improving the communication in MASs. The experimentation of this work uses the INGENIAS Development Kit, because it supports the development of fully functional Multi-agent Systems (MASs) from specification models. As a proof of concept, this work measures two variants of a MAS for managing a crisis situation in a city and shows the relationship between the proposed metrics and the quality of service.

Keywords: Communication load balancing, classification, metrics, multi-agent systems, quality of service, response times.

1 Introduction

During the execution of Multi-agent Systems, several resources (like CPU time and network traffic) are consumed, and consequently they must be properly designed. One of the key aspects of the design of MASs concretely is the manner agents communicate. The goal of this work is to provide a quantitative mechanism to aid designers in designing the communication of their MASs, and the mechanism is based on the load balancing. For instance, a group of agents can always choose the same agent, and the consequences are that an agent is overloaded with the continuous requests while others are not receiving any request. This fact causes the overall time response to be higher, and the overall consequence is a Quality of Service reduction.

For avoiding these kinds of situations, this work presents a corpus of metrics for measuring communication in MASs and detecting the origin of the unbalanced communications. The metrics measures the balance of communication in an active manner (sending messages) and in a passive manner (receiving messages).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 619–626, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The INGENIAS is a well-known established methodology that asserts the importance of communication in MAS. Its support tool, INGENIAS Development Kit (IDK) [1], allows building plug-ins to make the specifications work, and therefore, this fact facilitates the creation of a framework for measuring the proposed metrics. In the application of these metrics to several IDK developments, one can observe that the measurement values of these metrics are strongly related with the Quality of Service (QoS). This paper presents one of these cases, in which the metrics measures the communication of a MAS for crisis-management, and indicates which is the origin of the low QoS of this system.

2 Communication in Multi-Agent Systems

This work measures the amount and type of communication activity in several scopes. According to our study, there can be five main patterns of agents regarding the communication activity: overloaders, overloaded, isolated, overloaded-overloader, and regular. The first four types should not exist in a balanced system and the last one is the ideal agent:

1. *Overloaders*: Agents that overload the communication by sending too many messages.
2. *Overloaded*: Agents that are overloaded by receiving too many messages.
3. *Isolated*: Agents that neither send nor receive any message.
4. *Overloaded-overloader*: Agents that are overloaded but also overload other ones.
5. *Regular*: Agents whose behavior is ideal because they send and receive an adequate amount of messages according to the communication activity of the system.

The existence of the first four types is related with a low quality of QoS of the system. For instance, when an agent is occupied in paying attention to a great amount of requests (an overloaded agent), its response time delays and, consequently, the overall response time of the system also delays. In this example, the response time is used as an indicator of the QoS. The next section presents the necessary metrics for detecting the aforementioned types of agents.

3 Definition of MAS Communication Metrics

In order to determine when an agent is overloading or overloaded, this paper provides the definition of some metrics in two different scopes: the global system and the group of agents playing the same role The classification of the agents will be within each scope, e.g. an agent will be classified into any of the mentioned classes using the value of metrics. . It is important to distinguish both scopes because an agent can have a pattern (e.g. overloader) regarding the rest of the agents of scope (e.g. global scope), and have another pattern (e.g. regular) regarding the agents of the other scope (e.g role scope). For instance, the reason can be that its activity is neutralized by the other agents' activity within a particular scope.

3.1 Scope of the Global System

The scope of the global system is the performance of all agents in the system. Table 1 provides the metric name with each calculation (1), (2), and limits (3), (4). All the metrics presented in this paper use the following notation:

- A1..An are agents.
- S is the set of agents in the system.
- n(X) is the cardinality of X set.
- $r(A_x)$ is the number of the A_x agent's received messages.
- $s(A_x)$ is the number of the A_x agent's sent messages.
- R is the subset of agents playing the same role.

Table 1. Metrics for an agent performance in the system scope

Metric name	Description	Limits
$BS(A_j)$	OverloaderSystem Metric measures the amount of an agent's sent messages comparing it to the amount of sent messages in the system, with the following definition: $BS(A_i)=\frac{s(A_i)}{(\sum_{A_j \epsilon S} s(A_j))/n(S)}$ (1)	$BS(A_j) \epsilon [0, n(S)]$ $Min(BS(A_j))=0$ in case A_j does not send any message. $Max(BS(A_j))=n(S)$ if all messages are sent by the same agent A_j : $BS(A_j)=\frac{X}{\frac{X}{n(S)}}$ (3)
$MS(A_j)$	OverloadedSystem Metric measures the amount of an agent's received messages comparing it to the amount of received messages in the system, with the following definition: $MS(A_i)=\frac{r(A_i)}{(\sum_{A_j \epsilon S} r(A_j))/n(S)}$ (2)	$MS(A_j) \epsilon [0, n(S)]$ $Min(MS(A_j)) = 0$ in case A_j does not send any message. $Max(MS(A_j)) = n(S)$ if all messages are sent by the same agent A_j : $MS(A_j)=\frac{X}{\frac{X}{n(S)}}$ (4)

The following rules interpret the measurement with certain thresholds:

1. Agents who do not receive or send any messages will be classified as isolated.
2. Agents whose BS is much higher than the threshold will be classified as overloaders.
3. Agents whose MS is much higher than the threshold will be classified as overloaded.
4. Agents whose BS and MS are both much higher than the threshold will be classified as overloaded-overloader.
5. Agents whose BS and MS are close to the threshold will be classified as regular, because their communication is balanced.

3.2 Scope of Agents Playing the Same Role

The performance of each agent can also be measured in the scope of the agents playing the same role. Table 2 provides the metric names with each equation (5), (6), and limits (7), (8):

Table 2. Metrics for an agent performance in the role scope

Metric name	Description	Limits
$BR(A_j)$	OverloaderRole Metric measures the amount of an agent's sent messages comparing it to the amount of sent messages by the agents playing the same role, with the following definition: $BR(A_i)=\frac{s(A_i)}{(\sum_{A_j \epsilon R} s(A_j))/n(R)}$ (5)	$BR(A_j) \epsilon [0, n(R)]$ $Min(BR(A_j)) = 0$ in case A_j does not send any message. $Max(BR(A_j)) = n(R)$ if all messages are sent by the same agent A_j : $BR(A_j)=X/(X/n(R))$ (7)
$MR(A_j)$	OverloadedRole Metric measures the amount of an agent's received messages comparing it to the amount of received messages by the agents playing the same role, with the following definition: $MR(A_i)=\frac{r(A_i)}{(\sum_{A_j \epsilon R} r(A_j))/n(R)}$ (6)	$MR(A_j) \epsilon [0,n(R)]$ $Min(MR(A_j)) = 0$ in case A_j does not send any message. $Max(MR(A_j)) = n(R)$ if all messages are sent by the same agent A_j : $MR(A_j)=X/(X/n(R))$ (8)

The agents are classified with the following rules, considering certain thresholds:

1. Agents who do not receive or send any messages will be classified as isolated.
2. Agents whose BR is much higher than the threshold will be classified as overloaders.
3. Agents whose MR is much higher than the threshold will be classified as overloaded.
4. Agents whose BR and MR are both much higher than the threshold will be classified as overloaded-overloader.
5. Agents whose BR and MR are not much higher than the threshold will be classified as regular, because their communication is balanced.

4 Case Study: A Crisis-Management MAS

A MAS with overloaded communication effect has been tested to identify communication patterns. The MAS implements a crisis-management case [2]; in which, a poisonous material has been released into a city. Since the centralized medicals services are not enough, the citizens with medical capabilities can offer help to the affected people in nearby areas; at the same time citizens are warned of the affected locations, in order to avoid them. All the experiments presented in this work can be reproduced

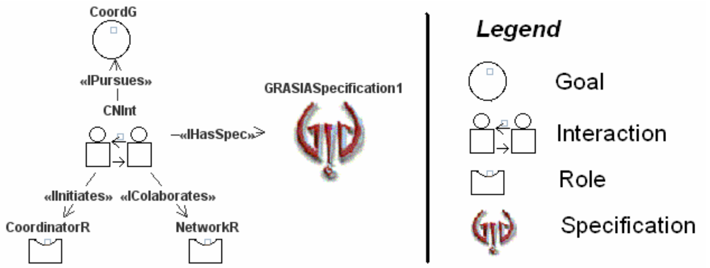


Fig. 1. The Coordinator-Network Interaction, with the INGENIAS notation

with both the *IDK 2.8*, which contains the crisis-management MAS and is available from *Grasia* Web (http://grasia.fdi.ucm.es in "*Software*" section).

There are four coordinator agents (beginning with CoordA prefix) and four network agents (beginning with NetworkA prefix), which exchange messages to coordinate help and notify the risk areas.

The experimentation considers two versions of the system: a crisis simulation with overloaded communication and another with balanced communication. In the first version, the crisis management system may be blocked because the coordinator agents send messages to the same interface agent due to the ingenuous selection in the communication process, which always selects the first agent of the yellow pages of the MAS. In the second version, this problem has been fixed by randomly selecting a network agent. In particular, this work only alters the mechanism of selecting the network agent for triggering the Coordinator-network interaction (CNInt in figure 1).

From the experience of this case study, in both versions, a threshold of unit order (3.00) provides an accurate way to distinguish the regular and irregular patterns, in the global and role scopes, because the values are of unit order.

4.1 Experimental Results in the Crisis System with Ingenuous Communication

Using the metrics from previous sections, concretely BS(A_j), MS(A_j), BR(A_j), and MR(A_j), the results in table 3 have been obtained:

Table 3. Values of the metrics for the agents

Agent ID	BS(A_j)	MS(A_j)	BR(A_j)	MR(A_j)
CoordA_034	0.61	1.85	1.00	1.33
CoordA_134	0.61	1.23	1.00	0.89
CoordA_234	0.61	1.23	1.00	0.89
CoordA_334	0.61	1.23	1.00	0.89
NetworkA_0	0.00	0.00	0.00	0.00
NetworkA_1	5.50	2.46	4.00	4.00
NetworkA_2	0.00	0.00	0.00	0.00
NetworkA_3	0.00	0.00	0.00	0.00

Classification results within the three scopes have been collected from this execution as in table 4:

Table 4. Classification results of the metrics for the agents

Agent ID	Global system	Agents playing the same role
Coord_034	Regular	Regular
Coord_134	Regular	Regular
Coord_234	Regular	Regular
Coord_334	Regular	Regular
Network_0	Isolated	Isolated
Network_1	Overloader	Overloaded-overloader
Network_2	Isolated	Isolated
Network_3	Isolated	Isolated

In the classification, none of the network agents follows a regular pattern; therefore, there is an unbalanced communication in both scopes. Since a Network agent is overloaded while the other Network agents are isolated, the problem is the mechanism-selection of the Network agents in the Coordinator-network interaction. The problem is fixed straightforward by changing this mechanism selection to a more balanced strategy, such as random-selection.

4.2 Experimental Results in the Crisis System with Random-Selection Communication

This architecture has the same number of agents than the previous one. As the origin of the unbalanced communication is the mechanism of selection of the coordinator-agent, this mechanism of selection is changed. Thus, in this version, coordinator-agents randomly interact with any agent of the network-role. In this system, the values of the metrics are presented in Table 5, and the results of applying the classification rules are collected in Table 6. The values of metrics are closer to 1 and all agents of the system are classified as regular; hence, all network agents have taken part in the system in a proper proportion.

Table 5. Values of the metrics for the agents

Agent ID	$BS(A_j)$	$MS(A_j)$	$BR(A_j)$	$MR(A_j)$
CoordA_034	0.55	1.38	1.00	0.95
CoordA_134	0.55	1.65	1.00	1.14
CoordA_234	0.55	1.38	1.00	0.95
CoordA_334	0.55	1.38	1.00	0.95
NetworkA_0	1.65	0.55	1.14	1.00
NetworkA_1	0.83	0.55	0.57	1.00
NetworkA_2	1.65	0.55	1.14	1.00
NetworkA_3	1.65	0.55	1.14	1.00

Table 6. Results of the classification of the agents within the three scopes

Agent ID	Global system	Agents playing the same role
Coord_034	Regular	Regular
…	…	…
Network_0	Regular	Regular
…	…	…

4.3 Response Times in the Crisis System with Ingenuous Communication and with Random-Selection Communication

Table 7 shows the response times for each agent and the system in both variants. The system response time is the average of the agents' response times. CoordA_034 agent has no response time because it is the one who offers the service requested. As suspected, response times are lower in the second variant, because the communication is well balanced and therefore, it improves the response time and quality of service. Therefore, the presented metrics are strongly related with the QoS of the system.

Table 7. Agent and System response times in milliseconds for 4 agents

Agent ID/System	Response time (Ingenuous Communication)	Response time (Random-selection Communication)
CoordA_034	N/A	N/A
CoordA_134	2524	1973
CoordA_234	2987	1287
CoordA_334	698	2526
System	2069	1928

5 Related Work

Load balancing in MASs has been pursued in several ways and applied in multiple scopes. [3] explains a multi-agent reinforcement learning process in MAS load balancing. When choosing a resource, the agent must optimize both the resource usage in the system and fairness by using purely local information. [4] mentions that mobile agents are a paradigm used to balance load in dynamic environments because this type of agent can decide when and where to move, depending on the system conditions. Although load balancing has been studied extensively in MAS context, the design of analysis tools of debugging data of MAS executions is only preliminary. In [5], a task of clustering has been made to detect of agents with similar communication activity. In [6], clustering algorithms and complex data visualization techniques have been applied to the logs of interactions resulting from the execution of a MAS on the JADE platform. [7] has a formalism, called Enhanced Dooley Graphs, to explicate relationships within agent conversations and provide aid to new designs. In contrast, our work has quantitatively evaluated the communication with metrics and classification rules; and these metrics are empirically proven to be strongly related with QoS. Moreover, the advantage of our approach is that assigning these patterns to agents using the IDK methodology, allows the straightforward identification of the interactions that are causing the unbalanced communication. With this information, the designer has just to change the selection mechanism of the affected interaction.

6 Conclusions and Future Work

The presented metrics measure the communication in MASs, and their measurement values are strongly related with the QoS. The metrics and the presented classification rules assist designers in creating balanced communications in MASs and, consequently, efficient MASs. In this case, the overloading effect has been due to a wrong agent selection-mechanism and has been solved by changing it.

In the future, the relationship between the metrics and QoS can be corroborated in other MAS methodologies and tools.

According to the scalability, our preliminary experiments advocate that the unbalanced patterns of communications have worse effects when MASs are larger. Thus, the presented metrics can become more valuable for huge MASs. One of our short-term goals is to corroborate this preliminary theory.

Moreover, the proposed metrics can be applied to open MASs, to facilitate the detection of malicious or problematic agents that considerably decrease the performance of the whole open-system.

Acknowledgments. We acknowledge support from the project Agent-based Modelling and Simulation of Complex Social Systems (SiCoSSys), supported by Spanish Council for Science and Innovation, with grant TIN2008-06464-C03-01.

References

1. Pavón, J., Gómez-Sanz, J.J.: An Agent Oriented Software Engineering with INGENIAS. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) CEEMAS 2003. LNCS, vol. 2691, pp. 394–403. Springer, Heidelberg (2003)
2. Garcia-Magariño, I., Gutierrez, C., Fuentes-Fernandez, R.: Organizing multi-agent systems for crisis Management. In: Proceedings of the 7th Ibero-American Workshop in Multi-Agent Systems, pp. 69–80 (2008)
3. Andrea, S., Shoham, Y., Tennenholtz, M.: Adaptive Load Balancing: A Study in Multi-Agent Learning. Journal of Artificial Intelligence Research 2, 475–500 (1995)
4. Gomoluch, J., Schroeder, M.: Information Agents on the Move: A Survey on Load-balancing with Mobile Agents. Software Focus 2, 31–36 (2001)
5. Botia, J.A., Hernansaez, J.M., Gomez-Skarmeta, A.F.: On the Application of Clustering Techniques to Support Debugging Large-Scale Multi-Agent Systems. In: Bordini, R.H., Dastani, M., Dix, J., El Fallah Seghrouchni, A. (eds.) PROMAS 2006. LNCS (LNAI), vol. 4411, pp. 217–227. Springer, Heidelberg (2007)
6. Botia, J.A., Gomez-Sanz, J.J., Pavon, J.: Intelligent Data Analysis for the Verification of Multi-Agent Systems Interactions. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 1207–1214. Springer, Heidelberg (2006)
7. Parunak, H.V.D.: Visualizing Agent Conversations: Using Enhanced Dooley Graphs for Agent Design and Analysis. In: Proceedings of 2nd International Conference on Multi-Agent Systems, pp. 275–282 (1996)

Agents Jumping in the Air: Dream or Reality?*

Oscar Urra, Sergio Ilarri, and Eduardo Mena

Department of Computer Science and Systems Engineering, University of Zaragoza,
María de Luna 1, 50018, Zaragoza, Spain
`ourra@ita.es`, `{silarri,emena}@unizar.es`

Abstract. Mobile agent technology has traditionally been recognized as a very useful approach to build applications for mobile computing and wireless environments. However, only a few studies report practical experiences with mobile agents in a wireless medium. This leads us to the following question: is mobile agent technology ready to be used in this environment?

In this paper, we study existing mobile agent platforms by analyzing if they could be used effectively in a wireless medium. We identify some key missing features in the platforms and highlight the requirements and challenges that lie ahead. With this work, we expose existing problems and hope to motivate further research in the area.

1 Introduction

Mobile agents [6] are programs that execute in contexts called *places*, hosted on computers, and can autonomously travel from *place* to *place* resuming their execution there. Thus, they are not bound to the computer where they were created; instead, they can move freely between computers.

Mobile agents provide interesting features, thanks to their autonomy, adaptability, and capability to move to remote computers. They can carry the computation wherever it is necessary, without the need of installing specialized servers there (only a generic *mobile agent platform* [12] is needed). In particular, the interest of mobile agent technology for wireless environments has been emphasized in the literature (e.g., see [10]). Thus, for example, instead of communicating a large amount of data from a computer to a mobile device, a mobile agent can move to that computer to process the data locally, filtering the non-relevant data that should not be communicated through the network. As another example, a mobile device could use a mobile agent to perform a processing-intensive task on a fixed computer with the required resources, relieving the overload of the mobile device. This will increase, in turn, its battery life, which is an important limitation on these devices.

However, and despite mobile agent technology has been evaluated in many research works in the context of distributed systems, there are only a few reported experiences on the use of mobile agents in real wireless environments with

---

* This work was supported by the CICYT project TIN2007-68091-C02-02.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 627–634, 2009.
© Springer-Verlag Berlin Heidelberg 2009

mobile devices. In most cases, a simple and static wireless environment is considered (e.g., we performed an experimental evaluation in this context in [13]). Moreover, we also believe that it is important to study the challenges that mobile environments imply for a mobile agent platform. For example, in ad hoc networks multi-hop protocols may be required to reach a certain node (as each node can only communicate with other nodes within its communication range) As another example, it becomes apparent that mobile agent platforms should provide services for agents to discover other nodes.

As far as we know, no other work has studied in depth the requirements and current limitations of mobile agent platforms to be usable in a mobile environment[1], which is the goal of this paper. We analyze the challenges that need to be solved and highlight the advantages and disadvantages of existing platforms in a mobile environment. The structure of the rest of this paper is as follows. In Section 2, we introduce the concept of mobile agent platform and motivate the development of this work. In Section 3, we study security issues. In Section 4, we consider the challenges of using wireless communications and mobile ad hoc networks. In Section 5, we focus on architectural elements that must be considered. Finally, in Section 6, we summarize our conclusions.

2 Mobile Agent Platforms in Wireless Environments

There exist several *mobile agent platforms*, which allow the execution of mobile agents and provide them different services (e.g., communication services, transportation services, security, etc.). These platforms differ in different aspects (such as their general architecture, communication style, etc.) and compare differently in terms of performance, reliability or scalability [12].

Mobile agent technology has been proposed as a key element in the development of many applications, for a variety of reasons (e.g., their capability to exploit the locality of data by moving to the data source instead of interacting remotely using a network). With the increasing popularity of mobile devices, it was a natural step forward to try to apply mobile agents to the mobile environment. Given its portable nature, mobile devices use wireless communications, creating a scenario completely different from a traditional distributed environment with fixed networks. Such an environment has a number of advantages (e.g., the processing is not tied to a fixed location) but also some drawbacks, such as the limited computational power of mobile devices and a short communication range based on wireless technologies –that usually offer a low bandwidth, a high latency, and intermittent/unreliable connectivity–. Thanks to their features, mobile agents can be very useful in wireless applications (e.g., see [10]), as they could help to reduce the negative effects of such limitations.

However, despite the increasing popularity of wireless services and the advantages of mobile agents in these environments, there are not many practical

---

[1] We are only aware of the work presented in [1], which focuses only on security issues over a P2P communication model for mobile agents in ubiquitous computing and does not analyze the existing platforms.

applications of this technology, except for some proofs of concept. A possible explanation could be that it is very difficult to develop and maintain such mobile agent based applications because existing platforms lack a number of features that should be present for their use in a wireless and mobile environment. Among them, we could emphasize:

- Features related to security, since the use of wireless communications broadcast data that could be intercepted or altered without having any notice.
- Features related to special network topologies, since there are multiple mobile nodes with short range and unstable communications, which can make the process of transfer data between two nodes challenging.
- Features related to the way the platform itself works, since the mobile agents need some services to perform their tasks efficiently, such as the transportation to other places or the communication with other agents in the network.

Not providing the features listed above is an important difficulty to develop applications based on mobile agents for mobile environments. Thus, for example, an agent should be able to detect the availability of new nearby devices (e.g., to travel to them) and to communicate with other agents easily and efficiently. Mobile agent platforms have usually been developed with a static context in mind and now they must be adapted to a more open and dynamic environment.

Besides, scalability and reliability are two key features that should also be provided. They are important even when the mobile agents rely on fixed networks, and therefore even more critical in a challenging environment with wireless networks. Some tests indicate that the scalability/reliability of some platforms should be improved [3]. However, as this need is not tied to the wireless case –although heightened by it– we will focus on the other features in the rest of this paper. We will consider the following platforms: SPRINGS (`http://osiris.cps.unizar.es/SPRINGS/`), Voyager Community Edition (`http://www.recursionsw.com/`), and JADE with LEAP (`http://jade.tilab.com/`). SPRINGS provides a high scalability in distributed environments [3]. Along with SPRINGS, Voyager is one of the most reliable platforms according to experiments performed in [12]. Finally, JADE is currently the most popular agent platform and its extension LEAP supports agents on mobile devices. Therefore, we believe that these platforms will be representative of the state of the art.

3 Security Issues

Security is always a major concern. Besides security problems with mobile agents in fixed networks [2,14], other problems particular to wireless environments arise.

3.1 Communication Encryption

Wireless communications usually broadcast the transmitted data in an omnidirectional way. The disadvantage is that these data will be received not only by the intended destination but also by anyone within the range of the originating communication device. Although almost every wireless communication protocol (such as Wi-Fi or Bluetooth) can encrypt the data, it is not mandatory and in some circumstances an unencrypted communication can be the only form available (e.g., with public access points or *hotspots*). If a mobile agent is transferred using an insecure channel, its code and data will be exposed to any nearby device. To avoid this problem, the mobile agent platform should be able to encrypt all its communications when connecting to others.

Regarding the considered existing mobile agent platforms, both Voyager and JADE-LEAP can natively use SSL connections –which assure data flow encryption–, whereas SPRINGS lacks this feature.

3.2 Code Integrity

The code of a mobile agent can be altered, either intentionally or accidentally, in many ways. For example, in the first case, a malicious user could modify it while it is running on its device. In the second case, a failure such as a memory loss or a problem with the wireless transmission of the agent could lead to a corruption of its code. The execution of an agent whose code has been altered can lead to unpredictable consequences and should be avoided. Thus, the mobile agent platform should verify the agent's code integrity before starting its execution, for example signing the code with a X.509 digital certificate.

Thanks to the use of SSL by Voyager and JADE-LEAP, any attempt to tamper with an agent while it is being transmitted will be detected. Moreover, Voyager provides a signing mechanism to assure that the agent is not modified while it is stored in the device's memory. SPRINGS lacks both features.

3.3 Authentication and Trust

A mobile device has a number of resources (such as the CPU, the memory, the file system, the user interface, etc.) susceptible to be used by an agent to accomplish its task. To avoid abuses on the use of these resources, the platform should state the extent to which it trusts an incoming agent –this is critical in an open environment–. Depending on it, the platform should allow, limit or even deny the mobile agent the access to the resources. There already exist mechanisms of authentication for distributed environments (e.g., Kerberos) that could be used to determine the owner of an agent. Once it is authenticated, different access control policies can determine the resources that the agent can use.

Voyager and JADE-LEAP have mechanisms to verify the identity of agents and other platform components, and grant different privilege levels. However, SPRINGS has no authentication mechanisms and just a basic access control.

4 Network Issues

Existing mobile agent platforms assume the existence of a TCP/IP network where every node can potentially connect to any other. This approach has the advantage of its simplicity, but prevents considering explicitly other forms of communication that could be beneficial in a context with wireless mobile agents.

One limitation of mobile devices is that their communication interfaces (usually Wi-Fi or Bluetooth) have a relatively short range (a few hundred meters). Therefore, in a mobile ad hoc network a multi-hop routing protocol may be needed for two nodes to communicate. For a mobile agent platform, it would be interesting to access information about: whether the connection is fixed or wireless, the bandwidth available, if an established link (e.g., a Wi-Fi connection) is encrypted or not, the coverage level or the strength of the received signal in the case of a wireless communication, the identifiers of nearby wireless networks or nodes that could be contacted, etc.

With all this information, the platform would be able to take different useful decisions. For example, it could select the most appropriate communication link if there are several options available, in the case of a secure channel it would avoid encrypting the communicated data –reducing the CPU utilization and therefore saving battery power in the mobile device–, it could decide not to retry a failed communication if the coverage is poor, etc. Making this information accessible to the agents would also be interesting. For example, an agent could decide to jump to another device/node if its current device is getting out of coverage.

As far as we know, current mobile agent platforms do not have the ability to obtain this kind of information, which would be important in order to make the platform truly adaptive to different network environments. The closest related features that the considered platforms have are the following: Voyager can measure the network latency and detect that a communication is broken when an abnormally high value is obtained; JADE-LEAP can get basic information about the link status (connected or disconnected); finally, SPRINGS retries failed communications automatically according to some predefined policy.

5 Architecture Issues

In this section, we provide an overview of some architectural issues that should be considered in a mobile agent platform suitable for a mobile environment.

5.1 Automatic Discovery Service

As opposed to a fixed distributed infrastructure for mobile agents, a mobile context usually presents an open environment where many different computers/devices may appear and disappear at any time. Therefore, it is of paramount importance to provide agents with appropriate mechanisms to locate nodes to where they can travel. Moreover, the capabilities/services offered by these nodes should be advertised to allow the agents to decide a convenient target node.

Discovery of Services. There exist several service discovery protocols, such as the *Service Location Protocol* (*SLP*), the *Universal Plug and Play* (*UPnP*), or the use of the *Jini* technology. The suitability of these protocols for ambient intelligence is analyzed in [9].

Mobile agents should be provided an automatic discovery mechanism to allow them the detection of services of interest[2]. However, as far as we know, these protocols have not been integrated in any existing mobile agent platform. Moreover, it is not clear if they are the best choice in a mobile environment. For example, JADE-LEAP has a federated Yellow Pages (YP) service, but allocating it to a node is not transparent to the programmer and movements of the mobile device may invalidate the convenience of using that allocated YP service.

Finally, we would like to highlight the importance of providing a *semantic matching* of services [11], not only *syntactic matching*, if we want to enable dynamic and flexible interactions among the mobile agents. Thus, *Agent Communication Languages* (*ACLs*) [4] could play an important role. The language proposed by the *Foundation for Intelligent Physical Agents* (the *FIPA* ACL –see `http://www.fipa.org/`–) is the most popular proposal and it is supported by several mobile agent platforms (e.g., JADE). If services for mobile agents are implemented as web services, as suggested in [15], then existing techniques for web services (UDDI, WSDL, SOAP, OWL-S, etc.) could also be adopted.

Discovery of Nodes. It is necessary for an agent to be able to detect the nodes that are reachable and which services/features provide them. Thus, nodes must be auto-descriptive. This is particularly important in a heterogeneous environment where there will be computers/devices with different processing and communication capabilities.

However, no platform provides mechanisms to allow an agent to detect other potential target computers/devices. Although some platforms provide name services to query the computers that can host an agent (e.g., the Region Name Server in SPRINGS [3]), they do not consider that some computers/devices may not be accessible from a given location (e.g., if the mobile agent is executing on a device that can only communicate with other devices within communication range); indeed, the name service itself may be unreachable.

5.2 Tracking of Mobile Agents and Name Services

As mobile agents move from one computer/device to another, tracking their locations efficiently (which is required if we want to be able to allow communications with those agents from any computer/device) is challenging. To solve this problem, different approaches have been considered. Some mobile agent platforms provide a naming service that can be used to locate an agent and then send a message to its address. An alternative approach supports communications by using *proxies* (similar to the *stubs* in *RMI*) as a convenient abstraction to refer to remote agents (e.g., to send them messages or call methods remotely). Several platforms, such as SPRINGS and Voyager, support proxies. Related to the

---

[2] Some works propose using mobile agents to implement service discovery [5,8].

Table 1. Comparison of mobile agent platforms: features for mobile environments

Feature		**Voyager**	**JADE/LEAP**	**SPRINGS**
Security Issues	Encryption	Yes (SSL)	Yes (SSL)	No
	Code integrity for transmissions	Yes (SSL)	Yes (SSL)	No
	Code integrity for execution	Some (signing)	No	No
	Authentication and trust	Yes (very rich)	Yes (rich)	Very basic
Network Issues	Access to link layer	No	No	No
	Adaptability	Latency measure	Conn. status	Conn. retrying
Architectural Issues	Discovery of nodes	No	No	No
	Transparent service discovery	No (YP)	No (DF)	No
	Adapted tracking approach	No (AgentSpaces)	No (AMS)	No (RNS)
Strong points		Messaging, devices	Messaging, ontologies	Scalable, reliable

concept of proxies, *dynamic proxies* remain valid independently of the agents' migrations (i.e., the reference is updated automatically) [3].

However, existing strategies have been developed with a fixed network in mind and are not appropriate in a mobile environment. For example, SPRINGS assumes the existence of stable *Region Name Servers* (*RNSs*) and *location servers* with tracking responsibilities. This could be unsuitable in a dynamic context because nodes can leave/enter the network at any time and some nodes could be temporarily unreachable. Similarly, according to [7], Voyager uses forwarding chains of proxies, which may be inconvenient because any link (pointer) in the chain could disappear at any time. Thus, we believe that new tracking techniques are needed in this context, avoiding mechanisms that rely on the availability of certain nodes or centralized approaches, in favor of adaptive tracking approaches.

Finally, we should mention that ensuring the uniqueness of agent names and at the same time providing user-friendly mechanisms to address the agents is a challenge in an open and dynamic environment. A potential solution would imply a shift in the way agents communicate. At present, it is usually assumed that agents identify their partners by name. An alternative would be to identify partners by service, which would make user-friendly agent names unnecessary.

6 Conclusions

Mobile agent technology has been highlighted as a very interesting approach to build applications for mobile environments. However, despite many theoretical studies proving the usefulness of this alternative, it is hard to find practical applications with real prototypes and using the available mobile agent platforms. One reason is probably that such platforms have been developed with a fixed distributed environment in mind, but not considering the features that may be of special interest in a mobile environment (e.g., reliance against security threats, adaptation to the network technology, and service/node discovery). Considering these features would make the adoption of the technology much easier.

In this paper, we have identified the requirements and desired features of mobile agent platforms to be used in a mobile environment. In light of these requirements, we have analyzed the missing features in some popular mobile agent platforms (see Table 1 for a summary). Thus, we cover a gap in the literature, where many works propose the use of mobile agents in wireless environments but without analyzing the applicability of current mobile agent platforms.

As future work, we plan to study these issues in detail and propose solutions for SPRINGS [3]. We hope that this work will encourage research and development efforts that will eventually lead to a big leap forward into the wireless.

References

1. Bagci, F., Schick, H., Petzold, J., Trumler, W., Ungerer, T.: Communication and security extensions for a ubiquitous mobile agent system (UbiMAS). In: 2nd Conf. on Computing Frontiers (CF 2005), pp. 246–251. ACM Press, New York (2005)
2. Greenberg, M., Byington, J., Harper, D.: Mobile agents and security. IEEE Communications Magazine 36(7), 76–85 (1998)
3. Ilarri, S., Trillo, R., Mena, E.: SPRINGS: A scalable platform for highly mobile agents in distributed computing environments. In: 4th Intl. WoWMoM 2006 Workshop on Mobile Distributed Computing (MDC 2006), pp. 633–637. IEEE, Los Alamitos (2006)
4. Kone, M.T., Shimazu, A., Nakajima, T.: The state of the art in agent communication languages. Knowledge and Information Systems 2(3), 259–284 (2000)
5. Meier, R.T., Dunkel, J., Kakuda, Y., Ohta, T.: Mobile agents for service discovery in ad hoc networks. In: 22nd Intl. Conf. on Advanced Information Networking and Applications (AINA 2008), pp. 114–121. IEEE, Los Alamitos (2008)
6. Milojicic, D., Douglis, F., Wheeler, R.: Mobility: processes, computers, and agents. ACM Press, New York (1999)
7. Moreau, L.: A fault-tolerant directory service for mobile agents based on forwarding pointers. In: Symp. on Applied Computing (SAC 2002), pp. 93–100. ACM, New York (2002)
8. Nehra, N., Patel, R.B., Bhat, V.K.: MASD: Mobile agent based service discovery in ad hoc networks. In: Aluru, S., Parashar, M., Badrinath, R., Prasanna, V.K. (eds.) HiPC 2007. LNCS, vol. 4873, pp. 612–624. Springer, Heidelberg (2007)
9. Preuveneers, D., Berbers, Y.: Suitability of existing service discovery protocols for mobile users in an ambient intelligence environent. In: Intl. Conf. on Pervasive Computing and Communications, pp. 760–764. CSREA Press (2004)
10. Spyrou, C., Samaras, G., Pitoura, E., Evripidou, P.: Mobile agents for wireless computing: the convergence of wireless computational models with mobile-agent technologies. Mobile Networks and Applications 9(5), 517–528 (2004)
11. Sycara, K.P., Widoff, S., Klusch, M., Lu, J.: Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. Autonomous Agents and Multi-Agent Systems 5(2), 173–203 (2002)
12. Trillo, R., Ilarri, S., Mena, E.: Comparison and performance evaluation of mobile agent platforms. In: 3rd Intl. Conf. on Autonomic and Autonomous Systems (ICAS 2007), p. 41. IEEE, Los Alamitos (2007)
13. Urra, O., Ilarri, S., Mena, E.: Testing mobile agent platforms over the air. In: 1st ICDE Workshop on Data and Services Management in Mobile Environments (DS2ME 2008), pp. 152–159. IEEE, Los Alamitos (2008)
14. Vigna, G. (ed.): Mobile Agents and Security. LNCS, vol. 1419. Springer, Heidelberg (1998)
15. Zhang, J., Wang, Y., Varadharajan, V.: A new security scheme for integration of mobile agents and web services. In: 2nd Intl. Conf. on Internet and Web Applications and Services (ICIW 2007), p. 43. IEEE, Los Alamitos (2007)

Using Scenarios to Draft the Support of Intelligent Tools for Frail Elders in the *SHARE-it* Approach

R. Annicchiarico[1], F. Campana[2], A. Federici[1], C. Barrué[3], U. Cortés[3],
A. Villar[3], and C. Caltagirone[1]

[1] Fondazione Santa Lucia, Rome, Italy
[2] CAD, Rome, Italy
[3] Technical University of Catalonia, Barcelona 08034, Spain

Abstract. A scenario is an example narrative description of typical interactions of users with the system. Scenarios are concrete, informal descriptions, and are not intended to describe all possible interactions. This document describes the scenarios used in the *SHARE-it* project. These scenarios have several purposes. The main purpose of scenarios is to provide a communication tool between the user experts and the technology experts. As scenarios describe the actual deployment of the system, it encapsulates both knowledge about how the system can be useful, and also what is technically feasible. Each scenario in this document has the following items: Purpose of the scenario, User Description, Narrative Scenario, Structure of the scenario, Roles of the Agents, Role of Communication.

1 Introduction

We do believe that in order to solve problems, we, humans are able to synthesize apparently unrelated concepts, take advantage of serendipitous opportunities and previous experiences, hypothesize, invent solutions to everyday problems in *well-know* environments.

SHARE-it is exploring *how* robotic and agent-based technologies, designed in forms as familiar as mobility platforms and other interface appliances, might be used to assist elders to stay longer at their preferred environment and to give support those who provide care. Mobility is a key aspect for elderly and impaired people, which the *SHARE-it* project addresses with the development of four different mobility platforms, namely Rolland, Spherik, *ı-Walker* and CARMEN. Interdisciplinary research requires the collaboration of several groups of participants with different backgrounds. In the *SHARE-it* project for instance, the medical doctors are experts in their domain, and have knowledge about how the system might be useful to support certain types of disability. On the other hand, the developers of the technology have more experience with building systems, but have little knowledge of the everyday activities and needs of the users. The former are called the application domain experts, and the latter

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 635–641, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the solution domain experts [4]. The main purpose of scenarios is to provide a communication tool between these expert groups. As scenarios describe the actual deployment of the system, they encapsulate both knowledge about how the system can be useful, and also what is technically feasible. A scenario is a description of a user's interaction with a system. Scenarios help focus design efforts on the user's requirements, which are distinct from technical or business requirements. Inhere scenarios are related to use cases, which describe interactions at a technical level. Unlike use cases, however, people who do not have any technical background can also understand scenarios. Natural language is used for scenarios, so that all participants could understand them easily. Scenarios are also used to determine what exactly the *SHARE-it* system should do. By designing and reviewing the scenarios, application domain experts know what to expect from the system, and solution domain experts know what must be implemented. Scenarios can be updated to reflect novel implementation decisions. The usability of a system can be characterized by *how well* it fits the users intentions. Therefore, a goal-directed design process, which explicitly takes these intentions into account, is a valuable method to develop useful systems [5].

The Assistive Technology (AT) developed during the life of this project is aimed to use users capabilities to enable them to do things that they previously could not do due to their needs not being addressed by the existing standard technology. In this document we assume that the mobility platforms mentioned provide equivalent services as the ones developed during the life of the *SHARE-it* project: CARMEN, *ı-Walker*, Rolland and Spherik. The technical characteristics of those mobility platforms are described in [1] [8] [9]. Although not typically part of a scenario description, this document also describes the high-level interaction of different system modules. The level of abstraction at which this is done is that of the agents, which are described more elaborately in [7] and [3]. In this paper, the goal is not to completely describe all possible interactions between the agents, but rather describe interactions that arise during a typical example of use. The specification of the communication level between the elements of the architecture is described in [2]. *SHARE-it* relies on Agent Based Control for the implementation of the middleware, which will be used for *interpretation and intelligent distribution of available information.* This agent system, composed of different Agents that provide a set of high level services, adapted to each user needs, as result of the interaction with other modules that require cognitive processes such as: a) Route Planning, b) Proactive monitoring of patient actions, c) Proactive monitoring of local environment conditions, d) Scheduling daily activities, e) Cognitive assistance and, f) Proactive emergency detection and planning in the system. Finally, the narratives of the scenario texts are also used to describe the *SHARE-it* project to the general public, and will be included on the *SHARE-it* website (http://www.ist-shareit.eu).

1.1 Plan of the Paper

In §2 we give a concise description of the *SHARE-it* Target Population. We give just the basic elements to allow the reader to put in place the complexity of the support that a user may need and the difficulties in to address the issue of personalization of the services. In §3 describes each one of the elements of the structure of an Scenario and §3.1 is *real* example of one of those scenarios. That is one that is implemented to show the capabilities of *SHARE-it*. In §4 we give some conclusions of our work and mark the future lines of work.

2 Target Population

A pivotal moment of the *SHARE-it* project is the modelling of the users and of the actions to be supported. The main aim of this section is to define the Target Population and the functions to be supported to help the user in recovering self-dependency through cognitive/physical support. Since the major purpose is represented by the possibility of assisting people older or disabled to stay or to get back to their homes, it has to be individuated a target population in terms of functional, social, and clinical features so that a number of individuals as large as possible could achieve benefits as effective as possible. According to these premises, our Target Population will operatively be individuated through the assessment of the presence of:

Mild functional impairment older and/or disabled populations are made up by individuals who present widely different and heterogeneous functional profiles. Considering the more prevalent diagnosis in this group of persons (hip fracture, stroke, Alzheimers disease), impairments range from extremely mild (people able to walk with a cane or affected by such a mild memory loss that allows them to live on their own) to extremely severe (persons bedridden or completely unable to understand a simple order). In the first case we are in the realm of prevention, in the second in that of palliative care or institutionalization. People with mild functional impairment those in the middle area of impairment are expected to have the best results through the use of proper assistive technology and are the target of the assistive intervention.

Possibility of changing functional profile mild disability is characterized by the possibility of increasing the functional level according to proper assistive or rehabilitative interventions. Users of a flexible assistive device can benefit from its adaptation so that it could help to improve their condition as a result of rehabilitation treatment, or can prevent the decrease of their autonomy due to intervening conditions or diseases.

It is difficult to define disability because it is a multi-dimensional concept with both objective and subjective characteristics. When interpreted as an illness or impairment, disability is seen as fixed in an individuals body or mind. When interpreted as a social construct, disability is seen in terms of the socio-economic, cultural and political disadvantages resulting from an individuals exclusion. Persons with disabilities, advocacy groups, medical practitioners and the general public all have a different view of disability. The meaning of disability has evolved over the years through various perspectives such as a moral perspective, a medical model as well as social and human rights perspectives

3 Scenarios in *SHARE-it*

A scenario is an example narrative description of typical interactions of the users with the system. There are several types of scenarios [6]:

- *As-is scenarios*, which describe the current situation.
- *Visionary scenarios*, which describe the future system. This is the main communication tool between the application and solution domain experts. In a sense, these scenarios are inexpensive prototypes.
- *Evaluation scenarios*, which are used to test if the system provides the desired functionality.

Scenarios are concrete, informal descriptions, and are not intended to describe all possible interactions [4]. In *SHARE-it* we defined 9 scenarios in here we will describe only one of them. For each scenario, we describe the following aspects:

Purpose of the scenario describes which aspects of the system and its interactions with the user are focused by the scenario.

User Description a description of the typical user for the scenario. The actor represents the typical individual of a defined target group. Target groups are described in §2.

Narrative Scenario the narrative that describes an example of the users interaction with the system. Born as visionary scenarios, which described in the first draft of this document the intended functionality of the systems architecture, at later stages, these descriptions can be used as evaluation scenarios to determine whether the intended functionality has been achieved.

Structure of the scenario this is a summary of the concepts present inside the scenario and is provided by the application domain experts. It describes the activities of the user, without the use of *SHARE-it* technology. Therefore, this part is the *as-is scenario*. At the same time, services provided by the *SHARE-it* system are anticipated (see table 1 as an example).

Roles of the Agents Describes the interactions between the different agents and the technology they represent. The main agent types interacting in these scenarios are:

- Patient Agents (*pa*), which will run in PDAs or Ultra-Mobile PCs. An instantiation of this agent should provide all the available and permitted services to each user for instance security, mobility, monitoring and help services.
- The Vehicle Agents (*va*), allocated in the assistive hardware devices related to the project, be it the different wheelchair (CARMEN, Rolland, Spherik) or in the *ı-Walker*. Most services will be related to mobility, monitoring and resource management.
- The Caregiver Agents (*cga*) will be situated in the computers belonging to the caregivers of *SHARE-it* target population as well as in their individual PDA. These agents will be in charge of managing all the user's help request messages, so they can be attended properly. Also, it will be notified if any anomalies are detected in the user's biometric signals.

Table 1. Structure of the Scenario

Disability	Degree	Disease	Device	Environment	Domain	Functional ADL	Services
Physical	Mild	Hip fracture (post surgery)	i-Walker	Bedroom	Functional Activities	Transfer	Transfer support
				Bedroom Corridor Kitchen	Functional Activities	Mobility	Navigation
				Kitchen	Functional Activities	Transfer	Transfer support

- The Environment Agent (*va*) will run in a network computer connected to all the environment network of sensors (*i.e.* AmI sensors). Its basic target is to distribute the information from all available sensors to all the agents interested.

Role of Communications describes the communication infrastructure used in the scenario.

Involved *SHARE-it* technologies describes which elements of the *SHARE-it* modular technology need to be available and active in the scenario.

3.1 Martha Goes to the Kitchen with the *ı-Walker*

Purpose Scenario 1 shows a typical interaction of the user with his/her *ı-Walker* [8]. The first scenario features two services: *assisted navigation* and *transfer support*.

User Description Martha is a 75-year old woman who lives alone since her husband died 5 years ago. She suffers from a moderate physical disability after hip fracture surgery and therefore uses as a mobility tool her *ı-Walker*.

Narrative Scenario as every morning, Martha wants to go to the kitchen and have a breakfast. She is in bed and needs the *ı-Walker* to help her stand-up and go to the kitchen. The *ı-Walker* stands still, with braked wheels, obtaining the necessary information from the environment sensors. Since Martha suffers from a decreased range of motion, she needs to lean on the *ı-Walker* especially when changing position (*i.e.* from sitting to standing). The *ı-Walker* detects an exceeded force on the right side and consequently breaks the ipsilateral wheels. Martha successfully gets up and starts walking towards the kitchen. On her way, Martha moves across a corridor where static obstacles mainly furniture exist. The interaction between the *ı-Walker* and the environment allows a gentle navigation after the necessary corrections (correct trajectory), provided by the*ı-Walker*. Martha gets to the kitchen. She leans on the *ı-Walker* once again when trying to sit down. Once again the *ı-Walker* detects an exceeded force on the right side and consequently breaks the ipsilateral wheels. Finally, Martha sits at the kitchen table and has her breakfast.

Roles of the Agents

1. Users are interacting with their *pa* placed in a UMPC or other computational device.
2. The *pa*s Agenda and Activity Monitor service has stored knowledge of the users daily routine, and the average timings spent with each activity. Following the daily schedule, the patient-agent may engage a Reminder acknowledging the user that is time for that activity.
3. Using the information of its Localizator service the *pa* will request the navigation module in *va* a plan to reach the kitchen (connected via Wi-Fi) and will be giving support only in the case of users disorientation.
4. Once the user reaches the kitchen, the *pa*'s Agenda and Activity Monitor service checks the snack activity as started and keeps monitoring. The activity will be summarized in the daily agenda and stored in the home-agent, so the user or the caregiver can check what has been done through the day.

Role of Communication in the most basic case, the *ı-Walker* could operate on its own and leave all considerations to Martha. Still, it would be interesting to at least have the localization network active so that the Agenda and Activity Monitor service can be up to date, so Wi-Fi should be available at Marthas home. If reminders are required and the home agent needs to communicate, a local network needs to be active in the environment. Basically, the localizator service outside the *ı-Walker* will communicate with the vehicle agent to correct trajectories if necessary by updating the target position. The Agenda and Activity Monitor may communicate with Martha via her handheld preferred device: a mobile phone or a PDA supporting Wi-Fi. In brief, it is required:

- Odometric localization and time stamp in *ı-Walker* connected to a server via Wi-Fi.
- Wi-Fi connection between localization network and *ı-Walker* vehicle agent
- If *pa* is attached to *ı-Walker*, it is connected to the server via *ı-Walker* Wi-Fi. Otherwise, a Wi-Fi enabled PDA is required.
- DLA software protocol used for internal communications.

Involved SHARE-it technologies this scenario relies on the most basic *ı-Walker* services for assisted navigation, meaning that the *ı-Walker* requires its range sensor to detect obstacles in the way and also the pressure sensors in the handles to check how weight is distributed. The *ı-Walker* CPU should be active to perform assisted navigation. If the person does not have cognitive disabilities, it would be interesting to have localization sensors related to the *ı-Walker* (ultrasound or RFID, KNX, Zb) so that the home agent can update activity record. Yet, in this case, no reminders need to be sent to the user and, hence, localization is not compulsory. If reminders need to be sent to the user, a terminal (*e.g.* PDA, mobile phone) is required and localization via the aforementioned technologies is compulsory and fed to the *ı-Walker* CPU to update targets in case Martha steers from the kitchen path. In this last case, the user needs a Wi-Fi network at home.

4 Conclusions

Trying to be patient-centered enables to delve into the minute details of our domain. Domain knowledge takes on particular importance for success when the user's characteristics are complex. Scenarios are a good vehicle for representing domain knowledge and to explain the real complexities of the kind of problems that one can expect when dealing with providing continuos personalized support to elders. Domain knowledge, in this case, includes the user's profile characteristics, the existing abilities and deficits, the environment, the caregiver environment, the supporting technology available, etc.

References

1. Annicchiarico, R., Cortés, U., Federici, A., Campana, F., Barrué, C., Martínez, A.B., Caltagirone, C.: The Impact of Cognitive Navigation Assistance on people with special needs. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 1060–1066. Springer, Heidelberg (2007)
2. Barrué, C., Céspedes, L., Cortés, U., Annicchiarico, R., Caltagirone, C.: e- toolsv1: The agent level. In: Proceedings of 2nd ECAI Workshop on Agents Applied in Health Care, pp. 55–61 (2004)
3. Barrué, C., Cortés, U., Martínez, A.B., Vázquez-Salceda, J., Annicchiarico, R., Caltagirone, C.: An *e*-institution framework for the deployment of e-tools to support persons with disabilities. In: 3^{rd} Workshop on Agents Applied in Health Care. 19^{th} International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 6–15 (2005)
4. Bruegge, B., Dutoit, A.H.: Object-Oriented Software Engineering: Using UML, Patterns and Java, 2nd edn. Prentice Hall, Upper Saddle River (2003)
5. Budde, S.H., Stulp, F., Sancho-Prade, D.L.: Using persona descriptions as a communication tool in interdisciplinary system design. Gerontechnology 2008 72(2), 82 (2008)
6. Carroll, J.M. (ed.): Scenario-Based Design: Envisioning Work and Technology in System Development, 1st edn. Wiley, New York (1995)
7. Cortés, U., Martínez, A.B., Barrué, C., Benedico, T., Campana, F., Annicchiarico, R.: An agent-based service to elders mobility using the i-walker. In: 7^{th} International Conference on autonomous agents and Multiagent systems (AAMAS 2008) (2008)
8. Martínez, A.B., Escoda, J., Benedico, T., Cortés, U., Annicchiarico, R., Barrué, C., Caltagirone, C.: Patient driven mobile platform to enhance conventional wheelchair, with mul-tiagent system supervisory control. In: Pěchouček, M., Petta, P., Varga, L.Z. (eds.) CEEMAS 2005. LNCS, vol. 3690, pp. 92–101. Springer, Heidelberg (2005)
9. Sánchez-Tato, I., Toro, J.C., Pérez, E.J., Urdiales, C., Cortés, U., Annicchiaricco, R., Sandoval, F., Caltagirone, C.: Collaborative control for personalized mobility assistance to persons with physical/cognitive disabilities. In: Proc. of IROS-2007-Workshop on Assistive Technologies (IROS 2007) (2007)

On the Road to an Abstract Architecture for Open Virtual Organizations*

M. Rebollo[1], A. Giret[1], E. Argente[1], C. Carrascosa[1], J.M. Corchado[2], A. Fernandez[3], and V. Julian[1]

[1] Universidad Politecnica de Valencia
[2] Universidad de Salamanca
[3] Universidad Rey Juan Carlos de Madrid (Spain)

Abstract. The paper presents an abstract architecture specifically addressed for the design of open multi-agent systems (MAS) and virtual organizations. The proposal takes into account all the components to cover the characteristics and needs for systems of this kind. This architecture has been called THOMAS (Me*TH*ods, Techniques and Tools for *O*pen *M*ulti-*A*gent *S*ystems) and consists of a set of modular services.

1 Introduction

Research into MAS organizations has ranged from basic organizational concepts, such as groups, communities, roles, functions [10,13,6,12]; organizational modeling [9,6,4]; Human Organization Theory [1]; structural topologies [8,2]; to normative research, including internal representation of norms [11], deontic logics [3] and institutional approaches [5]. But there is not an approach which covers all the needed lifecycle management of an agent organization. Another key problem for open MAS organizations development is the vision of agents and organizations as service-provider entities. This vision needs a main effort in the integration of agents and web services directed at masking services for redirection, aggregation, integration or administration purposes [7].

This work presents a new open multi-agent system architecture consisting of a related set of flexible modules that are suitable for the development of open MAS systems and virtual organizations where agents can offer and invoke services in a transparent way from other agents, virtual organizations or entities. This requires, as a first step, the high-level design of an abstract architecture. This new architecture has been called THOMAS (Me*TH*ods, Techniques and Tools for *O*pen *M*ulti-*A*gent *S*ystems).

This paper is structured as follows: section 2 briefly presents the proposed architecture model; section 3, 4 and 5 presents the three main modules of the proposal with a description of the services offered by each one of the modules; finally some conclusions and future lines of work are shown in section 6.

---

* This work has been partially funded by TIN2006-14630-C03 project of the Spanish goverment, FEDER funds and CONSOLIDER-INGENIO 2010 under grant CSD2007-00022.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 642–650, 2009.
© Springer-Verlag Berlin Heidelberg 2009

2 Architecture Model

THOMAS architecture consists basically of a set of modular services. THOMAS feeds initially of the FIPA architecture expanding its capabilities. The agents have access to the infrastructure offered by THOMAS through a range of services including on different modules or components. The main components of THOMAS are the following: (i) *Service Facilitator* (SF), this component offers simple and complex services to the active agents and organizations. Basically, its functionality is like a yellow page service and a service descriptor in charge of providing a green page service; (ii) *Organization Manager Service* (OMS), it is mainly responsible of the management of the organizations and their entities. Thus, it allows the creation and the management of any organization; (iii) *Platform Kernel* (PK), it maintains basic management services for an agent platform. The following sections describe the different components of the THOMAS architecture detailing their offered services.

3 Service Facilitator

The SF is a mechanism and support by which the organization and agents can, at the same time, offer and discover services. The SF provides a place in which the autonomous entities can register service descriptions as directory entries. The SF can find services searching for a given service profile or searching by the goals that can be fulfilled executing the service. This is done using the matchmaking and service composition mechanisms that are provided by the SF. The SF acts also as a yellow pages manager and in this way it can also find which entities provide a given service. A service represents an interaction between two entities, which are modeled as communications among independent processes. In our case, the Multi-agent Technology provides us with FIPA communication protocols which are well established mechanisms in order to standardize the interactions. In this way, every service has an associated protocol. The SF entries are service descriptions using the following structure:

<ServiceID, Providers, ServiceGoal, ServiceProfile>
Providers = <ServiceImpID, ProviderIDList, ServiceModel, ServiceGrounding> +
ProviderIDList = ProviderID +

- **ServiceID** is a unique service identifier.
- **Providers** is a set of tuples made up by a Providers identifier list (ProviderIDList), the service process model specification, and the particular instantiation of the service that is provided by these providers.
- **ProviderIDList** maintains a list of identifier of the service providers.
- **ServiceGoal** is a general definition of the goal that can be fulfilled executing the service. It provides a first abstraction level for service composition.
- **ServiceProfile**, specifies what the service does, in way readable for those agents that are searching information (or matchmaking agents which act as searching service agents). The ServiceProfile is specified using the OWL-S standard for service Profile definition is augmented with the following

attributes: *providerRole* and *clientRole*, which specify the roles of the entity which offer or require the service.

- **ServiceModel** details the semantic content for using the service, the situations in which the results are obtained, and, whenever it is required, the step by step processes to get these results. The ServiceModel is specified using the OWL-S standard.
- **ServiceGrounding** specifies in details how an agent can access the service. A grounding specifies a communication protocol, the message formats, and other specific details of the service such as the used port to contact the service. The ServiceGrounding is specified using the OWL-S standard augmented with the FIPA protocols.

The SF provides the following standard services:

1. **RegisterProfile**: it is used when an autonomous entity wants to register a service description. To do this the following structure has to be completed in order to provide the service description.

   ***RegisterProfile*(ServiceID ?sID, ServiceGoal ?sGoal, ServiceProfile ?sProfile)**

2. **RegisterProcess**: it is used when an agent wants to register a particular implementation of a given service. The ID of the service provider entity has to be specified.

   ***RegisterProcess*(ServiceID ?sID, ServiceModel ?sModel, ServiceGrounding ?sGrounding, EntityID ?ProviderID)**

   There could be several providers for the same service implementation. The providers can be added or modified calling the *AddProvider* and *RemoveProvider* services.

3. **DeregisterProfile**: it is used to delete a service description.

   ***DeregisterProfile*(ServiceID ?sID)**

4. **SearchService**: it searchs a service whose description satisfies the client request. The search process can use matchmaking, composition and other techniques to solve complex queries.

   SearchService(ServicePurpose ?sPurpose)

   where ServicePurpose is a general structure in which the request is stored. It can be expressed as a SeviceGoal, a ServiceProfile description or a combination of both.

5. **SearchProvider**: it is used to find a service provider for an specific service.

   ***SearchProvider*(ServiceID ?sID)**

6. **ModifyProfile**: it is used to modify the description (profile) of a registered service. The client specifies the part of the service to be modified. The ServiceId will not change.

   ***ModifyProfile*(ServiceID ?sID, ServiceGoal ?Sgoal, ServiceProfile ?Sprofile)**

7. **ModifyProcess**: it is used to modify the implementation of a registered service. The client specifies the part of the service to be modified. The ServiceId will not change.

   ModifyProcess**(ServiceID ?sID, ServiceModel ?Smodel, ServiceGrounding ?Sgrounding, EntityID ?ProviderID)**

   If more than one provider implements the service, then the implementation will not be modified.
8. **AddProvider**: adds a new provider to an existing service implementation.

   AddProvider**(ServiceID ?sID, EntityID ?ProviderID)**

9. **RemoveProvider**: it deletes a provider from a service implementation.

   RemoveProvider**(ServiceID ?sID, EntityID ?ProviderID)**

   If it is the last provider, the implementation is automatically erased. Furthermore, if this were the last implementation of the service, then the provider is alerted and it can deregister the service.

Summarizing, the SF supplies a set of standard services to manage the services provided by organizations or individual agents. These meta-services can also be used by the rest of THOMAS components to advertise their own services.

4 Organization Manager Service

This component is in charge of organizations life cycle management, including specification and administration of both their structural components (roles, units and norms) and their execution components (active entities in each moment). To carry out this management the OMS makes use of the following lists: (i) *UnitList*: it stores the list of existing units, together with their objectives, type and parent unit (*SuperUnit*); (ii) *RoleList*: is stores the list of roles defined in each unit and their attributes (accessibility, visibility, position and inheritance); (iii) *NormList*: it stores the list of norms defined in the system; (iv) *EntityPlayList*: it stores the list of units in which each agent has been registered as a member, together with its adopted roles inside.

OMS offers all services needed for a suitable organization performance. These services are classified as: structural services, that modify the structural and normative organization specification; and dynamical services, that allow agents to entry or leave the organization dynamically, as well as role adoption.

Structural services. The OMS provides a group of services for registering or deregistering structural components, specifically roles, norms and units. Also it offers services for informing about these components. A *role* represents a position inside the unit in which it is defined. It is related to some interaction norms, imposed by the unit structure and its concrete position inside the unit; and some behaviour norms, that specify its functionality (services that needs and offers), restrict its actions (prohibition, obligations and permissions) and establish the consequences of these norms (sanctions and rewards). Therefore, a *norm* indicates obligations, permissions and prohibitions of roles related to service registering, requesting and fulfilment; service composition, or quality of service results. Thus, a norm defines those restrictions that cannot be expressed by means of service preconditions or postconditions. Finally, a *unit* represents groups of agents and establishes the topological structure of the system. It is also a recursive concept, so units can be defined inside others units. It enables the representation of organizative structures like hierarchy, matrix, coalition, etc. Furthermore, it indicates which are the structural positions of the system (i.e. member, supervisor, subordinate), as well as the relationships among these positions imposed by the structure. Following, **register services** of structural components are described:

1. **RegisterRole:** service used for requesting the registration of a new role inside a unit. As input parameters, it requires the role identifier, the unit in which this role must be registered, its visibility (whether it is public or private), its accessibility (internal or external), its position (whether it inherits from "member", "supervisor", "subordinated"), as well as its parent role in the role hierarchy. Only role and unit identifiers are mandatory.

   RegisterRole(RoleID ?Role, UnitID ?Unit, Visibility ?Visible, Accessibility ?Accessible, Position ?position, IsA ?SuperRole)

2. **RegisterNorm:** used for requesting the registration of a new norm inside a unit. A norm definition includes which role it is addressed and which is its content (including deontic value, conditions, actions and associated sanctions or rewards). Optionally, it also indicates which role is in charge of the fulfilment of the norm (*issuer*), who will carry out the sanction (*defender*) and who is in charge of its reward (*promoter*).

   RegisterNorm(NormID ?NID, AddressedRole ?Role, Content ?Cont, Issuer ?IssuerRole, [Defender ?DefenderRole], [Promoter ?PromoterRole])

3. **RegisterUnit:** used for requesting the registration of a new empty unit in the organization, with a specific structure, goal and parent unit.

   RegisterUnit(UnitID ?AID, UnitType ?Type, UnitGoal ?Goal, [UnitParent ?UnitParent])

All these structural services are implemented by means of the FIPA-Request protocol. Moreover, services for modifying component features might also be offered. For example, a service for changing the visibility value of a specific role.

On the other hand, OMS offers services for deregistration of structural components. These **deregister services** are:

1. **DeregisterRole:** used for requesting the deregistration of a role. There must not be any agent playing this role nor any norm addressed to it.

   DeregisterRole(RoleName ?Role, UnitID ?Unit)

2. **DeregisterNorm:** used for deleting a norm. The role that requests this service should be the issuer of the norm, that is, the controller of the norm.

   DeregisterNorm(NormID ?NID)

3. **DeregisterUnit:** service used for deleting a unit. This unit must be completely empty, without agents, nor roles or units inside. If the *UnitParent* input parameter is not given, it is assumed that the unit belongs to a "virtual" unit created by the agent platform.

   DeregisterUnit(UnitID ?UID, [UnitParent ?UnitParent])

All these deregister structural services are also implemented by means of the FIPA-REQUEST protocol.

Information services offered by OMS, provide specific information of all components of the organization and they might be restricted to some internal roles of the system. Following, the set of informative services is detailed:

1. **InformAgentRole:** service used for requesting the list of roles and units in which an agent is in a specific moment.

   InformAgentRole(AgentId ?AID)

2. **InformMembers:** used for requesting the list of entities that are members of a specific unit. Optionally, it is possible to specify a role of this unit, so then only members playing this role are detailed.

   InformMembers(UnitID ?Unit [,RoleID ?Role])

3. **QuantityMembers:** used for requesting the number of current members of a specific unit. Optionally, if a role is indicated then only the quantity of members playing this role is detailed.

   QuantityMembers(UnitID ?Unit [,RoleID ?Role])

4. **InformUnit:** used for requesting information about a specific unit that has been registered.

   InformUnit(UnitID ?Unit)

5. **InformRole:** used for requesting the list of roles that have been registered inside a unit.

   InformRole(UnitID ?Unit)

6. **InformProfile:** used for requesting the list of profiles associated to a specific role, according to the norms assigned to this role. Those norms specify its functionality.

   InformProfile(RoleID ?Role)

7. **InformNorm:** used for requesting the list of norms addressed to a specific role. This service accesses to the *NormList*.

   InformNorm(RoleID ?Role)

All information services are implemented by means of the FIPA-Query protocol.

Dynamic Services. OMS offers a set of basic composed services for dynamical role adoption and entry/exit of unit members. Most of these basic services are not directly accessible for agents, but are combined through compound services.

Basic services for role adoption are:

1. **RegisterAgentRole:** used for registering a new item in *EntityPlayList*, indicating that an agent plays a specific role inside a unit. This service is not directly published in the SF.

   RegisterAgentRole(AgentID ?AID, RoleId ?Role, UnitID ?UID)

2. **DeregisterAgentRole:** used for deleting an item in *EntityPlayList*, so then a specific agent does not play the role in the unit anymore. This service is not directly published in the SF.

   DeregisterAgentRole(AgentID ?AID, RoleId ?Role, UnitID ?UID)

OMS also offers a set of compound services that can be used by agents for adopting roles, leaving them and apply sanctions. Following, these compound services are related: (i) **AcquireRole:** serviced used for acquiring a role in a specific unit; (ii) **LeaveRole:** service used for leaving a role inside a specific unit; (iii) **Expulse:** service for forcing an agent to leave a specific role.

To sum up, the OMS is responsible for managing the life-cycle of the organizations. Thus, it includes meta-services for defining structural components of organizations, i.e. roles, units and norms. These components could be dynamically modified over the lifetime of the organization. Moreover, it includes services for creating new organizations (i.e. creating new units), admitting new members (i.e. acquiring roles) and member resigning (i.e. expulsing or leaving roles).

5 Platform Kernel

This component is in charge of providing the usual services required in a multi-agent system. Therefore, it is responsible for managing the life cycle of the agents included in the different organizations, and also allows to have a communication channel (incorporating different message transport mechanisms) to facilitate the interaction among different entities. On the other hand, the PK offers a safe connectivity and the necessary mechanisms that allow multi-device interconnectivity. The services offered are in most cases FIPA legacy with some modifications.

The THOMAS approach does not pursue the development of a new multi-agent platform kernel. The services required are a subset of the services typically offered by diverse well-known agent platforms. So, PK functionalities can be provided by any agent platform which offers the minimum services mentioned above. Moreover, the chosen agent platform must offer an appropriate communication mechanism which at least offers FIPA compatibility.

6 Conclusions

An important aspect for the development of open multi-agent systems is to provide developers with methods, tools and appropiated architectures which support all the requirements for this kind of systems. This document has deepened into this problem trying to propose an abstract architecture for the development of virtual organizations. Moreover, the proposal tries to raise a total integration of two promising technologies, that is, multi-agent systems and service-oriented computing. In THOMAS architecture agents can offer and invoke services in a transparent way to other agents or entities, as well as external entities can interact with agents through the use of the offered services.

This architecture is the first step in order to obtain truly deployed virtual organizations. Currently, a detailed description of the THOMAS architecture, examples and the first version of an implementation of the architecture is available in *http://www.dsic.upv.es/users/ia/sma/tools/Thomas*. This implementation includes the whole set of services described in the abstract architecture, with a basic support for norm management.

References

1. Argente, E., Julian, V., Botti, V.: Multi-agent system development based on organizations. Electronic Notes in Theoretical Computer Science 150, 55–71 (2006)
2. Argente, E., Palanca, J., Aranda, G., Julian, V., et al.: Supporting agent organizations. In: Burkhard, H.-D., Lindemann, G., Verbrugge, R., Varga, L.Z. (eds.) CEEMAS 2007. LNCS, vol. 4696, pp. 236–245. Springer, Heidelberg (2007)
3. Dignum, V., Dignum, F.: A logic for agent organization. In: Proc. FAMAS@Agents 2007 (2007)
4. Dignum, V., Vazquez-Salceda, J., Dignum, F.: Omni: Introducing social structure, norms and ontologies into agent organizations. In: Bordini, R.H., Dastani, M., Dix, J., El Fallah Seghrouchni, A. (eds.) PROMAS 2004. LNCS (LNAI), vol. 3346, pp. 181–198. Springer, Heidelberg (2005)
5. Esteva, M., Rodriguez-Aguilar, J., Sierra, C., Arcos, J., Garcia, P.: On the Formal Specification of Electronic Institutions. In: Sierra, C., Dignum, F.P.M. (eds.) AgentLink 2000. LNCS (LNAI), vol. 1991, pp. 126–147. Springer, Heidelberg (2001)
6. Ferber, J., Gutknecht, O.: A meta-model for the analysis and design of organizations in multi-agent systems. In: Proc. of the Third International Conference on Multi-Agent Systems, pp. 128–135. IEEE Computer Society, Los Alamitos (1998)
7. Greenwood, D., Calisti, M.: Engineering web service - agent integration. In: IEEE Int. Conf. on Systems, Man and Cyb., vol. 2, pp. 1918–1925 (2004)
8. Horling, B., Lesser, V.: A survey of multiagent organizational paradigms. The Knowledge Engineering Review 19, 281–316 (2004)

9. Horling, B., Lesser, V.: Using ODML to Model Multi-Agent Organizations. In: Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology (2005)
10. Jennings, N.R., Wooldridge, M.: Agent-oriented software engineering. In: Handbook of Agent Technology (2002)
11. Lopez, F., Luck, M., d'Inverno, M.: A normative framework for agent-based systems. Computational and Mathematical Organization Theory 12, 227–250 (2006)
12. Odell, J., Nodine, M., Levy, R.: A Metamodel for Agents, Roles, and Groups. In: James Odell, J.M., Giorgini, P. (eds.) Agent-Oriented Software Engineering (AOSE) V. LNCS. Springer, Heidelberg (2005)
13. Zambonelli, F., Parunak, H.: From design to intention: Signs of a revolution. In: Proc. 1st Int. Joint Conference on AAMAS, pp. 455–456 (2002)

Using Rest Class and Control Paradigms for Brain Computer Interfacing

Siamac Fazli[1], Márton Danóczy[1], Florin Popescu[2], Benjamin Blankertz[1,2],
and Klaus-Robert Müller[1]

[1] Berlin Institute of Technology, Machine Learning group, Franklinstr. 28/29, Berlin
[2] IDA group, Fraunhofer FIRST, Kekuléstr. 7, Berlin

Abstract. The use of Electro-encephalography (EEG) for Brain Computer Interface (BCI) provides a cost-efficient, safe, portable and easy to use BCI for both healthy users and the disabled. This paper will first briefly review some of the current challenges in BCI research and then discuss two of them in more detail, namely modeling the "no command" (rest) state and the use of control paradigms in BCI. For effective prosthetic control of a BCI system or when employing BCI as an additional control-channel for gaming or other generic man machine interfacing, a user should not be required to be continuously in an active state, as is current practice. In our approach, the signals are first transduced by computing Gaussian probability distributions of signal features for each mental state, then a prior distribution of idle-state is inferred and subsequently adapted during use of the BCI. We furthermore investigate the effectiveness of introducing an intermediary state between state probabilities and interface command, driven by a dynamic control law, and outline the strategies used by 2 subjects to achieve idle state BCI control.

1 Introduction

Non-invasive Brain Computer Interfacing (BCI) has recently become a hot topic with research activities outside its traditional fields medicine, psychology, neuroscience and rehabilitation engineering [11,37,18]. As many novel applications beyond rehabilitation have emerged [23,11,17] also other disciplines such as computer science have started to contribute with novel signal processing, machine learning, software and man machine interaction concepts [3]. Furthermore novel sensors, amplifiers and open source software [30,32] have increased the ease of handling BCIs and have therefore lowered the overall threshold for new groups to enter this highly interdisciplinary field of neurotechnology. In particular employing machine learning techniques allows a successful BCI communication for novices even from the first session [2,4]: instead of several hundred hours of subject training now the machine learns to decode the brain states of BCI users individually [2,4,11]. This concept of 'letting the machine learn' (instead of the subjects) was introduced by the Berlin Brain Computer Interface, adapting feature extraction and classification to data acquired in a so-called brief calibration phase (less than 5 minutes) where the subject is focussing to reproducibly generate certain brain states, e.g. imagery movements. The learning machine computes

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 651–665, 2009.
© Springer-Verlag Berlin Heidelberg 2009

a statistical estimator from this calibration data which then allows to discriminate these learned brain states during the feedback part of the experiment where the subject can communicate with the machine by power of though only (see [4,15]).

The practical versatility of BCI – this novel additional modality of interaction between man and machine – is yet far from explored [11,23,3,17]. Note however that despite its bright future perspectives EEG based BCI faces a number of challenges that we would like to discuss following [25,11].

1.1 Challenges in BCI

First of all, the information transfer rate (ITR) achievable through EEG is approximately one order of magnitude lower than the one observed by invasive methods in monkey studies [29,24,34]. That said, the potential benefits of brain implant based BCI has so far not been demonstrated to be worth the associated cost and risk in the most disabled of patients, let alone in healthy users [13]. EEG seems for now the only practical brain-machine interaction (BMI) choice (cost and ITR limitations hamper other non-invasive methods).

The most elementary of EEG-BCI challenges for healthy users is not – at first glance – a computational one. Standard EEG practice involves the tedious application of conductive gel on EEG electrodes in order to provide for accurate measurements of the micro-volt level scalp potentials that constitute EEG signals. Without "dry-cap" technology the proper set-up of BCI sessions in, say, a home environment, is too tedious, messy and therefore impractical. Marketing promises of impending "dry-cap" EEG have already made some media impact, while we have also presented "dry-cap" EEG-BCI design and performance in a controlled study [28]. All foreseeable systems, for reasons of ease-of-use and cost, use fewer electrodes than found on standard EEG caps. The computational challenges which we have addressed are (1) optimal placement of the reduced number of electrodes and (2) robustness of BCI algorithms to the smaller set of recording sites. With only 6 uni-polar electrodes we can achieve about 70% of full gel cap BCI performance at sites above the motor cortex, while being able to discount any potential influence of muscle and eye movement artifacts. Most other dry-cap challenges remaining are of an engineering design nature, excluding perhaps the computational reduction of artifacts produced not by unrelated electro-physiological activity but by measured low-frequency voltage variations caused by the physical movement of the head.

A long-standing problem of BCI designs which detect EEG patterns related to some voluntarily produced brain state is that such paradigms work with varying success among subjects/patients. We distinguish mental task based BCI such as "movement imagination" BCI from paradigms based on involuntary stimulus related potentials such as P300 which are limited to very specific applications such as typing for locked-in patients and require constant focus on stimuli extraneous to the task at hand. The peak performance to be achieved even after multiple sessions, varies greatly among subjects. Using a recent study [6] and other unreported data by many research groups, we estimate that about 20%

of subjects do not show strong enough motor related mu-rhythm variations for effective asynchronous motor imagery BCI, that for another 30% performance is slow (<20 bits/min) and for up to 50% it is moderate to high (20-35 bits/min). It is still a matter of debate as to why BCI systems exhibit "illiteracy" in a significant minority of subjects and what in terms of signal processing and machine learning algorithms can be done about it. Our first steps using a fully adaptive approach for calibration seem promising.

So far most BCI systems are capable of performing asynchronously (self-paced), say, cursor control, however this can become tiring since the user is required to continuously imagine one of the two classes. Often the user does not intend to control anything but rather leave the BCI inactive, thus BCI usability would benefit from an "idle" or "rest" class where the cursor does not respond when no active class (from a set of two or more) is activated, on top the BCI being self-paced. The "idle" state may take one of two forms: a "relax" state where the subject stays still and tries to "think of nothing", or can do almost any other mental task than those which belong to the active classes, but the latter may not be intuitive. Remaining challenges are to find a classifier that can induce rest state without a "relax" cue and to optimize the relationship between classifier output and BCI command. Due to physiological variations in background EEG activity, where a main factor is fatigue, we believe that the introduction of a controller layer are necessary for maximal performance.

The remainder of the paper will discuss both rest class and control strategies in more detail. Note that we are well aware to only report first *conceptual* steps in this paper and empirical evidence drawn from two subjects only. Clearly, a broader exploration of this interesting direction will still have to come, along with larger subject studies.

1.2 Background on Rest Class and Controller Concepts

Relatively few results showing idle-state asynchronous BCI exist. Early neurofeedback studies did feature an "idle" state, but required months of training on the part of subject to learn to modulate the slow cortical potentials required for classification and exhibited relatively low Information Transfer Rate (ITR) [1].

Among "rest-class" studies based on movement imagination, which offers intuitive control for gaming and prosthetics, the important measures of interface quality are the false positive (FP) rate, which is the probabilty of choosing an active class during "rest", the true positive rate (TP), meaning the probability of detecting the correct active class when movement imagination occurs or should occur. Other than accuracy measures, speed related measures of interest are the maximal time duration of correct idle state detection and the latency of correct responses to active states. Birch et al. [8,20] used a single active class vs. rest paradigm with FP rates of 3%, TP rates of 70%, and idle state lengths of 7s in multiple normal subjects, and more limited success in spinal-cord injury patients, using nearest-neighbor classifiers on windowed band-power features. Also Millán et al. [9,10] have implemented a 2-active class rest-class interface based on Malahobnis distance classification of band-power features. After an earlier study in which rest-state was performed with eyes closed (doing so produces distinctive alpha wave EEG modulation), a follow-up study used "active relaxation" [10] and showed FP of 2%, TP of 70%, and 1.5s of relax state. In the last specialized BCI data competition [5] a synchronous "relax"-vs.-active data set was addressed by the winning submission described in [35].

The type of classifiers currently employed in BCI (linear discriminants (LDA) and combinations thereof, cf. [21]) may partition the feature space in an overly simple manner, not allowing for "rest" class to be separated from active class regions with appropriate efficiency. To take a first step beyond linear methods we assigned continuous class membership probabilities, using compact principal component bandpower features instead of channel bandpower features as in [10], and added an intermediate, continously moving state which is driven by classifier output via a differential equation, which we call a "control law", such that when that intermediate state reaches a threshold, the output of the BCI interface changes. In the current approach, the classifier outputs a set of distances from each active class mean (or equivalently, a probability measure centered on the mean). We present feedback data using two active classes (left hand vs. right hand – or foot – movement imagination, "L" and "R") and an idle state. For the "active" classes the probabilities are of elliptical, Gaussian shapes. A given state (set of features) is either of higher probability (i.e. closer) to state L or R or far from both. Outside of both volumes is an outlier class, or the idle state. Stipulating that the idle state itself can be anywhere outside the "active" zones but also knowing where it lies preferentially, we place a "prior" on it as well - in between the active class zones, based on training data for "active" classes only. Training trials are recorded during cues to the subject to imagine "left" or "right" hand movements: no "idle" or "relax" trials are cued or recorded, and therefore the paradigm was not tuned to active relaxation. In subsequent feedback trials, i.e. actual asynchronous BCI control that is cued so that performance can be measured, all 3 probability distributions adapt and overlap, and the resulting posterior probabilities are fed into the "control law". Thus the subjects adapt their strategies based on feedback they receive while the classifier adapts also.

Post-hoc analysis was performed and is presented which outlines more efficient extraction of control signals could be achieved, for "rest" class inference as well as multi-class classification (i.e. for more than two mental tasks) by optimizing the control law on collected data. Also, we investigate the role of occipital alpha band power (a frequency range of 8-12 Hz around the cortical location related to visual processing) in the strategies used by two different subjects in achieving rest-class control. An increase in occipital alpha activity is traditionally associated with lack of attention to visual stimuli [27], be it because of closing of the eyes while awake, in some phases of sleep, during a state of drowsiness, or during a general lack of focus on sensory stimuli, which occurs when subjects actively relax or do not perform any cognitive tasks [27,36]. We used this phenomenon to help determine if the subjects "actively relaxed" or tried to modulate their active state EEG patterns more discernibly.

2 Methods

2.1 Experimental Paradigm

A 1D cursor control paradigm was reproduced in this study with two healthy volunteer subjects (both male). A standard DC amplifier set-up (BrainAmp-128DC, Munich, Germany) was used (64 EEG channels, 4 bipolar EMG/EOG channels). In the first part of the experiment ("calibration session"), a sequence of 70 left/right cues was presented visually by means of a letter which appears in the middle of the computer screen. The subject was asked to imagine the cued class without moving either limbs or eyes. Apart from off-line checks, electromyograms and electro-oculograms were monitored.

In a second part of the experiment subjects were asked to move a cross displayed on the screen to a target represented by a bar on either the right or left side of the screen by imagining the corresponding class. The cross movement provided continuous performance feedback to the subjects. After training, 70 trials of normal, synchronous left/right target feedback trials were performed (no rest state, cursor moved as soon as the target was presented): this was done to ensure quality of common spatial patterns as well as to compare "usual" BCI performance to asynchronous performance. Cross-validation with LDA classifiers was used on the resulting Common Spatial Patterns (CSP) [7,19,14] only on training trials.

The subject then performed 80 idle-to-active feedback trial attempts, in 4 blocks of 20 trials each. Each trial within the feedback phase consists of two separate parts. Within the first part no cue is presented to the subject for a period randomly varying (first subject: 8–12 sec, second subject: 10–20 sec), while the subject is instructed to relax. During this period the cursor is free to move, but not shown to the subject.

In general the cursor moves on a horizontal line, the center representing the relaxed state and respective edges active classes, e.g. left and right or left and foot. Not providing cursor feedback to the subject during this first part prevents the subject from steering the cursor stably into the middle by imagining movements. If during this "relax" period the subject reaches any target – without being aware of it – , an error is recorded and a new trial attempt begins. If neither of the two active classes are selected during the "relax" cue, the feedback progresses to the second part, where a visual cue (target on the left or right edge of the window) is presented, indicating the desired active class. In this second part the cursor is made visible to the subject. There is no time limit as to when a class has to be selected, but the subject is instructed to reach the cued target as fast as possible. After selection of any of the two active classes (the correct one or incorrect one) a new trial begins. The cursor does NOT represent an object whose position the subject is trying to control, it is merely a 1-D intermediate "state" which allows the subject to output a command when its position reaches the "left" or "right" boundaries.

2.2 Feature Extraction

A semi-automatic search for the time interval of the event-related desynchronization (ERDs) and frequency band whose power discriminates most between classes for each subjects generally selects the so-called mu- and beta-rhythms (10-25 Hz) in the motor cortex [2,26]. Note that the lower part of this frequency range (the mu range) slightly overlaps the alpha range (8-12) Hz and that due to brain and scalp electrical conductivity EEG signals are known to be spatially "mixed", that is, a single source of electrical acitvity in the cortex can be recorded at multiple locations on the scalp, albeit with higher amplitudes at electrodes close to that cortical location. The discriminating frequency band search determined a band-pass filter which attenuated signal amplitude outside these bands thereby accomplishing a temporal "demixing". The resulting filtered multivariate signals, segmented in the ERDs time interval, are used to compute two covariance matrices Σ_1 and Σ_2 from the calibration data. The CSP algorithm (see e.g. [7,19,14]) searches for a matrix W and a vector of n values $0 \leq d_i \leq 1$ which achieves:

$$W\Sigma_1 W^\top = D \quad \text{and} \quad W\Sigma_2 W^\top = I - D, \tag{1}$$

where n is the number of channels and D is a diagonal matrix with entries d_i. Using z-transform notation for digital signals, for any trial, the spatio-temporally de-mixed data is:

$$\boldsymbol{f}(z) = W\,H(z)\,\boldsymbol{s}(z) \tag{2}$$

Where $\boldsymbol{x}$ is the raw EEG signal and $H(z)$ is a diagonal matrix of identical band-pass filter transforms. The columns of the source to signal transform W^{-1} are called the Common Spatial Patterns (CSPs). The CSP decomposition can be thought of as a coupled decomposition of 2 matrices (for 2 classes) similar to a principal components analysis yielding eigenvectors and eigenvalues. As the eigenvalues d_i are equal to the power ratio of signals of class 1 by class 2 in the corresponding CSP filter (eigenvector in i-th column of matrix W), best discrimination is provided by filters with very high (i.e. near 1) or very low (i.e. near 0) eigenvalues. Accordingly CSP projections with the highest 2 and lowest 2 eigenvalues were chosen as features ($n = 4$).

2.3 Feature Processing

Let $\boldsymbol{x} \in \mathbb{R}^n$ the input feature vector as determined by CSP ($n = 4$) and $y \in \{1, 0, -1\}$ be the class label. The prior distributions $\mathrm{p}(\boldsymbol{x}|y = i, \boldsymbol{\mu}_i, \Sigma_i)$ for each class i are modelled as:

$$\mathrm{p}(\boldsymbol{x}|y = i, \boldsymbol{\mu}_i, \Sigma_i) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \Sigma_i)\ \forall\, i, \tag{3}$$

where $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)$ is the Gaussian density function with mean $\boldsymbol{\mu}$ and variance Σ. We chose the distributions' parameters $\Theta = \{\boldsymbol{\mu}_i, \Sigma_i\}$ by maximizing the logarithm of the data's joint likelihood:

$$\log \mathcal{L} = \sum_t \log \mathrm{P}(y_t, \boldsymbol{x}_t|\Theta) = \sum_t \log \frac{\mathrm{p}(\boldsymbol{x}_t|y_t, \Theta)\,\mathrm{P}(y_t)}{\mathrm{p}(\boldsymbol{x}_t)} \tag{4}$$

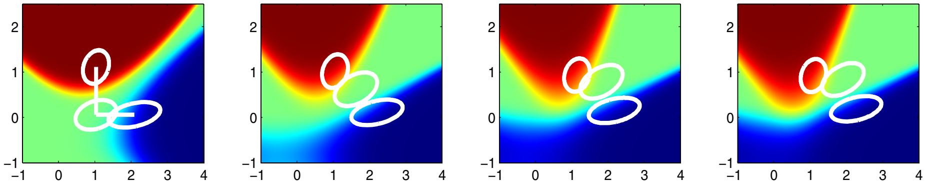


Fig. 1. Prior and posterior class membership probabilities during adaptation; after training, after 12, 24 and 36 trials, from left to right. In each panel, top white ellipse is left class, middle is rest class and bottom is right class 68% confidence region. The classifier output $[1 - \mathrm{P}(y = 0|\boldsymbol{x})]\,[\mathrm{P}(y = +1|\boldsymbol{x}) - \mathrm{P}(y = -1|\boldsymbol{x})]$, see eq. 11, is encoded as grayscale, with the light area having a high probability for "rest" and dark (blue, red) areas having high probabilities for "left" or "right". Only the largest and smallest CSP projections (the x and y axes of the plots above) are shown for visualization purposes, as they are the 2 most relevant of the 4 features. The two right-angled lines on the leftmost panel indicate the method of selecting the rest class prior mean: the minimum of movement class means in each CSP projection, corresponding to minimal bandpower difference.

The maximum likelihood estimators for the parameters of a Gaussian probability density function are given by the empirical mean $\hat{\boldsymbol{\mu}}$ and the empirical covariance matrix $\hat{\Sigma}$:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{x}_t \quad \text{and} \quad \hat{\Sigma} = \frac{1}{n}\sum_{t=1}^{n} (\boldsymbol{x}_t - \hat{\boldsymbol{\mu}})\,(\boldsymbol{x}_t - \hat{\boldsymbol{\mu}})^\top \tag{5}$$

Training data give us the CSPs which decompose each training trial, yielding a population of 4-dimensional vectors $\boldsymbol{x}$ for each of the 2 movement (i.e. active) classes, by taking the logarithm of the variance of each of the 4 chosen CSP projections over a (moving) time window of 800ms.

As we have chosen not to collect training data for the rest class, we must instead initialize the parameters of its probability distribution heuristically. If in rest state, we expect both hemispheres to generate roughly the same power in the alpha and beta band. We therefore initialize each component of the rest state's mean at the minimum of the respective means of the two active classes for each CSP component (see Figure 1, leftmost panel). The rest state's initial covariance matrix is set to the average of the covariance matrix of the two active classes.

2.4 Adaptation

In order to track nonstationarities in feedback BCI performance, we adapt by iteratively adding each new data point to the empirical means and covariance matrices. We make use of the sequential formulas for updating means:

$$\boldsymbol{\mu}_{n+1} = \frac{1}{n+1}\sum_{t=1}^{n} \boldsymbol{x}_t + \frac{1}{n+1}\boldsymbol{x}_{n+1} = \frac{n}{n+1}\boldsymbol{\mu}_n + \frac{1}{n+1}\boldsymbol{x}_{n+1} \tag{6}$$

Substituting the fraction $1/(n+1)$ by the constant α we introduce a forgetting factor (i.e. we compute leaky average):

$$\boldsymbol{\mu}_{n+1} = (1-\alpha)\,\boldsymbol{\mu}_n + \alpha \boldsymbol{x}_{n+1} \tag{7}$$

For the covariance matrix Σ, we implement a similar procedure (covariance estimation with forgetting factor). In the following, the notation $\boldsymbol{x}^{2\top} = \boldsymbol{x}\,\boldsymbol{x}^{\top}$ signifies a vector's outer product with itself.

$$\Sigma_{n+1} = \frac{1}{n+1}\sum_{t=1}^{n}\left(\boldsymbol{x}_t - \boldsymbol{\mu}_{n+1}\right)^{2\top} + \frac{1}{n+1}\left(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_{n+1}\right)^{2\top} \tag{8}$$

Substituting the leaky mean estimate for $\boldsymbol{\mu}_{n+1}$ from eq. 7 yields:

$$\Sigma_{n+1} = \frac{n}{n+1}\Sigma_n + \left(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n\right)^{2\top} \times \left[\alpha^2 \frac{n}{n+1} + (1-\alpha)^2 \frac{1}{n+1}\right] \tag{9}$$

For the sequential estimation of Σ, too, we introduce a constant β for $1/(n+1)$, thereby weighting data points with a possibly different forgetting factor as above:

$$\Sigma_{n+1} = (1-\beta)\,\Sigma_n + \left(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n\right)^{2\top} \times \left[\alpha^2\,(1-\beta) + (1-\alpha)^2\,\beta\right] \tag{10}$$

Thus the pdf's of the active classes adapt to feedback trials in a straightforward fashion. The prior for the rest class (i.e the outlier class) is formed by choosing the mean as a combination of the active class means (See Figure 1) while the covariance is the mean of the two active class covariances. The adaptation proceeds similarly as for the active classes, described above.

2.5 Determination of Cursor Speed

The (left/right) cursor position c, which is also the "cross" sometimes presented to the subject and is the "intermediate state" given by the control law, is calculated by numerically integrating its time derivative, the cursor speed $\dot{c}$. The cursor speed depends directly on the current class membership probabilities:

$$\dot{c} = (1-p_0)\ s\ (p_{+1} - p_{-1}) - p_0\,\frac{c}{\tau} \tag{11}$$

where $p_i = \mathrm{P}(y = i|\boldsymbol{x})$.

This reduces to the simple rate control law $\dot{c} = s\ (p_{+1} - p_{-1})$ with the speed constant s if the probability of the rest state is zero. However, if the rest-state probability is non-zero, the exponential decay term $-p_0\,1/\tau\,c$ pulls back the cursor to its middle position $c = 0$ with the time constant τ, thereby effectively avoiding the accumulation of small movements to either side.

3 Results

The accuracy of classification for synchronous feedback is given in Table 1. Information Transfer Rates (ITR) of 11.7 and 7 bits/min were achieved by the two subjects. Table 2 summarizes the average results of the 4 individual asynchronous feedback sessions. Note that most asynchronous trials reach target under 5s for Subject A. The shape and adaptive response of the classifier are shown in Figure 1.

Table 1. Main classification results for 1D feedback. Band stands for the frequency band, selected by a heuristic. xval stands for cross-validation error of training trials and TP is the accuracy of the two active classes.

Subject	A	B
classes	left-right	left-foot
band [Hz]	[9 22.5]	[11 13]
xval [%]	1.5 ± 4.2	6.4 ± 5.6
time/trial [s]	2.23	2.98
TP [%]	94.3	88.9
ITR [bit/min]	11.7	7.04

Table 2. Idle state feedback results. Idle state timeout stands for the time the subject was required to stay in the idle state without activating any classes, FP stands for the percentage of trials, where an active class was selected during the required idle-state period. To illustrate effectiveness of adaptivity in our classifier, in FP* the first 5 trials were disregarded, showing higher performance after most of the adaptation occured.

Subject	A	B
Idle state timeout [s]	10 ± 2	15 ± 5
FP [%]	11.2	18.7
FP* [%]	7.8	13.7
TP [%]	100	96.9
Activation time [s]	6.00	17.24
Under 10sec [%]	91.3	55.3

3.1 Alpha Power

Alpha power was calculated from raw EEG data sampled at 100 Hz from the six occipital electrodes PO3, POz, PO4, O1, Oz, O2. After applying a a band-pass butterworth filter in the alpha band ranging between 8-12 Hz, the log variance of the resulting signal was calculated and averaged over channels over shifting

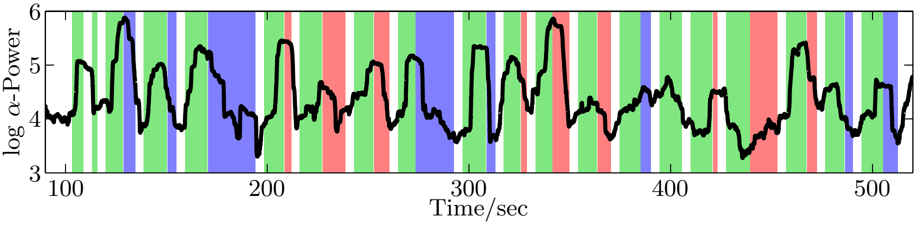


Fig. 2. Trace of occipital alpha for a sample experimental session. Vertical bars are cued rest (green) left (blue) and right (red) states.

Table 3. Per-class alpha power for both subjects. The column pVal shows the p-Value for the hypothesis that the alpha powers of each active class and the rest class are distributed with the same mean. The hypothesis can not be rejected at the 5% level for any of the classes and subjects.

Subject A				
class	N_{trials}	mean	stdev	pVal
left	34	4.63	0.69	0.36
right	37	4.59	0.68	0.23
rest	80	4.74	0.59	-

Subject B				
class	N_{trials}	mean	stdev	pVal
left	32	1.70	0.89	0.64
right	33	2.00	1.31	0.03
rest	80	1.63	0.52	-

causal time windows of 800ms (same windowing as in CSP calculation fed into the classifier). Then, mean and standard deviation of the average alpha power per trial was evaluated for each class (Table 3). The p-values shown for the t-test of the mean difference show no statistical difference between either active class and the rest class. Figure 2 shows actual traces of alpha power along with labeled bars indicating detected states for a typical session. The spaces between bars correspond to time between cued trials. Although alpha power is being modulated it is not significantly higher in the rest state than active states.

3.2 Post-hoc Optimization of Meta-parameters

We investigate whether we could have attained higher performance if we had chosen other meta-parameters, these being the decay time constant τ, the moving window size w and cursor speed constant s. For this the EEG recordings, transduced into CSP component traces, were fed off-line into a simulated controller and adaptive classifier having different meta-parameter values. Through a

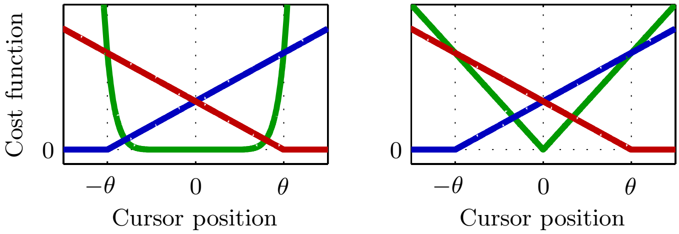


Fig. 3. The two cost functions used for optimization. A weighted integral of trajectory over time is computed with weights as functions of cursor state as shown for each class. The sum over all trials is minimized. Left "slow but accurate" cost function ($e_X = 10$, $e_{LR} = 1$), right panel "fast but less accurate" ($e_X = 1$, $e_{LR} = 1$) cost function for each of the three classes (left: blue, right: red, rest: green). Vertical scaling is arbitrary. The variable v (Eq. 12) is selected to scale the function at the threshold to equal values.

combination of brute-force parameter search followed by downhill-simplex minimization, the following two performance cost functions were evaluated:

$$\text{cost} \sim \sum_i \begin{cases} v\,|c|^{e_X} & \text{if } y_i = 0 \text{ (i.e. state 'X')} \\ \Theta(\theta + c)(\theta + c)^{e_{LR}} & \text{if } y_i = -1 \text{ (i.e. state 'L')} \\ \Theta(\theta - c)(\theta - c)^{e_{LR}} & \text{if } y_i = +1 \text{ (i.e. state 'R')}, \end{cases} \tag{12}$$

Optimization variables v, e_X and e_{LR} control the trade-off between accuracy and speed, Θ is the Heaviside step function (see Figure 3)

The resulting traces, before and after optimization, for Subject A are shown in Figure 4. The traces shown correspond to the intermediate state (the cursor). Intersection of treshold (θ) means that a L or R command is given. First note that one of the active states is detected much faster than the other, within 2s. This is consistent with both subjects reporting that they had less difficulty moving the cursor in one direction than another. The hypothetical trajectories that "would have been" from parameters optimized by the different cost functions differ. In one case (the first cost function) the rest state is more robust (i.e. the cursor trajectories are further from the treshold and the FP rate is lower) and using the second cost function the active classes would have responded faster, but with a rest class that is less robust.

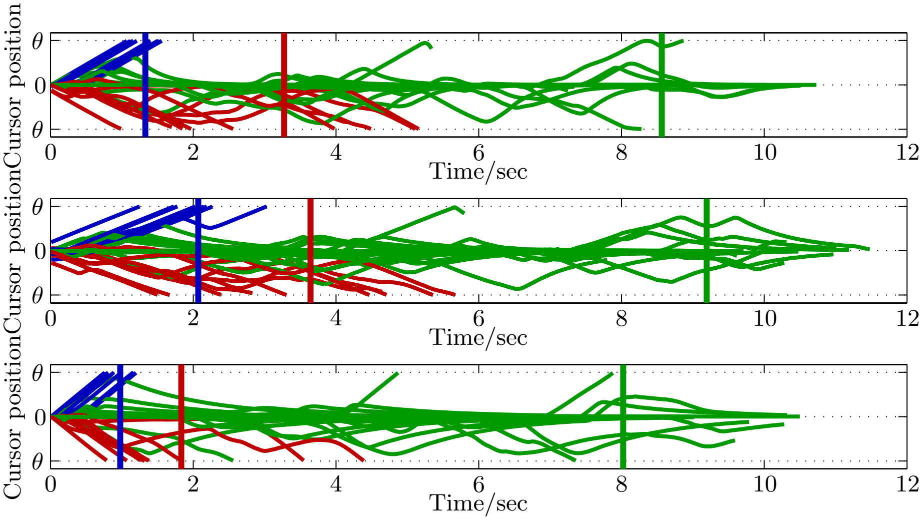


Fig. 4. (Data in Table 4) Top: original trajectories for one subject. Middle: "slow but accurate" optimized trajectories. Bottom: "fast but less accurate" optimized trajectories. The vertical bars indicate average time to hit in each class. Green: rest, red: right, blue: left. Note that in the middle panel there are no false positives, the rest trajectory which seems to hit threshold is in fact slightly below it.

Table 4. (Data for Figure 4) Control law parameters and results for original experiment (orig) and after the 2 optimizations, for accuracy (slow) and for speed (fast). The control law parameters are: Window size w [msec], speed constant (s), decay rate ($1/\tau$), TP rate, FP rate, activation time tAct [sec] and rest state duration tRst [sec]

	Subject A						
	w	s	$1/\tau$	TP	FP	tAct	tRst
orig	1000	0.70	0.20	100%	10%	2.30	8.56
slow	546	0.31	0.14	100%	0%	2.81	9.19
fast	1547	1.14	0.50	100%	20%	1.45	8.02

4 Conclusion and Outlook

The study reported success in an asynchronous feedback BCI paradigm using minimal subject training involving a "rest" class. It incorporated several features postulated to be important in "switch" type classification. One was the classifier type: adaptive as to account for rest state non-stationarities (see also [31,33]), and distance-based rather than linear-partition based such that it could classify "rest" state as outlier (distant) from either active state. Although the adaptation scheme used involves merely a leaky average of covariance and mean of class-conditional probability densities, our results show that performance increases quickly with practice. The free design parameters of this transduction algorithm are the coefficients of the implemented control law and the learning rates of the leaky running average.

Fortunately, the current heuristic search for the initial adaptibility meta-parameters was rather quick: an adaptive system needs to be neither excessively adaptive or inert with respect to the process it needs to track (in this case, the behavior of the subject), and for the control law, the cursor needed to be responsive but not so fast as to reach thresholds in less than 2-3 sec: data from prior experiments was used to choose these. Post-hoc analysis revealed that optimizing control law parameters for each subject could be useful, however there is a fundamental trade-off that is due to the integrative nature of the control law: one can make the interface more responsive but at a higher risk of sending inadvertent commands. Furthermore, the off-line analysis must be interpreted with caution: as users are active learners their behavior would adapt if the meta-parameters changed. Thus in future studies we aim to make this optimization run on-line, adaptively, such that the interface changes behavior gradually, allowing stable convergence of combined user and interface performance. Long term adaptation, over many sessions and many days, is also a future direction of research. An anecdotal note: the classifier was robust to the subject sometimes moving, or talking during the "rest" state, which suggests that this is in part an "idle" state.

The classification scheme we have employed makes "anything other than active state" detection possible, as it is based on "distance" from a particular mental state and an "outlier" cutoff or competition between distances. Future studies will therefore consider also non-parametric outlier models (e.g. [22]).

The benefit of adaptation and lack of explicit rest class training is the following: instead of collecting training data by instructing the subject to relax, or recording the other mental tasks which do not correspond to commands but which the subject may otherwise perform during BCI use, we only train the active classes, and through cued performance which provides feedback or performance, we allow the subject to find the approach that works best for him/her. The classifier then adapts to the strategy chosen or changes therein. Although they could have chosen to "think of nothing", and one of the subjects reported that he sometimes attempted to do so, this was not evidenced by the available physiological measure of active relaxation, namely occipital alpha-band EEG activity. This could be seen as further evidence that the rest state detected was an idle state and not a relaxation state - without objective measures, no matter what the subjects report it is difficult to tell between the two. It remains to be seen whether the optimized control law, then placed in an uncued "real-world" BCI use environment translates to effective rest-state BCI control which allows the subject to attend to other tasks than BCI commands. As such, new paradigms must be designed in which the BCI performance is quantified against various levels of background cognitive workload. Concluding, we envisage an EEG BCI scenario in which users purchase an affordable computer peripheral which is simply placed on the head and requires no gel. Novel users undergo a one-time calibration procedure which takes maximally 5 minutes, ideally even less [16,12]. They then proceed to use the BCI system in a game environment, to control a robot or wheelchair, and the performance of the system slowly adapts to the users' brain patterns, reacting only when they intend to control it. At each repeated use, parameters from previous sessions are recalled and re-calibration is rarely, if ever, necessary. We strongly believe such a system, capable of an average performance of about >20 bits/min, is achievable within the next few years. Clearly, challenges as the ones discussed above need to be met, in order to bring EEG BCI technology closer to becoming a commonplace computer peripheral.

Acknowledgments. The studies were partly supported by BFNT, BMBF FKZ 01IBE01A/B, by DFG MU 987/3-1 and by the EU under PASCAL2. This publication only reflects the authors' views.

References

1. Birbaumer, N., Hinterberger, T., Kübler, A., Neumann, N.: The thought-translation device (TTD): neurobehavioral mechanisms and clinical outcome. IEEE Trans. Neural Syst. Rehabil. Eng. 11(2), 120–123 (2003)
2. Blankertz, B., Dornhege, G., Krauledat, M., Müller, K.-R., Curio, G.: The non-invasive Berlin Brain-Computer Interface: fast acquisition of effective performance in untrained subjects. Neuroimage 37(2), 539–550 (2007)
3. Blankertz, B., Krauledat, M., Dornhege, G., Williamson, J., Murray-Smith, R., Müller, K.-R.: A note on brain actuated spelling with the Berlin Brain-Computer Interface. In: Stephanidis, C. (ed.) UAHCI 2007 (Part II). LNCS, vol. 4555, pp. 759–768. Springer, Heidelberg (2007)

4. Blankertz, B., Losch, F., Krauledat, M., Dornhege, G., Curio, G., Müller, K.-R.: The berlin brain-computer interface: Accurate performance from first-session in bci-naive subjects. IEEE Transactions on Biomedical Engineering 55(10), 2452–2462 (2008)
5. Blankertz, B., Müller, K.-R., Krusienski, D.J., Schalk, G., Wolpaw, J.R., Schlögl, A., Pfurtscheller, G., del, J., Schlögl, A., Pfurtscheller, G., Millán, J.d.R., Schrder, M., Birbaumer, N.: The BCI competition. III: Validating alternative approaches to actual BCI problems. IEEE Trans. Neural Syst. Rehabil Eng. 14(2), 153–159 (2006)
6. Blankertz, B., Sannelli, C., Halder, S., Hammer, E.M., Kübler, A., Müller, K.-R., Curio, G., Dickhaus, T.: Neurophysiological predictor of SMR-based BCI performance. Plosbiol (submitted, 2009)
7. Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., Müller, K.-R.: Optimizing spatial filters for robust eeg single-trial analysis. IEEE Signal Processing Magazine 25(1), 41–56 (2008)
8. Borisoff, J.F., Mason, S.G., Bashashati, A., Birch, G.E.: Brain-computer interface design for asynchronous control applications: improvements to the LF-ASD asynchronous brain switch. IEEE Trans. Biomed. Eng. 51(6), 985–992 (2004)
9. del Millán, J.R., Mourino, J.: Asynchronous BCI and local neural classifiers: an overview of the Adaptive Brain Interface project. IEEE Trans. Neural Syst. Rehabil. Eng. 11(2), 159–161 (2003)
10. del Millán, J.R., Renkens, F., Mourino, J., Gerstner, W.: Non-Invasive Brain-Actuated Control of a Mobile Robot by Human EEG. In: 2006 IMIA Yearbook of Medical Informatics, Schattauer Verlag (2006)
11. Dornhege, G., del Millán, J.R., Hinterberger, T., McFarland, D., Müller, K.-R. (eds.): Towards Brain-Computer Interfacing. MIT Press, Cambridge (2007)
12. Fazli, S., Popescu, F., Danóczy, M., Blankertz, B., Müller, K.-R., Grozea, C.: Subject independent mental state classification in single trials. Neural Networks, Special Issue: Brain Machine Interface (in review)
13. Hochberg, L.R., Serruya, M.D., Friehs, G.M., Mukand, J.A., Saleh, M., Caplan, A.H., Branner, A., Chen, D., Penn, R.D., Donoghue, J.P.: Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature 442, 164–171 (2006)
14. Koles, Z.J.: The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. Electroencephalogr Clin Neurophysiol. 79, 440–447 (1991)
15. Krauledat, M., Losch, F., Curio, G.: Brain state differences between calibration and application session influence BCI classification accuracy. In: Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course 2006, pp. 60–61. Verlag der Technischen Universität Graz (2006)
16. Krauledat, M., Tangermann, M., Blankertz, B., Müller, K.-R.: Towards zero training for brain-computer interfacing. PLoS ONE 3, e2967 (2008)
17. Krepki, R., Blankertz, B., Curio, G., Müller, K.-R.: The berlin brain-computer interface (bbci) — towards a new communication channel for online control in gaming applications. Multimedia Tools Appl. 33(1), 73–90 (2007)
18. Kübler, A., Kotchoubey, B., Kaiser, J., Wolpaw, J.R., Birbaumer, N.: Brain-computer communication: unlocking the locked in. Psychol. Bull. 127, 358–375 (2001)
19. Lemm, S., Blankertz, B., Curio, G., Müller, K.-R.: Spatio-spectral filters for improving the classification of single trial EEG. IEEE Trans. Biomed. Eng. 52, 1541–1548 (2005)

20. Mason, S.G., Birch, G.E.: A brain-controlled switch for asynchronous control applications. IEEE Trans. Biomed. Eng. 47(10), 1297–1307 (2000)
21. Müller, K.-R., Anderson, C.W., Birch, G.E.: Linear and nonlinear methods for brain-computer interfaces. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11(2), 165–169 (2003)
22. Müller, K.-R., Mika, S., Ratsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12(2), 181–201 (2001)
23. Müller, K.-R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., Blankertz, B.: Machine learning for real-time single-trial EEG-analysis: from brain-computer interfacing to mental state monitoring. J. Neurosci. Methods 167, 82–90 (2008)
24. Nicolelis, M.A.: Actions from thoughts. Nature 409, 403–407 (2001)
25. Nijholt, A., Tan, D., Pfurtscheller, G., Brunner, C., de Millán, J.R., Allison, B., Graimann, B., Popescu, F., Blankertz, B., Müller, K.-R.: Brain-computer interfacing for intelligent systems. IEEE Intelligent Systems 23(3), 72–79 (2008)
26. Pfurtscheller, G., Brunner, C., Schlögl, A., Lopes da Silva, F.H.: Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks. Neuroimage 31(1), 153–159 (2006)
27. Plotkin, W.B.: On the self-regulation of the occipital alpha rhythm: control strategies, states of consciousness, and the role of physiological feedback. J. Exp. Psychol. Gen. 105(1), 66–99 (1976)
28. Popescu, F., Fazli, S., Badower, Y., Blankertz, B., Müller, K.-R.: Single trial classification of motor imagination using 6 dry EEG electrodes. PLoS ONE 2(7), e637 (2007)
29. Santhanam, G., Ryu, S.I., Yu, B.M., Afshar, A., Shenoy, K.V.: A high-performance brain-computer interface. Nature 442, 195–198 (2006)
30. Schalk, G., Mcfarland, D.J., Hinterberger, T., Birbaumer, N., Wolpaw, J.R.: Bci2000: a general-purpose brain-computer interface (bci) system. IEEE Transactions on Biomedical Engineering 51(6), 1034–1043 (2004)
31. Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P., Müller, K.-R.: Towards adaptive classification for BCI. J. Neural Eng. 3, 13–23 (2006)
32. Sonnenburg, S., Braun, M.L., Ong, C.S., Bengio, S., Bottou, L., Holmes, G., LeCun, Y., Müller, K.-R., Pereira, F., Rasmussen, C.E., Rätsch, G., Schölkopf, B., Smola, A., Vincent, P., Weston, J., Williamson, R.: The need for open source software in machine learning. J. Mach. Learn. Res. 8, 2443–2466 (2007)
33. Sugiyama, M., Krauledat, M., Müller, K.-R.: Covariate shift adaption by importance weighted cross validation. Journal of Machine Learning Research 8, 985–1005 (2007)
34. Taylor, D.M., Tillery, S.I., Schwartz, A.B.: Direct cortical control of 3D neuroprosthetic devices. Science 296, 1829–1832 (2002)
35. Wang, Y., Zhang, Z., Li, Y., Gao, X., Gao, S., Yang, F.: BCI Competition 2003–Data set IV: an algorithm based on CSSD and FDA for classifying single-trial EEG. IEEE Trans. Biomed. Eng. 51(6), 1081–1086 (2004)
36. Williamson, S.J., Kaufman, L., Lu, Z.L., Wang, J.Z., Karron, D.: Study of human occipital alpha rhythm: the alphon hypothesis and alpha suppression. Int. J. Psychophysiol. 26(1-3), 63–76 (1997)
37. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clin. Neurophysiol. 113, 767–791 (2002)

The Training Issue in Brain-Computer Interface: A Multi-disciplinary Field

Ricardo Ron-Angevin[1], Miguel Angel Lopez[2], and Francisco Pelayo[3]

[1] Department of Electronics Technology,
University of Malaga
rron@uma.es
[2] Department of Signal Theory, Telematic and Communication,
University of Granada
malg@ugr.es
[3] Department of Computer Architecture and Computer Technology,
University of Granada
fpelayo@ugr.es

Abstract. A tough question to address would be who can be considered the inventor of the Brain-computer interface. Many researchers have contributed to the evolution of this concept, from the basis conception of the neurons, synaptic connections and rudimentary EEG measurement systems up to the use of virtual reality environments with modern wireless devices. In the middle, series of devoted people with the honourable aim of provide a better quality of life to disabled patients. In the course of this evolution a key point is the use of efficient techniques of mutual training for both the patient and the system. In this paper we introduce the multi-disciplinary nature of the Brain-computer interfaces, for later focus on training techniques used in those based in the sensorimotor rhythm.

Keywords: Brain-computer interface, training, virtual reality environment, sensorimotor rhythm.

1 Antecedents of the BCI

At the end of the XIX century, the Spanish doctor Santiago Ramon y Cajal came up with a revolutionary theory about the organization and structure of the mind, based in the fact that it is composed of series of independent cells, neurons, which communicate each other by means of functional contacts, synapses. This revolutionary theory was worth him to obtain the Nobel Prize in 1906. This and other relevant advances set up the basis for the electroencephalography (EEG).

Few years after, in 1924, the German psychiatrist Hans Berger, considered to be the father of the EEG, made the first record of EEG in a human being and afterwards in 1929 stated the existence of rhythms alpha and beta [1]. Afterwards the American William Grey Walter used cutting edge electronic to built, in 1957, the first brain topography machine, able to build maps of cerebral activity, revealed as a useful tool in the diagnosis of mental diseases (e.g. epilepsy).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 666–673, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Afterwards in the 60’s, the Japanese researcher Kasamatsu, verified that the transcendental meditation Zen carried out by Buddhist monks was able to voluntarily modify both the frequency and amplitude of the alpha wave as an indication of a specific change of consciousness [2]. In order to enhance this effect, this physiological information was fed back to the subject, in such a way that he could adjust it voluntarily. Neal Millar is considered the father of this technique called biofeedback. A pioneer study of modulation of the alpha wave and auditory biofeedback is that from Nowlis y Kamiya [3]. Afterwards, in 1974, Sterman used biofeedback to control the sensorimotor rhythm or mu rhythm [4].

The fact that a subject was able to voluntarily modulate his cerebral activity hence generating information, that this information could be registered by means of EEG, processed and translated to statements or commands over a device or computer and, finally, the subject could know the results of those commands by means of biofeedback and hence the loop could be closed; all together sets up the basis of a full-duplex communication system between the man and the machine.

Some researchers came up with definitions for this new communication system. The first contribution in 1973 by J. J. Vidal was "The BCI system is geared to use both the spontaneous EEG and the specific evoked responses triggered by time-dependent stimulation under various conditions for the purpose of controlling such external apparatus as e.g. prosthetic devices” [5]. But , perhaps the most popular one is the one from the researches J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, y T. M. Vaughan “A direct brain-computer interface is a device that provides the brain with a new, non-muscular communication and control channel” [6].

The next questions to address would be: Who can benefit of such a communication system? The answer can be found in subjects with some kind of sever motor impairment, such as subjects with amyotrophic lateral sclerosis or in a wheelchair. The communication is a process of cognitive nature, hence this communication should not be limited by second order issues, such as motor impairments, when the most important task is executed in the brain.

From these thoughts, from the innate desire to overcome the adversities and, specially, thanks to the effort of so many researchers, the Brain-computer interface is nowadays a multi-disciplinary reality. However to guarantee successful control of a BCI, many problems must be resolved, being the required training to operate a BCI, one of the most important.

2 Importance of Training Protocol

Some BCIs are based on an individual's capacity to control some features of EEG activity, which include a variety of different rhythms that are identified by their frequency and location, and are associated with various aspects of brain function. Performance of BCI will depend, especially, on the ability of the subject to control his or her own EEG patterns. To this end, it is necessary to provide suitable training, which can sometimes go on for several months [7]. It is also very important to provide some type of feedback allowing subjects to see their progress [8], [10].

Nowadays, the vast majority of research on the BCI field focuses on improving features extraction methods and classification algorithms. Emphasis is usually placed on the importance of developing training methods based on biofeedback techniques that would improve human performance [8],[9].

However, the purpose of BCI is not to select subjects with high EEG control ability and then train them to achieve greater classification accuracy. No matter how sophisticated the feature extraction procedures and classification algorithms are, BCI will not be useful if it cannot be used by subjects with serious motor function problems, who after all, are those the system is being designed for. If an inappropriate training programme is added to the difficulty of controlling EEG signals, the result will be frustration and early retirement for many subjects. To avoid this problem, working on a suitable training technique is considered necessary.

Few studies have dealt with training techniques, and those have tended to concentrate on the advantages of providing continuous or discrete feedback [10],[11]. In previous studies, to obtain results, classification accuracy is usually the only parameter considered, and not the effects that feedback can have on subjects. Besides the subject's capacity to control EEG signals, other psychological factors such as concentration, frustration, fatigue and motivation, which can affect EEG signals [12], should be considered to define a suitable training protocol. However, these factors are generally not taken into account when defining a suitable training protocol as a means to achieve a more effective learning.

The design of a training protocol should take into consideration several different aspects: 1) training paradigm timing, 2) presentation and type of feedback and 3) choice of the mental tasks to be carried out. The first includes: i) duration, repetitiveness and pause between trials, ii) duration of the mental activities, and iii) duration of the session. The second combines: i) sessions with or without feedback, ii) whether feedback is continuous or discrete, and iii) how the feedback is presented. Nowadays, conventional systems of feedback are based on cursor control and horizontal bar extension. Due to the lengthy training in BCI systems, this type of feedback may result tiring or somewhat boring. In order to improve the effectiveness of the training process and reduce training time, feedback needs to be attractive, thus motivating subjects to control their EEG signals. The third aspect, the choice of the mental tasks, is one of the most important to consider. The difficulty of the mental task has been shown to affect EEG signals [13], and in the performance of BCI.

In order to obtain reliable results and conclusions regarding what the best training techniques might be, is it not advisable to select only those subjects who have been previously trained, nor to consider failure and success rates as the only parameters when classifying mental tasks. Objective and subjective measures should be taken, based on the record of other psychophysiological signals, attention tests and questionnaires allowing subjects to describe their own impressions.

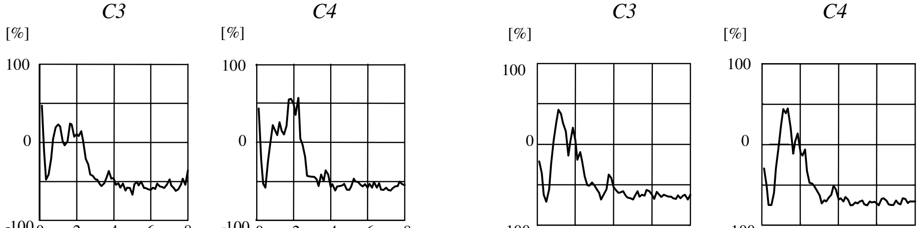
3 Choice of Mental Tasks

The performance of the BCI systems can depend on many parameters related to the training protocol, being one of these parameters, the different mental tasks to discriminate. Some BCIs discriminate between 3 [14], [15] and 5 different mental tasks [16]; however, the vast majority of them are based on the classification of 2 mental tasks. The fact is that increasing the number of different mental tasks to discriminate is an option to increase the number of commands. However, some studies have reported that an increasing number of classes resulted in a decrease of the classification accuracy [16], [17].

Little research has been conducted on the different mental tasks that may turn out to be more effective to manage and control a BCI system. Some training protocols consist of letting the subject find any mental task to control the EEG signals [18-20]; however, the majority of training protocol ask subject to employ specific mental task to produce different EEG patterns.

The motor imagery is the most commonly used mental task in BCI. Some BCI systems discriminate between imagination of right/left hand movements [11], [21], [22] or imagination of hand/foot movements [9]. Other systems discriminate between a motor imagery and another cognitive task, such as a math task [23], [24] or simply relaxing state [22], [25].

Even though motor imagery has proven to be a good option to control a BCI, it may be inappropriate for certain groups of subjects, for example for a person who has been paralyzed for many years or, indeed, from birth [25], [26]. Furthermore, for untrained subjects, combining different motor imagery tasks could not be the best option, as suggested in [22]. In this study, a group of 10 untrained subjetcs followed a training protocol which consisted of the discrimination between imagination of right and left hand movement. The discrimination between these mental tasks is based on amplitude changes of mu and central beta rhythm. These rhythms display an amplitude attenuation (Event-Related Desynchronization, ERD) or an amplitude enhancement (Event-Related Synchronization, ERS), over the contralateral hemisphere [27]. The majority of these subjects (8 subjects) obtained poor classification results, and although many of them showed, from the beginning of sessions, a clear ERD during the motor imagery, this was produced in both hemispheres independently of the mental task (Figure 1) and not in the contralateral hemisphere.



Imagination of right hand movement *Imagination of left hand movement*

Fig. 1. Time course of band power in the mu rhythm during imagination of right and left hand movement. The cue is presented at 3s and the mental tasks are carried out during the period 3-8s. Data from one untrained subject.

Difficulty of untrained subjects in producing contralateral desynchronization, suggested that these mental tasks was not suitable to learn to control a BCI system, causing them frequent frustration. In order not to discourage subjects and to avoid retirement, it could be advantageous to start training with two mental tasks that were easier to discriminate. These mental tasks could be relaxed state and imagination of right hand movement. As is showed in [22], with these mental tasks, a larger number of subjects got control in EEG signals, and training supposed less effort, less fatigue and more satisfaction for subjects.

Training protocols must not be chosen at random. On the contrary, in many cases, they must adapt to the subject to be effective, as is suggested in [16].

2 The Use of Virtual Reality to Improve Training Performance

Virtual reality is a powerful tool with graphical possibilities to improve BCI-feedback presentation. Virtual reality technology has the capability of creating immersive and motivating environments, which are very important in guaranteeing a successful training [28]. To prove this theory, a study has been carried out in our laboratory (DIANA research group) [29]. This one consisted of submitting different group of untrained subjects to a training protocol providing different types of feedback (Figure 2).

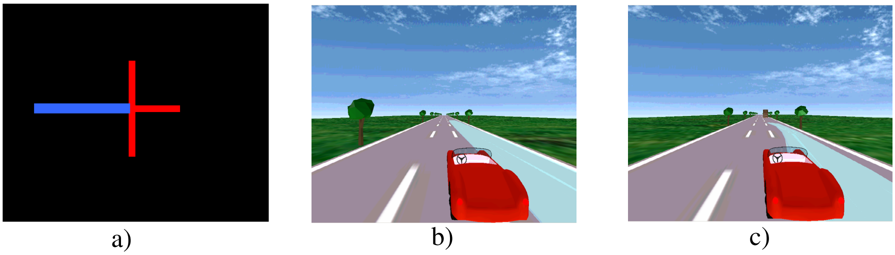


Fig. 2. Feedback base on: a) bar extension, b) controlling a car to avoid puddles and c) controlling a car to avoid puddle and obstacles located at the end (wall)

A group of subjects (group 1) was trained using a system equivalent to the Graz BCI [21] which used conventional feedback. This consisted of a horizontal bar, which extends in varying degrees to the right or to the left depending on the classification result (Figure 2a). Two other groups were trained using a system which submits subjects to a more familiar environment, such as controlling a car to avoid obstacles. For one group (group 2), the obstacle was a puddle which appeared in the left or right lane (Figure 2b). The subject controlled the movement of the car to the right or left, according to the mental task required in order to avoid the puddle. For the other group (group 3), the same paradigm was used but different obstacles were incorporated at the end of the puddle (Figure 2c). The training was carried out discriminating between mental relaxation and imagined right hand movements. The duration of each trial was 8 seconds, being the feedback period between 4.25 and 8s. In figure 3, the times course of the grand average error rate (average between error rates of each subject) for each group is showed.

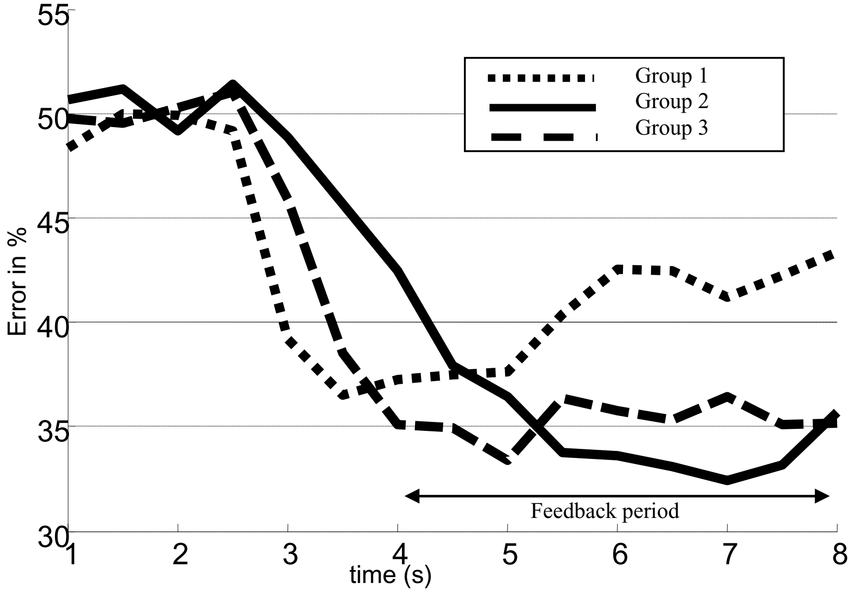


Fig. 3. Times course of the grand average error rate for each group (average of error time courses between all subjects of each group) during the trial time

Significant differences in feedback control were obtained, suggesting that an interface based on virtual reality techniques can improve the feedback control, specifically for untrained subjects. During feedback period (4.25-8 s), subjects of group 1 tended to have a higher error percentage and had more difficulty controlling the EEG signals, suggesting a lack of concentration, On the contrary, subjects of group 2 and 3 managed to decrease their error rate, achieving the lowest error rates during that same period. However, a worsening of this control was noticed during the last second of feedback (7-8 s) for subjects of group 2, due probably to the fact that subjects did not make an effort when they realised the puddle was almost behind them. The presence of an obstacle located at the end of the puddle seems to help subjects to maintain the control of the EEG signals until the end of the feedback period (group 3).

The obtained results suggest that EEG behaviour can be modified via feedback presentation. In BCI systems, training must be as easy and attractive as possible. The graphical possibilities of a multimodal interface combining 3D display and sound would be a good option in developing training techniques which help subjects to achieve a better control of the BCI.

Acknowledgments. This work was partially supported by the Innovation, Science and Enterprise Council of the Junta de Andalucía (Spain), project P07-TIC-03310.

References

1. Berger, H.: Uber das Electrenkephalogramm des Menchen. Arch. Psychiat. Nervenkr. 87, 527–570 (1929)
2. Kasamatsu, A., Hirai, T.: An electroencephalographic study on the zen meditation (Zazen). Folia Psychiatrica et Neurologica Japonica 20(4), 315–336 (1966)
3. Nowlis, D.P., Kamiya, J.: The control of electroencephalographic alpha rhythms through auditory feedback and the associated mental activity. Psychophysiology 6, 476–484 (1970)
4. Sterman, M.B., MacDonald, L.R., Stone, R.K.: Biofeedback training of sensorimotor EEG in man and its effects on epilepsy. Epilepsia 15, 395–416 (1974)
5. Vidal, J.J.: Toward Direct Brain–Computer communication. Annu. Rev. Biophys. Bioeng., 157–180 (1973)
6. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clin. Neurophysiol. 113, 767–791 (2002)
7. Wolpaw, J.R., Ramoser, H., McFarland, D.J., Pfurtscheller, G.: EEG- Based Communication: Improved Accuracy by Response Verification. IEEE Transactions on Neural Systems and Rehabilitation Engineering 6, 326–333 (1998)
8. Wolpaw, J.R., Birbaumer, N., Heetderks, W.J., McFarland, D.J., Peckham, P.H., Schalk, G., Donchin, E., Quatrano, L.A., Robinson, C.J., Vaughan, T.M.: Brain-computer interface technology: A review of the first international meeting. IEEE Transactions on Neural Systems and Rehabilitation Engineering 8, 164–173 (2000)
9. Guger, C., Edlinger, G., Harkam, W., Niedermayer, I., Pfurtscheller, G.: How many people are able to operate an EEG-based brain-computer interface (BCI)? IEEE Transactions on Neural Systems and Rehabilitation Engineering 9(1), 49–58 (2001)
10. McFarland, D., McCane, L., Wolpaw, J.: EEG-Based Communication and control: Short-Term Role of Feedback. IEEE Rehabilitation Engineering 6, 7–11 (1998)
11. Neuper, C., Schlögl, A., Pfurtscheller, G.: Enhancement of left-right sensorimotor EEG differences during feedback-regulated motor imagery. Journal Clinical Neurophysiology 16, 373–382 (1999)
12. Curran, E.A., Stokes, M.J.: Learning to control brain activity: A review of the production and control of EEG components for driving brain-computer interface (BCI) systems. Brain and Cognition 51, 326–336 (2003)
13. Gevins, A., Smith, M.E., McEvoy, L., Yu, D.: High-resolution EEG mapping of cortical activation related to working memory: effects of task difficulty, type of processing and practice. Cerebral Cortex 7, 375–385 (1997)
14. Millán, J., del, R., Mouriño, J., Marciani, M.G., Babiloni, F., Topani, F., Canale, I., Heikkonen, J., Kaski, K.: Adaptive brain interfaces for physically-disabled people. In: Proceeding of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Hong Kong, China, vol. 4, pp. 2008–2011 (1998)
15. Geng, T., Dyson, M., Tsui Chun, S.L., Gan John, Q.: A 3-class asynchronous BCI controlling a simulated mobile robot. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society, pp. 2524–2527 (2007)
16. Obermaier, B., Neuper, C., Guger, C., Pfurtscheller, G.: Information transfer rate in a five-classes brain-computer interface. IEEE Transactions on Neural Systems and Rehabilitation Engineering 9(3), 283–289 (2001)
17. Kronegg, J., Chanel, G., Voloshynovskiy, S., Pun, T.: EEG-based synchronized Brain-computer interfaces: A model for optimizing the number of mental tasks. IEEE Transactions on Neural Systems and Rehabilitation Engineering 15(1), 50–58 (2001)

18. Wolpaw, J.R., Mc Farland, D.J., Neat, G.W., Forneris, C.A.: An EEG-based brain-computer interface for cursor control. Electroencephalography Clinical Neurophysiology 78(3), 252–259 (1991)
19. Polak, M., Kostov, A.: Development of brain-computer interface: preliminary results. In: Proceeding of the 19th International Conference- IEEE/EMBS, Chicago, USA, pp. 1543–1546 (1997)
20. Birbaumer, N., Kübler, A., Ghanayim, N., Hinterberger, T., Perelmouter, J., Kaiser, J., Iversen, I., Kotchoubey, B., Neumann, N., Flor, H.: The thought translation device (TTD) for completely paralyzed patients. IEEE Transactions on Neural Systems and Rehabilitation Engineering 8(2), 190–193 (2000)
21. Guger, C., Schlögl, A., Neuper, C., Walterspacher, D., Stein, T., Pfurtscheller, G.: Rapid prototyping of an EEG-based brain-computer interface (BCI). IEEE Transactions on Neural Systems and Rehabilitation Engineering 11(2), 145–147 (2003)
22. Ron-Angevin, R., Díaz-Estrella, A.: Training protocol evaluation of a brain-computer interface: mental tasks proposal. Revista Neurología 47(4), 197–203 (2008)
23. Anderson, C.W., Stolz, E.A., Shamsunder, S.: Discriminating mental tasks using EEG represented by AR models. In: Proceeding of the 17th Annual Conference of the IEEE Engineering in Medicine and Biology Society, Montreal, Canada, pp. 20–23 (1995)
24. Penny, W.D., Roberts, S.J.: Experiments with an EEG-based computer interface. Technical Report, BCI Workshop, Albany, USA (1999)
25. Neuper, C., Müller, G.R., Kübler, A., Birbaumer, N., Pfurtscheller, G.: Clinical application of an EEG-based brain-computer interface: a case study in a patient with severe motor impairment. Clinical Neurophysiology 114(3), 399–409 (2003)
26. Curran, E.A., Aykacek, P., Stokes, M.J., Roberts, S.J., Penny, W., Johnsrude, I., Owen, A.M.: Cognitive tasks for driving a brain-computer interfacing system: a pilot Study. IEEE Transactions on Neural Systems and Rehabilitation Engineering 12(1), 48–54 (2003)
27. Pfurtscheller, G., Neuper, C., Flotzinger, D., Prezenger, M.: EEG-Based discrimination between imagination of right and left hand movement. Electroencephalography Clinical Neurophysiology 103, 642–651 (1997)
28. Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., Pfurtscheller, G.: Brain-Computer Communication: Motivation, aim and impact of exploring a virtual apartment. IEEE Transactions on Neural Systems and Rehabilitation Engineering 15, 473–482 (2007)
29. Ron-Angevin, R., Díaz-Estrella, A.: Brain-computer interface: Changes in performance using virtual reality techniques. Neuroscience Letters 449(2), 12–127 (2009)

A Maxmin Approach to Optimize Spatial Filters for EEG Single-Trial Classification

Motoaki Kawanabe[1,2], Carmen Vidaurre[2], Benjamin Blankertz[1,2],
and Klaus-Robert Müller[2]

[1] IDA group, FIRST, Fraunhofer Institute, Kekulestr. 7
12489 Berlin, Germany
[2] Berlin Institute of Technology, Computer Science Faculty, Machine Learning department,
Franklinstr. 28/29
10587 Berlin, Germany
`motoaki.kawanabe@first.fraunhofer.de`,
`vidcar@cs.tu-berlin.de`,
`blankertz@cs.tu-berlin.de`,
`klaus-robert.mueller@tu-berlin.de`

Abstract. Electroencephalographic single-trial analysis requires methods that are robust with respect to noise, artifacts and non-stationarity among other problems. This work contributes by developing a maxmin approach to robustify the common spatial patterns (CSP) algorithm. By optimizing the worst-case objective function within a prefixed set of the covariance matrices, we can transform the respective complex mathematical program into a simple generalized eigenvalue problem and thus obtain robust spatial filters very efficiently. We test our maxmin CSP method with real world brain-computer interface (BCI) data sets in which we expect substantial fluctuations caused by day-to-day or paradigm-to-paradigm variability or different forms of stimuli. The results clearly show that the proposed method significantly improves the classical CSP approach in multiple BCI scenarios.

1 Introduction

Feature extraction is an important prerequisite for analyzing high dimensional real world data. For single-trial electroencephalographic (EEG) classification tasks, spatial filters have become very popular feature extractors. Data driven approaches that optimize spatial filters for each subject individually have been proven useful [1], in particular in brain-computer interfaces (BCI), which translate the users intent (coded by a small set of mental tasks) into control actions such as computer applications or neuroprostheses [2,3,4,5].

In the past years machine learning methods have led to significant advances in the analysis and modeling of neural signals. While early EEG-BCI efforts required neurofeedback training on the part of the user that lasted on the order of days, in systems based on machine learning (ML) it suffices to collect examples of EEG signals in a so-called *calibration measurement* during which the user is cued to perform repeatedly a small number of e.g. motor imagery tasks [6,7,8]. This data is then used to adapt the system to the specific brain signals of each user (*machine training*). This step of adaption

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 674–682, 2009.
© Springer-Verlag Berlin Heidelberg 2009

seems to be instrumental for effective BCI performance despite the large inter-subject variability of the respective brain signals [9]. After this preparation step, which is very short compared to the complementary subject training in the operant conditioning approach [10], the feedback application can start. Here, the users can actually transfer information through their brain activity and control applications. In this phase, the system is composed of the classifier that discriminates between different mental states and the control logic that translates the classifier output into control signals, e.g. a cursor position or a selection from an alphabet.

There are several aspects in which BCI research can profit from improvement, see the ‘Challenges’ section of [11].

One of them is to make the system more *robust* against non task-related fluctuations and/or non-stationarity of the measured EEG signals. These fluctuations may be caused by changes in the subject’s brain processes, e.g. change of task involvement, fatigue etc., or by artifacts such as swallowing, blinking or yawning.

Recently Kim et al. [12] applied a maxmin approach to Fisher discriminant analysis (FDA) and proposed a novel robust classification method.

From their maxmin theorem, the maxmin FDA is guaranteed to have larger Rayleigh coefficients for any fluctuations within a prefixed tolerance set.

The present paper contributes by investigating a maxmin approach in the spirit of [12] to common spatial patterns (CSP) [13], which is one of the working horses for spatial filtering in BCI applications. In contrast to the FDA case, we can obtain the worst case covariance matrices analytically, which leads a modified generalized eigenvalue problem.

2 Sensorimotor Rhythms and Common Spatial Patterns

Apart from transient components, EEG comprises rhythmic activity located over various areas. Most of these rhythms are so-called idle rhythms, which are generated by large populations of neurons in the respective cortex that fire in rhythmical synchrony when they are not engaged in a specific task. Over motor and sensorimotor areas in most subjects oscillations with a fundamental frequency between 9 and 13 Hz can be observed, the so called μ-rhythm. Due to its comb-shape, the μ-rhythm is composed of several harmonics, i.e., components of double and sometimes also triple the fundamental frequency with a fixed phase synchronization. These sensorimotor rhythms (SMRs) are attenuated in the corresponding cortical area when a motor task (e.g. movement or motor imagery) takes place. As this effect is due to loss of synchrony in the neural populations, it is termed event-related desynchronization (ERD), see [14]. The increase of oscillatory EEG (i.e., the reestablishment of neuronal synchrony) is called event-related synchronization (ERS). The ERD in the motor and/or sensory cortex can be observed even when a subject is only thinking of a movement or imagining a sensation in the specific limb. This phenomenon makes the ERD/ERS feature attractive for BCIs.

For ‘decoding’ of different motor intentions from brain activity, the essential task is to distinguish different spatial localization of SMR modulations. Due to the topographical arrangement in the motor and somatosensory cortex, these locations are related to corresponding parts of the body. For example, left hand and right hand have corresponding areas in the contralateral, i.e., right and left motor cortex, respectively. Thus, spatial filters are an essential step for a meaningful feature extraction for the classification of motor intentions, and far from being a black-box method, learned spatial filters can be visualized appropriately and checked with neurophysiological knowledge.

The common spatial pattern algorithm is successful in calculating spatial filters for detecting modulations of the SMR or other ERD/ERS effects. Given two distributions in a high-dimensional space, the CSP algorithm finds directions (i.e., spatial filters) that maximize variance for one class and simultaneously minimize variance for the other class. Since band-power can be calculated as the variance of band-pass filtered signals, this criterion corresponds to ERD/ERS effects.

Technically CSP analysis works as follows. Let Σ_+ and Σ_- be estimates of the covariance matrices of the band-pass filtered EEG signals under the two conditions (in this paper, motor imagery of two different limbs). These two matrices are simultaneously diagonalized such that the eigenvalues of Σ_+ and Σ_- sum to 1. Practically this can be done by calculating the generalized eigenvectors W:

$$\Sigma_+ W = (\Sigma_+ + \Sigma_-) W D. \tag{1}$$

Here, the diagonal matrix D contains the (generalized) eigenvalues of Σ_+ (defined such that they are between 0 and 1) and the column vectors of W are the filters w's for computing the CSP features. By this procedure a full decomposition of the sensor space is determined. Best contrast is provided by those filters with high eigenvalues (large variance for condition 1 and small variance for condition 2) and by filters with low eigenvalues (vice versa). Therefore, the common practice in a classification setting is to use several eigenvectors from both ends of the eigenvalue spectrum as features for classification. The solution for the eigenvector with the largest eigenvalue can also be obtained by maximizing the Rayleigh quotient:

$$\underset{w \in \mathbb{R}^C}{\text{maximize}} \qquad \frac{w^\top \Sigma_+ w}{w^\top (\Sigma_+ + \Sigma_-) w}. \tag{2}$$

This correspondence is often useful for algorithmic considerations. CSP filters can be visualized as scalp maps and chosen according to physiological plausibility. For more details, see the CSP tutorial [1].

3 Robust Spatial Filters Based on Maxmin Framework

The class covariance matrices Σ_+ and Σ_- used in CSP can vary substantially because of non task-related fluctuations and/or non-stationarity of the EEG signals. In BCI applications, it is important to make the features robust against various kinds of fluctuations, e.g. caused by change of task involvement or by changes in the subject's brain processes. Under such situations, the maxmin approach which Kim et al. [12] successfully applied to FDA could be one of the promising directions to construct robust CSP filters. The key idea is that, instead of just two single matrices, we consider convex sets $\mathcal{S}_+$ and $\mathcal{S}_-$ for the class covariance matrices Σ_+ and Σ_-, respectively. These sets specify the tolerant regions of fluctuations around the class covariance matrices. For simplicity, we assume that the sets $\mathcal{S}_+$ and $\mathcal{S}_-$ are independent of each other.

Based on the maxmin framework, robust CSP can be constructed by maximizing the worst case (minimum) Rayleigh quotient within all possible covariance matrices in the tolerant regions. That is, the optimization problems for our maxmin CSP can be expressed as

$$\max_{w \neq 0} \min_{\Sigma_+ \in \mathcal{S}_+, \Sigma_- \in \mathcal{S}_-} \frac{w^\top \Sigma_+ w}{w^\top \left(\Sigma_+ + \Sigma_-\right) w} \tag{3}$$

$$\max_{w \neq 0} \min_{\Sigma_+ \in \mathcal{S}_+, \Sigma_- \in \mathcal{S}_-} \frac{w^\top \Sigma_- w}{w^\top \left(\Sigma_+ + \Sigma_-\right) w} \tag{4}$$

For a moment, we will consider only the first optimization problem (3), because the other (4) can be handled in the same way.

3.1 Derivation of the Maxmin Filters

We will consider special cases where the subspaces $\mathcal{S}_+$ and $\mathcal{S}_-$ can be defined by balls in the space of $C \times C$ positive definite matrices $\mathrm{Pd}(C)$ centered at $\overline{\Sigma}_+$ and $\overline{\Sigma}_-$

$$\begin{aligned} \mathcal{S}_+ &= \left\{ \Sigma_+ \,\middle|\, \Sigma_+ \succeq 0,\ \|\Sigma_+ - \overline{\Sigma}_+\|_{P_+} \leq \delta_+ \right\}, \\ \mathcal{S}_- &= \left\{ \Sigma_- \,\middle|\, \Sigma_- \succeq 0,\ \|\Sigma_- - \overline{\Sigma}_-\|_{P_-} \leq \delta_- \right\}, \end{aligned} \tag{5}$$

where $\|X\|_P^2 = \mathrm{Tr}\left(P^{-1}XP^{-1}X\right)$ is the norm of a symmetric matrix X and the positive definite matrix P specifies the metric of $\mathrm{Pd}(C)$, or in other words, shape of the balls.

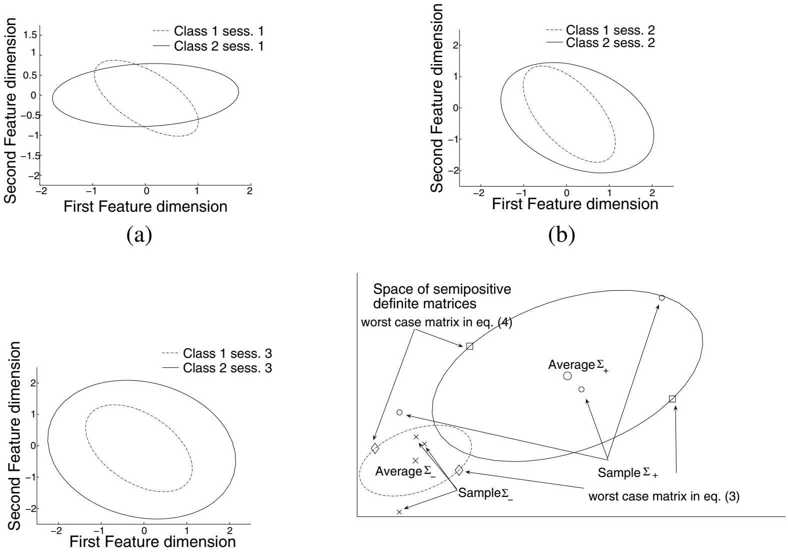
Lemma 1. *For the sets $\mathcal{S}_+$ and $\mathcal{S}_-$ defined in Eq. (5), the worst case Rayleigh quotient becomes*

$$\frac{w^\top \left(\overline{\Sigma}_+ - \delta_+ P_+\right) w}{w^\top \left(\overline{\Sigma}_+ + \overline{\Sigma}_- - \delta_+ P_+ + \delta_- P_-\right) w}, \qquad \forall w, \tag{6}$$

if $\overline{\Sigma}_+ - \delta_+ P_+ \succeq 0$.

3.2 Choice of the Parameters of the Tolerance Sets

The remaining problem is how to determine the sets $\mathcal{S}_+$ and $\mathcal{S}_-$ of the class covariance matrices. We used the average class covariance matrices for the centers $\overline{\Sigma}_+$ and $\overline{\Sigma}_-$. There are several choices of the matrix P_+ and P_- which specify the shape of the tolerant balls. The standard norms of such forms are 'Frobenius' ($P_+ = P_- = I$) or $\overline{\Sigma}_+$ and $\overline{\Sigma}_-$ (the centers). However, in the latter case the filters coincide with those of CSP. Although the identity matrix ignores plausible directions of fluctuation in EEG signals, the maxmin CSP with this setting still improved the performance in the day-to-day transfer experiment. We conjecture that this is analogous to the fact that Bayesian regularization helps even with non-informative priors. If we have extra (prior) information about possible fluctuations as is the case with the real world BCI data in [15], the covariance of the distortions can be used for the matrices P_+ and P_-. This approach was called invariant CSP (iCSP). Future work should be done to analyze the scatter of



(a) (b) (c) (d)

Fig. 1. Figs. (a), (b) and (c) represent Σ_+ and Σ_- at different time points. Mean of the features is 0, as it is bandpass filtered data. Fig. (d) represents the previous matrices as points in the space of positive definite matrices. The ellipsoids in Fig. (d) are the tolerant sets S_+ and S_- centered at the average matrices $\overline{\Sigma}_+$ and $\overline{\Sigma}_-$, respectively. From both ellipsoids, a pair of the worst case covariances is obtained for each optimization problem (3) or (4).

the short-term covariance matrices in detail and to find a reasonable choice of these matrices. The sizes of the balls δ_+ and δ_- are assumed to be equal and are selected by cross-validation on the training set.

Fig. 1 is an illustrative explanation of our method. Although we develop the theory only for the first eigenvectors, in the experiments we will use a few eigenvectors each. Further work should be done to extend the lemma 1 for multiple eigenvectors.

4 Application to EEG Data from Brain-Computer Interfacing

In this paper we evaluate the proposed algorithm on off-line data in which substantial fluctuations are expected. First we show that maxmin CSP can work under the same settings as iCSP, that is robustifying the filters when a known and measurable distortion affects the data. Later we also test the algorithm to robustify the filters against session-to-session (day-to-day) variability which may be caused by different mental conditions, materials (cap and electrodes) and different preparation of the measurement devices. In the last example, we test whether classifiers trained with recordings from multiple paradigms can be transferred to data in yet another paradigm. The non-stationarity is induced by having different background conditions (presentation of the cues) for the same primary task (motor imagery). In this paper we use ‘calibration measurements’ in which no feedback is provided to avoid bias toward any method. The trials of these 'calibration data' have fixed length of 3.5 seconds in which an imagery motor task is performed (right or left hand or foot movement). The cue is given as auditory ('imag_audi') or visual stimuli. The visual stimuli can be a letter L, R, F ('imag_lett'), a fixed arrow pointing to left, right or down ('imag_arrow') or a randomly moving small rhomboid ('imag_move') with either its left, right or bottom corner filled to indicate the task. The movement of the object in 'imag_move' is independent from the indicated targets with the aim to induce target-uncorrelated eye movements. For the evaluation, we only considered binary classification. From the three motor imaginary tasks, the best pair was selected based on separability for each subject. The data was recorded using 48 Ag/AgCl electrodes in an extended 10-20 system sampled at 1000 Hz with a band-pass from 0.05 to 200 Hz.

Linear discriminant analysis (LDA) is used as the classifier and the performance is measured by error rate.

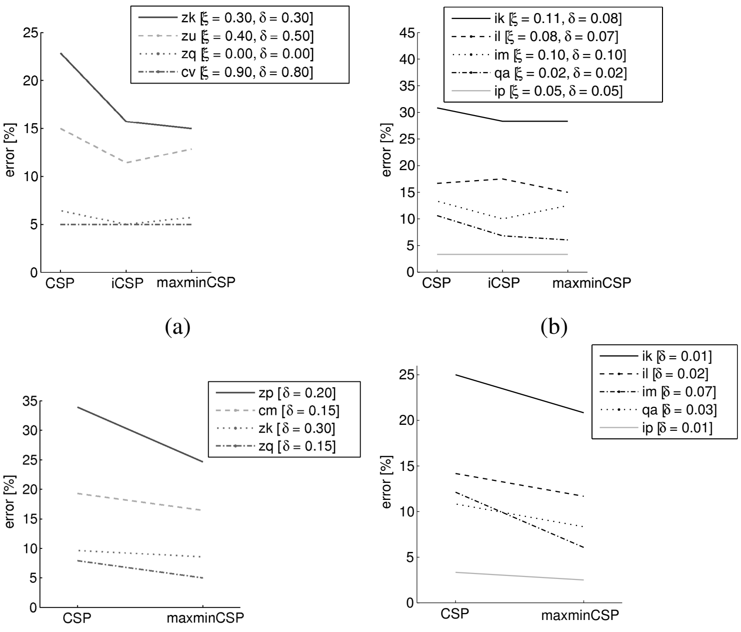
Maxmin CSP Working as iCSP. First we want to show that maxmin CSP can work under the same conditions of iCSP obtaining similar results. Therefore we realize two experiments. The first one is a replica of that presented in [15]. As in [15], we investigate whether it is possible to robustify CSP against different demands in visual processing, which cause substantial differences in the background brain activity [15]. We train CSP using 'imag_move'. Additional data ('sham_feedback') with task uncorrelated eye movements and visual activity is recorded to create a disturbance matrix (for more information of the paradigm please refer to [15]). Fig. 2(a) depicts the results of 4 subjects in which we see that iCSP and maxmin CSP exhibit very similar performance.

The second experiment is done with 5 subjects to test maxmin CSPs invariance properties in another setting. We use 'imag_move' data for training

CSP and some data recorded while subjects were told to keep their eyes closed, is used to characterize the expected disturbance (matrices $P_+ = P_-$). Maxmin CSP is then tested in 'imag_lett', in which the visual activity is supposed to be less intense (more similar to an 'eyes-closed' condition). Results are depicted in Fig. 2(b). Again iCSP and maxmin CSP perform very similarly, however, better than the original CSP.

Session-to-session Transfer. For this test we use data from four subjects for whom we had recorded several sessions in different days (even with more than 12 months difference). For subject *zq* we use 6 available datasets, 5 datasets for *cm* and *zp* and finally 4 for subject *zk*. All files except one are used for training the maxmin filters. Matrices P_+ and P_- are the identity matrix. The last file is used to test the performance of each subject. All datasets correspond to 'calibration measurements' of type 'imag_lett'. The performance of the subjects is described using error rate and the trials are classified using LDA. Fig. 2(c) shows this error rate when using CSP or maxmin CSP for preprocessing the data. Again the new maxmin CSP method outperforms CSP in all cases.

Paradigm-to-paradigm Transfer. To test whether it is possible to use 'calibration measurements' recorded using different paradigms, we perform an experiment in which 'imag_move' and 'imag_audi' data are used to train maxmin CSP, whereas the performance test is done in 'imag_arrow', in which the visual task is less demanding than



(a) (b)

(c) (d)

Fig. 2. Test errors of our maxmin CSP compared with CSP and iCSP. The legend provides subject codes (two letter codes) and the selected parameters in square brackets. (a) maxmin CSP working as iCSP (with sham_feedback), (b) maxmin CSP working as iCSP (with 'eyes closed' measurement), (c) session-to-session transfer, only possible with maxmin CSP, (d) paradigm transfer, only possible with maxmin CSP.

'imag_move'. The number of trials used from 'imag_audi' is found subject-specifically by cross validation on the training set. Note that no additional and specific recording is necessary to estimate maxmin CSP. In this setting, we use data from five subjects and their performance using CSP and maxmin CSP is depicted in Fig. 2(d). Again, matrices P_+ and P_- are the identity matrix. We see that the performance of all five subjects can be improved by maxmin CSP.

All parameters of the experiments are fixed on the training set using 10 fold cross-validation before testing in unseen data: δ for maxmin CSP, ξ for iCSP and the number of trials from 'imag_audi' in the paradigm-to-paradigm transfer experiment.

5 Conclusions

BCI data is contaminated by a variety of noise sources, artifacts, non-stationarities and outliers that make it indispensable to strive for more robust learning methods. In this paper we proposed a novel classification algorithm that is inspired by the work of Kim et al. [12].

In particular, we analyze the worst case performance among possible class covariance matrices and optimize the respective CSP-like filters based on such a criterion. When we take hyperspheres in the matrix space for the sets of the covariances, the algorithm can be elegantly reduced to a generalized eigenvalue problem similar to the original CSP, but with modified covariance matrices. Extensive simulations show that our new CSP framework is indeed more robust as it allows to transfer BCI classifier knowledge from session to session. This is again a further step towards a BCI system that is more stable with respect to non-stationarities and non-task related fluctuations.

In future studies we will continue working towards more robust BCIs and also evaluate the present approach in feedback experiments. Since the maxmin approach is an extreme case, we will furthermore pursue Bayesian approaches which may bridge the gap between this extreme method and classical CSP.

Acknowledgments. The studies were partly supported by the *Bundesministerium fuer Bildung und Forschung* (BMBF), FKZ 01IBE01A/B, by the German Science Foundation (DFG, contract MU 987/3-1), by the EU's IST Programme under the PASCAL2 Network of Excellence, IST-216996, by the EU's Marie Curie grant number MEIF-CT-040666 and by the EU's ICT Programm under TOBI, ICT-224631. This publication only reflects the authors' views.

References

1. Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., Müller, K.R.: Optimizing spatial filters for robust EEG single-trial analysis. IEEE Signal Proc. Magazine 25(1), 41–56 (2008)
2. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clin. Neurophysiol. 113, 767–791 (2002)
3. Millán, J.: Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge (2002)
4. Dornhege, G., del, R., Millán, J., Hinterberger, T., McFarland, D., Müller, K.R. (eds.): Toward Brain-Computer Interfacing. MIT Press, Cambridge (2007)
5. Blankertz, B., Müller, K.R., Krusienski, D., Schalk, G., Wolpaw, J.R., Schlögl, A., Pfurtscheller, G., del, R., Millán, J., Schröder, M., Birbaumer, N.: The BCI competition III: Validating alternative approachs to actual BCI problems. IEEE Trans. Neural Sys. Rehab. Eng. 14(2), 153–159 (2006)
6. Blankertz, B., Dornhege, G., Krauledat, M., Müller, K.R., Curio, G.: The non-invasive Berlin Brain-Computer Interface: Fast acquisition of effective performance in untrained subjects. NeuroImage 37(2), 539–550 (2007)
7. Müller, K.R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., Blankertz, B.: Machine learning for real-time single-trial EEG-analysis: From brain-computer interfacing to mental state monitoring. J. Neurosci. Methods 167(1), 82–90 (2008)
8. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Neural Networks 12(2), 181–201 (2001)
9. Blankertz, B., Dornhege, G., Lemm, S., Krauledat, M., Curio, G., Müller, K.R.: The Berlin Brain-Computer Interface: Machine learning based detection of user specific brain states. J. Universal Computer Sci. 12(6), 581–607 (2006)
10. Elbert, T., Rockstroh, B., Lutzenberger, W., Birbaumer, N.: Biofeedback of slow cortical potentials. I. Electroencephalogr. Clin. Neurophysiol. 48, 293–301 (1980)

11. Nijholt, A., Tan, D., Pfurtscheller, G., Brunner, C., Millán, J., Allison, B., Grainmann, B., Popescu, F., Blankertz, B., Müller, K.R.: Brain-computer interfacing for intelligent systems. IEEE Intelligent Systems 23(3), 72–79 (2008)
12. Kim, S.J., Magnani, A., Boyd, S.: Robust fisher discriminant analysis. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems, vol. 18, pp. 659–666. MIT Press, Cambridge (2006)
13. Koles, Z.J.: The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. Electroencephalogr. Clin. Neurophysiol. 79(6), 440–447 (1991)
14. Pfurtscheller, G., Lopes da Silva, F.: Event-related EEG/MEG synchronization and desynchronization: basic principles. Clin. Neurophysiol. 110(11), 1842–1857 (1999)
15. Blankertz, B., Kawanabe, M., Tomioka, R., Hohlefeld, F., Nikulin, V., Müller, K.R.: Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing. In: Advances in Neural Information Processing Systems, vol. 20, pp. 113–120. MIT Press, Cambridge (2008)

Multiple AM Modulated Visual Stimuli in Brain-Computer Interface

M.-A. Lopez[1], H. Pomares[2], A. Prieto[2], and F. Pelayo[2]

[1] Department of Signal Theory, Telematics and Communications, University of Granada
`malg@ugr.es`
[2] Department of Computer Architecture and Technology, University of Granada
`hpomares@atc.ugr.es, aprieto@atc.ugr.es, fpelayo@ugr.es`

Abstract. Steady-state visual evoked potential (SSVEP) based Brain-computer interfaces (BCIs) use the spectral power at the flickering frequencies of the stimuli as the feature for classification in an Attend/Ignore multi-class paradigm. The performance of a BCI based on this principle increases with the number of stimuli. However the number of the corresponding flickering frequencies is limited due to several factors. Besides the frequency response of SSVEPs is not uniform for the full range of frequencies and varies among individuals, being difficult to establish accurate decision boundaries. In this paper we propose a new technique to overcome this limitation based on the AM modulation of the flickering stimuli that reuses the same modulating frequencies with different phase.

Keywords: Brain-computer interfaces, steady-state visual evoked potentials, selective attention, AM modulation, LVQ.

1 Introduction

The SSVEP is a periodic response elicited by the repetitive presentation of a visual stimulus, at a rate of 6 Hz or more, whose power extends over an extremely narrow bandwidth as the periodicity of the response matches that of the stimulus [1]. The SSVEP amplitude is substantially increased when attention is focused upon the location of the flickering stimulus and it is more pronounced in recordings over the posterior scalp contralateral to the visual field of stimulation over the occipital area [2]. BCIs based on this principle use a set of flickering frequencies that are presented to the subject in an Attend/Ignore paradigm [3], [4]. The data of the trial is submitted to Fourier analysis and the amplitude of the spectral component at the flickering frequency and first harmonic are, generally used as the criteria for classification [5], [6].

A common problem in the design of SSVEP based BCIs is the selection of the flickering frequencies, as it is a key point in the performance of the system and is limited by several factors. In one hand an efficient stimulus, in terms of amplitude of the response, is limited to the band [6..28] Hz. This range of frequencies includes de alpha band that should be avoided if possible. In the other hand a discrete number of frequencies should be used to avoid spectral leaking. This restriction applies to frequencies with an exact number of cycles in the duration of a trial as well as

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 683–689, 2009.
© Springer-Verlag Berlin Heidelberg 2009

submultiples of the CRT refresh rate. In this paper we propose a new technique that avoids the limitations in the number of suitable frequencies by the AM modulation of the flickering stimulus by pure tones with different phase shifts. In this way different stimuli using the exactly the same modulating frequency can be classified by the information conveyed by the phase.

2 Multiple Modulated SSVEPs

The multiple modulated SSVEPs is a technique based on the multiple AM modulation of the flickering frequency that elicits the SSVP by one or more modulating pure tones with the same or different frequencies and phases. If two AM modulated visual stimuli according to expressions $s_0 = A[1 + a_m \cos(2\pi f_1 t)]\cos(2\pi f_0 t)$ and $s_1 = A[1 + a_m \cos(2\pi f_1 t + \psi)]\cos(2\pi f_0 t)$ where A is the amplitude of the signal, a_m the factor of modulation, f_0 the frequency of the flickering stimulus, f_1 the frequency of the modulating pure tone and ψ an arbitrary phase, are shown to the subject and he is instructed to attend one while ignore the other, it is expected to measure the same arbitrary phase ψ in f_1, based in the physiological principle that selective attention does not noticeably affect the latencies of SSVEPs [7]. This scheme can be repeated for m stimuli $s_0 \ldots s_{m-1}$, reusing the same flickering and modulating frequency with m equal spaced phases $\psi_k = 2\pi k / m$, with the only limitation of the requested SNR for a certain accuracy in classification.

Taking into account that the human visual system can be modelled as a linear, minimum phase shift plus a fixed delay [8] and the cognitive process of selective attention does not affect significantly the delay of the SSVEP, it can be concluded that the phase shift of two input visual stimuli flickering at the same frequency keeps constant from the retina up to the visual cortex. This conclusion is consistent with studios about the use of the phase as an independent feature for classification in SSVEP based BCI [9], [10].

One advantage of this technique is that the number of suitable frequencies of stimulation can be increased. Submultiples of vertical scan rate of the CRT should be used to avoid spectral leaking. The use of a display with a refresh rate of 100 Hz only offers 13 suitable submultiples frequencies in the range [6..28] Hz, that is [25.00, 20.00, 16.67, 14.29, 12.5, 11.11, 10.00, 9.09, 8.33, 7.69, 7.14, 6.67, 6.25] Hz. This number is not enough for the design of a high performance BCI, as it increases with the $\log_2$ of the number of stimuli. For instant in [11] the performance of the system achieved 1.13 bps of information transmission rate (ITR) using 48 stimuli. In this study the stimuli were generated by means of LEDs and associated circuitry, at the cost of a paradigm without visual images. In other study frequencies were reused allocating them by rows and columns in a matrix of stimuli [12].

The SSVEP presents a frequency response with different amplitudes for different flickering frequencies [13]. Hence, for an optimal performance the system must take this into consideration, with the need of a previous session to estimate it. The use of a small number of frequencies can overcome this problem. Another advantage is that reusing the same frequencies, all the stimuli are affected uniformly by the same noise or in-band artefacts.

3 Methodology

In order to check the convenience of the technique depicted in the previous section an experiment was proposed.

3.1 Recordings and Stimulation

The system used for recording was the Geodesic EEG System 200, by Electrical Geodesic. Data in each electrode was amplified by 1000, collected at a sampling rate of 250 Hz, filtered with a low-pass filter of 100 Hz bandwidth and digitized using 16 bits per sample. According to the ISCEV standard, only one channel is used at position Oz with reference at Fp [14].

An envelope detector was used as AM demodulator. Butterworth filters were used as the phase varies linearly with the frequency causing small distortion in the signal. The visual stimuli were presented using a CRT with vertical refresh rate of 75.1 Hz measured during the experiment. It was situated 75 cm away from the subject. The screen was broken down into four sectors with one stimulus each. The modulating frequencies and phases of each sector are depicted in Table 1.

Table 1. Modulating frecuencies and phases

Sector	Freq(Hz)-Phase(º)	Sector	Freq(Hz)-Phase(º)
1	1 - 0	2	1 - 90
Sector	Freq(Hz)-Phase(º)	Sector	Freq(Hz)-Phase(º)
3	1.5 - 180	4	1.5 - 270

3.2 Experimental Design

A group of three individuals, average age of 32, with university studies and normal vision participated in the experiment. They remained comfortably sat down in a quiet testing room, isolated from noise and external disruptions. They were encouraged to relax in order to reduce the background noise level when the EEG was being recorded.

The experiment was performed in 1 session with 4 sets of 10 trials. In each set of trials they were told to focus attention to the stimuli of one sector, ignoring the others, gazing at the corresponding sector. Each trial lasted for 6 seconds, and there were 2 to 4 seconds rest inter-trial. As selective attention is an inherent feature of human beings, no previous training was needed.

3.3 Data Analysis

The system classified in 4 classes, 1 per sector, using information of frequencies and phase of the demodulated signal. A simple LVQ network was used to classify offline the input vectors with components the amplitudes and phases of the modulating tones extracted from the EEG records using Fourier analysis. As the number of available training vectors was only 40, cross-validation was used partitioning the data in 2 subsets of 39 vs. 1 vector, used for training and evaluation respectively. For each configuration the evaluation was executed 100 times and average and standard deviation were calculated. The results shown in the next section has been averaged across the group of subjects.

4 Results

The majority of the 40 trials presented higher amplitudes of the SSVEPs corresponding to the attended stimuli than those ignored. That is consistent with the effect of selective attention on attended stimuli. Besides, for a certain frequency, the averaged phase shift between stimuli matched of that stimulation, that is approximately 90º. This is coincident with the original phase shift between the stimuli and thus consistent with the expected behaviour.

The Fig. 1 shows the ratio of amplitudes of spectral components at 1 Hz and 1.5 Hz ($Amplitude(1Hz) / Amplitude(1.5Hz)$) versus the number of trials. It shows ratios above the green horizontal threshold for trials 1 to 20 and below for trials 21 to 40. That is consistent with the expected behaviour as during the first 20 trials the attention was focused on sectors 1 and 2, both flickering at 1 Hz whereas during the second half the attention was focused on sectors 3 and 4, flickering at 1.5 Hz. The fact that the horizontal threshold in the figure is not set up at 1.00 ca be due to the frequency response.

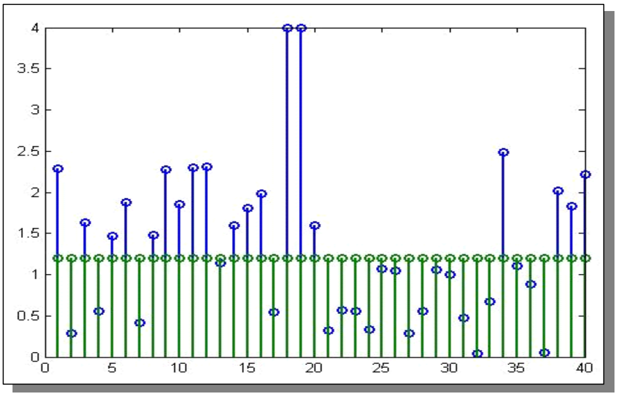
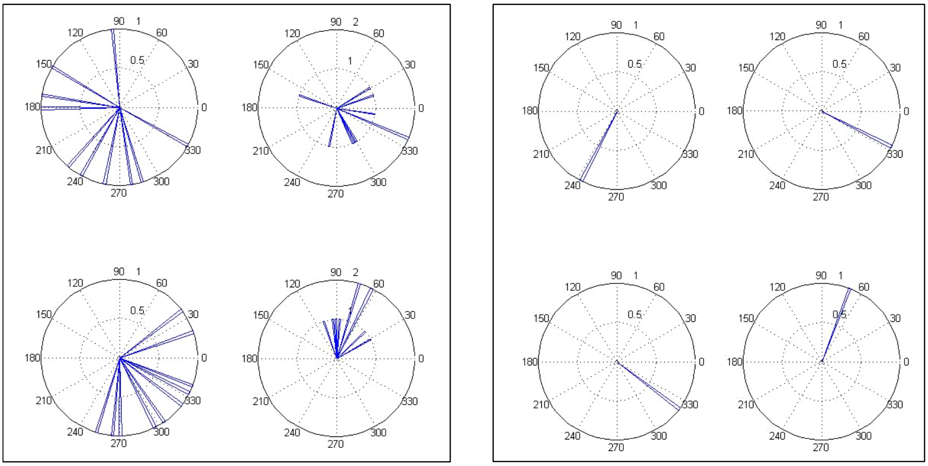


Fig. 1. Ratio of amplitudes of spectral components at 1 Hz and 1.5 Hz versus the 40 trials

The Fig. 2.a shows one histogram per sector with the phases of the modulating tones of the 40 trials. On the right, the Fig. 2.b shows the phases of the grand average. The calculation of the phase is a not linear operation with more accurate results in the presence of less noise. That justifies the differences between columns 4 and 6 of Table 2.



(a) *(b)*

Fig. 2. (a) Histograms per sector of the absolute phases of the spectral component of SSVEPs for the 40 trial. (b) Absolute phases per sector of the spectral component of SSVEPs of the grand average.

The differences of phases between sectors 1-2 and 3-4 are -96º and -117º respectively (see Table 2, column 4), that reasonably match the expected -90º initial phase shift (see Table 1). That is consistent with the hypothesis supported in this paper that the human visual system behaves like a linear, minimum phase shift plus a fixed delay [8]. Columns 5 and 6 of Table 2 show the absolute phases per sector of the grand average. In this case we obtain a better approximation to the theoretical -90º phase differences (-91º and -105º for sectors 1-2 and 3-4 respectively). The results obtained in columns 5 and 6 are more accurate than those from columns 3 and 4 as the phase was estimated in the presence of less noise.

Table 2. Phases measured averaging 1 and 10 trials for the 4 stimuli

Sector	Stimuli phaseshift (º)	Averaged phase (º)	Difference (º)	Phase of grand average(º)	Difference (º)
1	0	-130	-96	-117	-91
2	90	-34		-26	
3	180	-45	-117	-37	-105
4	270	72		68	

The Table 3 shows the results in classification of the LVQ network. The LVQ network classified among 4 classes corresponding to the 4 sectors with a successful classification rate of 69.80%.

Table 3. Classification with a LVQ network

LVQ Network			Successfull classfication (%)			
Neurons	Classes	Simulations	av.	min.	max.	std.
8	4	100	69.80	65.00	75.00	2.61

The equation $ITR(bps) = \frac{\left[\log_2(N)\right] + p_a \log_2(p_a) + (1-p_a)\log_2\left[(1-p_a)/(N-1)\right]}{T}$ estimate the information transmission rate in bits per second, with N the number of classes, T de length of a trial and p_a the accuracy [15]. Applying this equation, in our experiment it is achieved and ITR of only 6 bps. However we must keep in mind that the purpose of this experiment was not to achieve a high ITR, but to verify the reuse of frequencies by means of phase shifts.

5 Conclusions

In this paper we propose the AM modulation of visual stimuli to overcome the limitation in the number of suitable frequencies that elicit SSVEP in an efficient way. It has been shown that the original differences of phases among visual AM-modulated stimuli are kept across the human visual system and it can be recovered. If the visual stimuli that elicits the SSVPS are previously multiple AM modulated by modulating pure tones with well-know phase shifts, these phases can be easily recovered and introduced in the classification system as a new independent feature. In this way using the information of the phase, the same frequency can be reused, with the only limitation of the SNR requested for a target probability of error.

Another advantage of reusing frequencies, is that the stimuli are affected uniformly by the same noise or in-band artefacts.

Acknowledgments. The authors would like to thank the Department of Experimental Psychology and Physiology of Behavior of the University of Granada for the support in the design and execution of the experiment. This papers has been sponsored with founds from the R+D Excellence Project code TIC 3310 of regional authorities in Andalucia.

References

1. Kelly, S.P., Lalor, E.C., Finucane, C., McDarby, G., Reilly, R.B.: Visual Spatial Attention Control in an Independent Brain-Computer Interface. IEEE Trans. Biomed. Eng. 52(9), 1588–1596 (2005)
2. Muller, M.M., Picton, T.W., Valdes-Sosa, P., Riera, J., Teder-Salejarvi, W.A., Hillyard, S.A.: Effects of spatial selective attention on the steady-state visual evoked potential in the 20–28 Hz range. Cognitive Brain Research 6(4), 249–261 (1998)

3. Piccini, L., Parini, S., Maggi, L., Andreoni, G.: A Wearable Home BCI system: preliminary results with SSVEP protocol. In: Proceedings of the IEEE Engineering in Medicine and Biology 27th Annual Conference Shanghai, China, pp. 5384–5387 (September 2005)
4. Trejo, L.J., Rosipal, R., Matthews, B.: Brain–Computer Interfaces for 1-D and 2-D Cursor Control: Designs Using Volitional Control of the EEG Spectrum or Steady-State Visual Evoked Potentials. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14(2), 225–233 (2006)
5. Lin, Z., Zhang, C., Wu, W., Gao, X.: Frequency Recognition Based on Canonical Correlation Analysis for SSVEP-Based BCIs. IEEE Transactions on Biomedical Engineering 54(6), 1172–1176 (2007)
6. Muller-Putz, G.R., Scherer, R., Brauneis, C., Pfurtscheller, G.: Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. Institute of Physics Publishing, Journal of Neural Engineering 2, 123–130 (2005)
7. Hillyard, S.A., Vogel, E.K., Luck, S.J.: Sensory Gain Control (Amplification) as a Mechanism of Selective Attention: Electro-Physiological and Neuroimaging Evidence. Phil. Trans. R. Soc. Lond. B 353, 1257–1270 (1998)
8. Leonard Diamond, A.: Latency of the Steady State Visual Evoked Potential. Electroencephalogray and Clinical Neurophysiology 42, 125–127 (1977)
9. Wang, Y., Gao, X., Hong, B., Jia, C., Gao, S.: Brain–Computer Interfaces Based on Visual Evoked Potentials, Feasibility of Practical System Designs. IEEE Engineering in Medicine and Biology Magazine, 64–71 (September/October 2008)
10. Hartmann, M., Kluge, T.: Phase Coherent Detection of Steady-State Evoked Potentials: Theory and Performance Analysis. In: Proceedings of the 3rd International IEEE EMBS Conference on Neural Engineering Kohala Coast, Hawaii, USA, May 2-5, pp. 179–183 (2007)
11. Gao, X., Xu, D., Cheng, M., Gao, S.: A BCI-Based Environmental Controller for the Motion-Disabled. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11(2), 137–140 (2003)
12. Cheng, M., Gao, X., Gao, S., Xu, D.: Design and Implementation of a Brain-Computer Interface With High Transfer Rates. IEEE Transactions on Biomedical Engineering 49(10), 1181–1186 (2002)
13. Pastor, M.A., Artieda, J., Arbizu, J., Valencia, M., Masdeu, J.C.: Human Cerebral Activation during Steady-State Visual-Evoked Responses. The Journal of Neuroscience 23(37), 11621–11627 (2003)
14. Brigell, M., Chair, M.B., Barber, C., Moskowitz, A., Robson, J.: Guidelines for calibration of stimulus and recording parameters used in clinical electrophysiology of vision. Calibration Standard Committee of the International Society for Clinical Electrophysiology of Vision (ISCEV), Documenta Ophthalmologica, Rep. 107, 185–193 (2003)
15. Schlogl, A., Kronegg, J., Huggins, J.E., Mason, S.G.: Evaluation Criteria for BCI Research. In: Dornhege, G., Millan, J.R., Hinterberger, T., McFarland, D.J., Muller, K.-R. (eds.) Toward Brain-Computer Interfacing, ch. 19, Massachusetts Institute of Technology (2007) ISBN 978-0-262-04244-4

A Brain-Computer Interface Based on Steady State Visual Evoked Potentials for Controlling a Robot

Robert Prueckl and Christoph Guger

g.tec, Guger Technologies OEG, Sierningstr. 14, 4521 Schiedlberg, Austria
`{prueckl,guger}@gtec.at`

Abstract. In this paper a brain computer interface (BCI) based on steady state visual evoked potentials (SSVEP) is presented. For stimulation a box equipped with LEDs (for forward, backward, left and right commands) is used that flicker with different frequencies (10, 11, 12, 13 Hz) to induce the SSVEPs. Eight channels of EEG were derived mostly over visual cortex for the experiment with 3 subjects. To calculate features and to classify the EEG data Minimum Energy and Fast Fourier Transformation with linear discriminant analysis was used. Finally the change rate (fluctuation of the classification result) and the majority weight were calculated to increase the robustness and to provide a null classification. As feedback a tiny robot was used that moved forward, backward, to the left and to the right and stopped the movement if the subject did not look at the stimulation LEDs.

1 Introduction

A brain computer interface (BCI) is a new way of communication between humans and computers. It utilizes a very uncommon, but on the other hand probably the most direct way of access to the intentions of a person. The communication towards the computer – the will of the person – which is fed into the machine gets collected at its source – the brain.

With a BCI a person ideally does not have to make use of the common output pathways of peripheral nerves and muscles, which is the main argument for a BCI-system. A BCI-system provides a completely new output pathway and this is perhaps the only way a person can express herself if he/she suffers for example on disorders like amyotrophic lateral sclerosis (ALS), brainstem stroke, brain or spinal cord injury or other diseases which impair the function of the common output pathways which are responsible for the control of muscles or impair the muscles itself [1]. In such a case one possibility is to work with the electrical brainwaves of the person. These are measured with the well-known electroencephalography (EEG), which was primarily used for clinical purposes only in the past, amplified and fed into a personal computer which is under certain circumstances and with appropriate algorithms able to process them to give the person a new kind of communication channel.

For the proposed BCI a neurological phenomenon called steady state visual evoked potential (SSVEP) is utilized. A visual evoked potential (VEP) is an electrical potential-difference, which can be derived from the scalp after a visual stimulus, for

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 690–697, 2009.
© Springer-Verlag Berlin Heidelberg 2009

example a flash-light. VEPs after stimuli with a frequency $\leq$ 3.5 Hz are called "transient" VEPs. If the stimulation frequency is > 3.5 Hz they are called "steady state" VEPs because the individual responses overlap and result a quasi-sinusoid oscillation with the same frequency as the stimulus [2]. The goal is to detect this frequency reliably with high accuracy and furthermore to detect when the frequency is not present, thus when the person does not look at the stimulus. The later one is a very challenging task in BCI systems. The paper will introduce signal processing methods that allow answering these questions.

In the following section the methods used for measuring EEG, extracting features and classification are described. In Section 3 test results of three test-subjects are presented and interpreted. Section 4 summarizes the proposed BCI-system and makes suggestions for future work.

2 Methods

2.1 Experiment

Three healthy subjects participated in the BCI experiment and performed first the training and then the testing procedure. The training/test procedure is depicted in Figure 1.

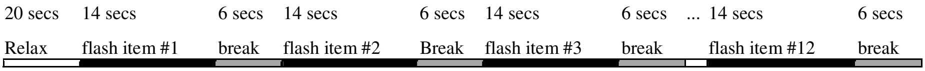


Fig. 1. The training procedure starts with a 20 second brake to have baseline EEG activity. Then each light is flashing sequentially for 14 s with 6 s breaks in between. This loop is repeated 3 times. The test procedure layout looks identical with the only exception that the lights are flashing three times each in random order.

2.2 Communication Channels

This BCI consists of three communication channels. Two of them direct from the computer to the test person and one of them directs from the person to the computer.

The first channel is the stimulation channel in which the computer produces the stimulus with certain frequencies. This is realized with a 12x12cm box (see Figure 2) equipped with four LED-groups containing three LEDs each. Each LED has a diameter of 8 mm and according to the manufacturer a light intensity of 1500 mcd. A semitransparent foil was put over the LEDs to make them look like one compact light source. Additionally four arrow LEDs were added to indicate the index the user has to look at (for training the BCI system). The LEDs are controlled by a microcontroller connected to the computer via USB. The accuracy of the produced frequencies has been validated using a digital oscilloscope. The measured maximum frequency error is < 0.025 Hz at room temperature.

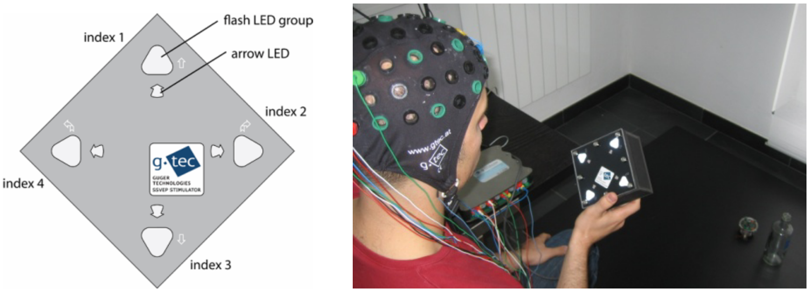


Fig. 2. *Left*: The layout of the LED stimulation box. *Right*: A test person wearing an electrode cap with mounted electrodes and holding the LED stimulation box. The electrodes are connected to the biosignal amplifier g.USBamp (g.tec medical engineering GmbH, Austria). The robot is located on the right side of the picture, besides the bottle.

The second communication channel is the EEG-data which is derived from the test person. Eight gold electrodes placed mostly over visual cortex on positions POz, PO3, PO4, PO7, PO8, O1, O2 and Oz of the international 10-20 system were used with an additional reference electrode at the right earlobe and a ground electrode at position Fpz. Abrasive gel was applied to make sure that impedances were below 5 kΩ. The electrodes were connected to an EEG-amplifier, the g.USBamp (g.tec medical engineering GmbH, Austria) which fed the signals over a USB connection into a personal computer. The internal bandpass and notch filters of the g.USBamp were used. The bandpass was set to 0.5 to 60 Hz and the notch filter was set to 50 Hz.

The last communication channel is the feedback the computer gives. The EEG data is analyzed with feature extraction and classification methods resulting in a classification output for each method. Each classifier has a discrete output in the form of a number (1, 2, 3 and 4). Finally in the last processing stage, the change rate / majority weight analysis step adds a 0 to this set of outputs. The device driver of the robot transforms these five numbers semantically to driving commands (0-stop, 1-forward, 2-right, 3-backward, 4-left) and sends them to the device, the robot, which moves and gives the feedback back to the user.

The output of the first communication channel, the stimulation is more or less unvarying. Four LED-groups are flickering with different frequencies. In case of the tests the frequencies were 10, 11, 12 and 13 Hz. These frequencies have been chosen in preceding off-line tests and showed good performance for five test subjects.

The processing of the EEG-data, thus the signals of the second communication channel is the core piece of this BCI. The programming environment to achieve the detection of the frequencies is MATLAB and Simulink.

EEG data is recorded with a sampling rate of 256 Hz. The overall process (core system in Figure 3) works on 2-second windows (512 samples) with an overlap of 448 samples and consists of three steps: pre-processing, classification and change rate/majority weight analysis. These three steps are executed four times a second to have a new output every 250 ms. The paradigm controls the stimulation (see Section 2.1 – Experiment).

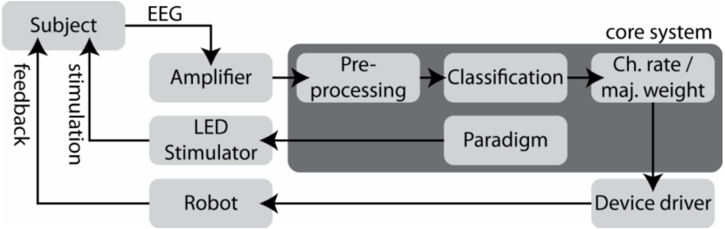


Fig. 3. Overview of the BCI-system

2.3 Pre-processing

In the pre-processing step the incoming signal windows from the g.USBamp are combined using unweighted Laplacian derivations to form some of the input signals for the classifiers [3]. Each Laplacian derivation is composed of one center signal X_c and an arbitrary number $n > 1$ of side signals $X_{s,i}, i = 1, \dots, n$ which are arranged symmetrically around the center signal. These signals are then combined to a new signal $Y_j = n \cdot X_c - (X_{s,1} + \cdots + X_{s,n})$ where j is the index of the derivation. Building the derivations in such a way performs superior to common reference or bipolar derivations in terms of artefact removal and noise cancellation.

To choose the optimal channel combinations nearly 30 different Laplacian derivations were tested on five different subjects to determine which ones deliver the best SSVEP-response. The following four derivations have performed best and were chosen for the experiment:

$Y_1 : X_c =$ Oz, $X_s = \{O1, O2, PO7, PO8\}$
$Y_2 : X_c =$ Oz, $X_s = \{O1, O2\}$
$Y_3 : X_c =$ Oz, $X_s = \{O1, O2, PO3, PO4\}$
$Y_4 : X_c =$ Oz, $X_s = \{PO7, PO8\}$

2.4 Feature Extraction / Classification

Currently classification is done with two different methods. One is the minimum energy approach (ME), which was published by O. Friman et.al. in 2007 [4] and requires no training. This algorithm is fed with raw EEG-data channels, thus no derivations, since it selects the best combination of channels by itself. First of all the EEG-data gets “cleaned” of potential SSVEP-signals. This is done by projecting artificial oscillations with stimulation frequencies (and harmonics) onto the orthogonal complement of the EEG-signals. After that operation the signals contain (theoretically) just the unwanted noise. Now a weight vector is generated, which has the property of combining the channels in a way, that the outcome has minimal energy. Now SSVEP detection is done utilizing a test statistic which calculates the ratio between the signal with an estimated SSVEP-response and the signal where no visual stimulus is present. This is done for all stimulation frequencies and all EEG-channels. The output of this classifier is the index of the frequency with the highest signal/noise ratio.

As second method a straight forward approach with the Fast Fourier Transformation (FFT) and linear discriminant analysis (LDA) using the Laplacian derivations is used. First of all the incoming data gets transformed to the frequency spectrum with a 1024-point FFT. A feature vector is extracted by taking the values on the points of the stimulation frequencies and their 1st and 2nd harmonics of all input channels. With these feature vectors a weight/bias vector must be generated for each user in a training procedure described in Section 2.1. When the training was completed successfully the classifier can then be used to classify new feature vectors to one of the stimulation frequency indices.

In the model used for the experiments described in this paper four ME classification units and four FFT+LDA classification units were used. In Table 1 the input configurations of all classifiers are listed.

Table 1. Input configurations of the eight classification units

Classifier Nr	**Input channels**	
	FFT+LDA	**ME**
1	Y_2, Y_3, Y_4	Oz, O1, O2, PO7, PO8
2	Y_1, Y_2, Y_4	Oz, O1, O2, POz
3	Y_3, Y_4	Oz, O1, O2, PO7, PO8, POz
4	Y_2, Y_4	Oz, PO7, PO8

2.5 Change Rate / Majority Weight Analysis

The last step is a procedure called change rate/majority weight analysis. By having multiple classification units configured with slightly different input data there will be in general random classification results on noise input.

This effect is used on one side to produce a zero decision when the outputs of the classifiers are changing heavily and are very different. On the other side a low change rate and a high majority weight (the number of classifications of the different algorithms which are pointing in the same direction) can be used to strengthen the robustness of the decision. Calculation is made on the last four classification results, thus on the last second. Default thresholds of 0.25 for change rate and 0.75 (1 – all outputs are pointing into the same direction) for majority weight were used. These thresholds were chosen more or less instinctively, but have performed well during the tests. However, fine tuning these thresholds is an important task for future work.

The first step of the procedure is to look at the change rate. If it is above the threshold the procedure returns a final classification result of 0 which corresponds to stop the robot. Otherwise, if it is below the threshold the next step is to look at the majority weight. If this is above the threshold the majority is taken as final result, otherwise the final output is again 0. In Figure 4 you can see the in- and outputs of the procedure.

The final classification is then sent to the device controller and finally to the robot which then provides feedback (the third communication channel) to the user by moving towards the corresponding direction (or stopping).

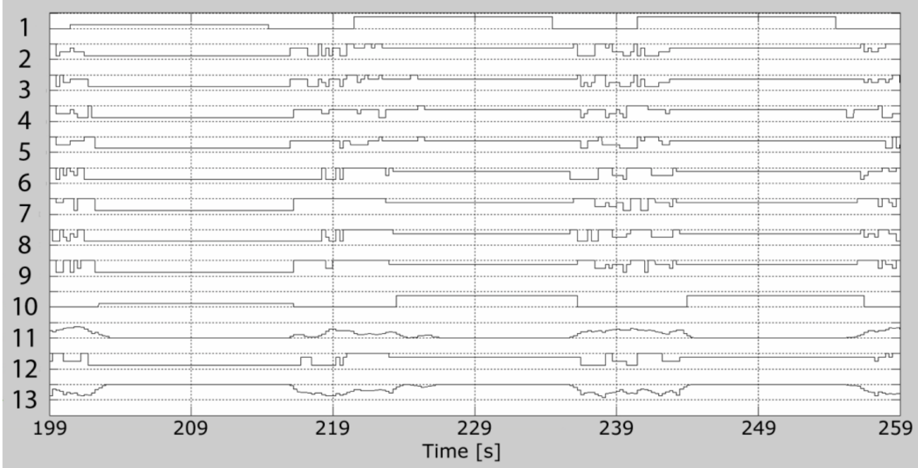


Fig. 4. Classification procedure of the BCI-system on the example of 3 decisions (1 -forward, 3- back, 3- back). Channel 1 shows the target classification of three trials (each 14 seconds in length and a 6 seconds break). Channels 2 to 5 are the outputs of the four ME classification units and channels 6 to 9 are from the FFT+LDA units. Channel 11 shows the change rate and channel 13 shows the majority weight. These two values range between 0 and 1. Channel 12 is the majority and channel 10 the final classification result which also shows classifications 1, 3, 3 with breaks in between which are correctly classified as 0. Note the delay of the final classification in comparison to the target.

3 Results

Table 2 shows the results of the testing phase. The error rate includes the whole data set which means that also the breaks were classified to test the null classification when the subject was not looking at the lights to stop the robot.

Subject 1 for example had an overall error rate of 22.7%. This error breaks down in 58.5% with no decision (robot stopped where SSVEP stimulation actually happened) and 41.5% of wrong decisions (where the chosen class was wrong, unconcerned if there had been stimulation or not).

As mentioned above there exists a delay between the target classification and the real output of the BCI. This is caused on one hand by the data collection for the 2-second analysis window of the classifiers and on the other hand by the change rate/majority analysis section which collects four classification outputs for its calculations, thus needs 1 second[1]. The sum of this delay is 3 seconds. To get an idea how the error looks if this delay is disregarded the final classification result (channel 10 in Figure 3) was shifted backwards for 768 samples. For online processing this would theoretically mean that at the time of analysis the time windows of the system would already be filled with appropriate EEG data which was generated by the brain

---

[1] Smaller delays like the physiological delay of SSVEP itself, from perception of the stimulus until the potential is measurable, or delays between sending the "stimulation on/off" signal from the computer to the microcontroller of the stimulation box, have been unattended here.

processing the right visual signals. This gives a more objective view to the classification quality itself. As you can see on the right side of Table 2 subject 1 had an overall error rate of 9.5% with a fraction of 28.3% of wrong classifications. This means only 28 wrong classifications were made during the whole experiment including the breaks (in total 1046 decisions).

Table 2. Results of SSVEP tests for 3 subjects. The error is calculated by comparing the target with the actual classification. The table shows the results without delay (target and actual classification are directly compared) and with a 3 seconds delay (the actual classification has a delay of about 3 seconds and therefore the target was shifted forward for the comparison). The overall error splits up into two subcategories of errors. No decision: no majority could be found for one class. Wrong class: the classification result was > 0 and not equal to the target classification. 'Rel' is the percentage with regard to the overall error. 'Abs' is the percentage with regard to the whole experiment.

	Without delay			Shifted by 768 samples		
Subject	Overall error [%] ME+LDA	No decision [%] Rel / Abs	Wrong class [%] Rel / Abs	Overall error [%] ME+LDA	No decision [%] Rel / Abs	Wrong class [%] Rel / Abs
S1	22.7	58.5 / 13.3	41.5 / 9.4	9.5	71.7 / 6.8	28.3 / 2.7
S2	35.7	77.9 / 27.8	22.1 / 7.9	23.5	92.7 / 21.8	7.3 / 1.7
S3	28.7	63.8 / 18.3	36.2 / 10.4	18.9	75.0 / 14.2	25.0 / 4.7
Mean	**29.0**	**66.7 / 19.3**	**33.3 / 9.7**	**17.3**	**79.8 / 13.8**	**20.2 / 3.5**

4 Conclusion

A BCI system based on SSVEPs was presented which delivers a quasi-continuous command stream and has a robust zero-classification mechanism. Three subjects participated in tests and achieved an average error rate of 29%. Of these errors 66.7% on average are zero-class errors where the robot remains stopped and executed no wrong command. Thus the average percentage of wrong commands seen for the whole experiments was 9.7%. This is a great performance for controlling the movement of a robot including the zero class.

In future test runs it is necessary to evaluate other parameter configurations (source derivations, electrode positions, analysis window lengths, feature extraction procedures, thresholds for change rate/majority analysis) to optimize the error rates and the delay. This is important for providing fast feedback to the user to give him a precise and crisp feeling of control for the robot.

Further tests will use a predefined route that must be followed with the robot to observe performance parameters such as duration, excursions, behaviour of the test person when looking between box and feedback of the robot, ... That would not only give an impression of the error rate, but also of the usability of the system.

It would also be very interesting to test the performance of the computer screen stimulator and compare it to the LED stimulator.

In some other test runs partly other electrode positions were used which lay below position Oz. Experiments showed that this yields to a further improvement.

Furthermore tests have shown that for some subjects LDA had superior performance and for other subjects ME worked better. Further experiments are needed to optimize this configuration.

References

1. Wolpaw, J.R., Birbaumer, N., McFarlanda, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clinical Neurophysiology 113, 767–791 (2002)
2. Paulus, W.: Elektroretinographie (ERG) und visuell evozierte Potenziale (VEP). In: Buchner, H., Noth, J. (eds.) Evozierte Potenziale, neurovegetative Diagnostik, Okulographie: Methodik und klinische Anwendungen, Thieme, Stuttgart - New York, pp. 57–65 (2005)
3. Lagerlund, T.D.: EEG Source Localization (Model-Dependent and Model-Independent Methods). In: Niedermeyer, E., de Silva, F.L. (eds.) Electroencephalography: Basic Principles, Clinical Applications, and Related Fields, pp. 829–844. Lippincott Williams & Wilkins, Baltimore (2004)
4. Friman, O., Volosyak, I., Graser, A.: Multiple Channel Detection of Steady-State Visual Evoked Potentials for Brain-Computer Interfaces. IEEE Transactions on Biomedical Engineering 54, 742–750 (2007)

Asynchronous Brain-Computer Interface to Navigate in Virtual Environments Using One Motor Imagery

Francisco Velasco-Álvarez and Ricardo Ron-Angevin

Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga,
Campus Universitario de Teatinos, 29071, Málaga, Spain
`{fvelasco,rra}@dte.uma.es`

Abstract. A Brain-Computer Interface (BCI) application focused on the control of a wheelchair must consider the danger which a wrong command would involve in a real situation. Virtual reality is a suitable tool to provide subjects with the opportunity to train and test the application before using it under real conditions. Recent studies aimed at such control let the subject decide the timing of the interaction, those are the so-called asynchronous BCI. One way to reduce the probability of misclassification is to achieve control with only two different mental tasks. The system presented in this paper combines the mentioned advantages in a paradigm that enables the control of a virtual wheelchair with three commands: move forward, turn left and turn right. The results obtained over three subjects support the viability of the proposed system.

Keywords: Brain-computer interface (BCI), virtual environment (VE), navigation, asynchronous, motor imagery (MI), mental tasks.

1 Introduction

A Brain-Computer Interface (BCI) is a system that enables a communication that is not based on muscular movements but during brain activity. This activity can be measured through electroencephalographic (EEG) signals. Several EEG signals can be detected, resulting in different types of BCI. Sensorimotor rhythm-based BCIs (SMR-BCI) are based on the changes of μ and β rhythms [1]. These rhythms correspond to specific features of the EEG signals characterized by their frequencies that can be modified by voluntary thoughts. When a person performs a movement, it causes a synchronization/desynchronization in this activity (event-related synchronization /desynchronization, ERS/ERD) which involves a μ rhythm amplitude change. However, the outstanding property of these signals is their behaviour when someone merely considers movement (motor imagery, MI), because this causes a similar amplitude change [2]. This relevant characteristic is what makes SMR suitable to be used as input for a BCI.

The main aim of the BCI research is to provide severely disabled people with an alternative non-muscular communication channel. Researchers have worked to develop many applications to improve the quality of life of these patients, for whom a BCI represents a viable channel to interact within their environment. Among these

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 698–705, 2009.
© Springer-Verlag Berlin Heidelberg 2009

applications, there are some where subjects take control of a real wheelchair in an experimental situation [3]. However, in most cases subjects are instructed to drive a simulated wheelchair in a virtual environment (VE). Before people can use a wheelchair in a real situation, it is necessary to guarantee that they have enough control to avoid dangerous scenarios. Virtual reality (VR) is a suitable tool to provide subjects with the opportunity to train and test the application. In order to control a wheelchair, several paradigms can be used. We will mention some of the works related to this field, in the following paragraph.

Some studies describe a system in which the wheelchair moves in only one direction (move forward) [4], [5]. Because of this restricted movement, only one command (and therefore, one mental task) is needed. Other systems let the subjects choose among more commands. In [6], a simulated robot performs two actions ('turn left then move forward', or 'turn right then move forward') in response to left or right hand MI. A more versatile application can be found in [7] with three possible commands (three MI tasks): turn left, turn right and move forward. A higher number of commands makes it easier to control the wheelchair, since the subject has more choices to move freely (by means of an information transfer rate increase). The mentioned works bind the number of commands to the number of mental tasks. Nevertheless, it has been reported in several studies ([8], [9]) that the best classification accuracy is achieved when only two classes are discriminated. For this kind of applications, the aim of which is to control a wheelchair, a classification error (a wrong command) can cause dangerous situations, so it is crucial to guarantee a minimum error rate to keep the users safe. Many works aimed at navigation in VE have the limitation of being synchronous: the control is system-paced, that is, the subject can only interact with the environment at certain moments given by the system. Recent research tends to let the subjects decide when to control the system [3]-[7], which is a more natural form of interaction. These systems, called self-paced or asynchronous, distinguish between two states: (i) a non-control (NC) state in which subjects can be involved in a mental activity different to controlling the BCI, and (ii) an intentional control (IC) state, in which subjects can control the system by specific mental tasks [5], [7]. Subjects switch voluntarily the state they are in.

The purpose of this study is to present an asynchronous paradigm to improve the synchronous system proposed in [10]. The new system waits in a NC state, until subjects voluntarily enter the IC state. In such state, subjects are offered a graphical interface with three navigation commands (move forward, turn right and turn left) they can choose carrying out only one MI. After a command is selected, the system performs the appropriate action and returns to the NC state.

2 Methods

2.1 Subjects and Data Acquisition

Three subjects (male, right-handed, age 24 ± 11.4), named S1, S2 and S3 participated in the study. They had all previous BCI experience.

The EEG was recorded from two bipolar channels. The active electrodes were placed 2.5 cm anterior and posterior to electrode position C3 and C4 (right and left hand sensorimotor area, respectively) according to the 10/20 international system. The ground electrode was placed at FPz position. Signals were amplified by a four channel Coulborn V75-08 amplifier and then digitazed at 128 Hz by a 12 bit resolution data acquisition card DAQCard-6025E (National Instruments).

2.2 Initial Training and Signal Processing

Although all subjects had experience in BCI, they underwent two initial training sessions for calibration purposes. This training was the same as the used in [10], and was based in the paradigm proposed by the Graz group (see [11] for details), in which a bar extended according to the mental task carried out (relaxed state or right hand MI). The training protocol consisted of two sessions: a first one without feedback and the other one providing continuous feedback. An offline processing of the first session (see [10] for details) made possible to determine the parameters for the feedback session. The same parameters were used to calibrate the system for the VE exploration sessions. This processing is based in the procedure detailed in [12], and consisted of estimating the average band power of each channel in predefined, subject specific reactive frequency (manually selected) bands at interval of 500 ms. In the feedback session, the length of the bar was computed on line every 31.25 ms as a result of a Linear Discriminant Analysis (LDA) classification. The trial paradigm and all the algorithms used in the signal processing were implemented in MATLAB

2.3 Virtual Environment Exploration Sessions

Paradigm Description. The VE created to navigate through consisted of a park where subjects could move freely (Figures 1 and 2) but needed to avoid several obstacles (trees, bushes, lamps, benches and others).

The system waited in a NC state in which it an NC interface was shown. The NC interface enabled subjects to remain in the NC state (not generating any command) until they decided to change to the IC state, where the control was achieved through the IC interface.

The NC interface consisted of a semi-transparent vertical blue bar placed in the centre of the screen. The bar length was computed every 62.5 ms as a result of the LDA classification: if the classifier determined that the mental task was right hand MI, the bar extended; otherwise (relaxed state), the bar length remained at its minimum size. When the length exceeded a subject dependant selection threshold (Figure 1) for a selection time (also chosen by each subject) the system changed to the IC state. Once the threshold was exceeded, the bar colour changed immediately to yellow, and then, progressively to red while the length was above the threshold, until the selection time had passed. If the length was temporarily lower than the threshold, the bar colour did not change unless it lasted less than the threshold for a reset time; in this case the colour turned again to blue, and the selection time was reset. The colour changes represented another kind of feedback that enabled subjects to know how much time (approximately) they still needed to succeed in their selection.

The IC interface was similar to the one presented in [10]: a circle divided into three parts, which corresponded to the possible navigation commands (move forward, turn right and turn left), with a bar placed in the centre of the circle that was continuously rotating clockwise. The subject could extend the bar carrying out the MI task to select a command when the bar was pointing at it. The way the selection worked in this interface was the same as in the NC interface, with the same selection and reset time and selection threshold. In the IC interface, another threshold was defined (stop threshold). When it was exceeded, the bar stopped its rotation in order to help the subject in the command selection. The rotation speed was fixed to 2.5 degrees every computation iteration, so it lasted 9 s to complete a turn if there was not any stop.

Once a command was selected, the subject moved in the VE as a consequence in a discrete way: it turned 90º right or left, or it moved forward a fixed distance (1 m in the VE) or until it collided with an obstacle. After the movement, the system changed again to the NC state. Subjects could choose a pause time in the changes between the two states.

To increase the degree of inmersion, the VE was projected on a large screen (2 x 1.5 m) and subjects were placed at a distance of 3 m. The VE was created with VRML 2.0, and its interaction with MATLAB was achieved using the MATLAB Virtual Reality Toolbox.

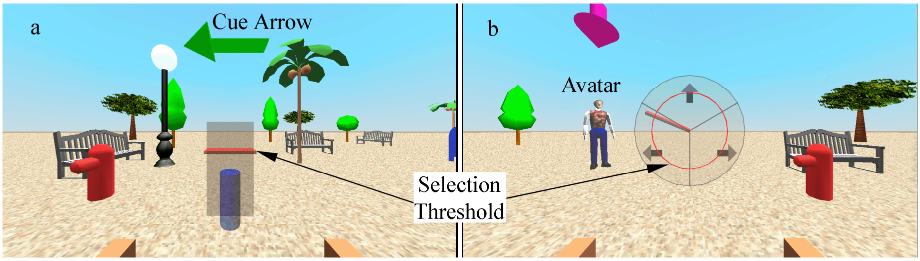


Fig. 1. NC interface (a) and IC interface (b)

Evaluation Sessions. One important issue in a BCI system is its evaluation. In an asynchronous system, in which subjects can freely select a command (whatever command, whenever they want), this evaluation is difficult to establish.

What we propose in this paper to evaluate our paradigm, is to use it firstly as a synchronous BCI, before letting subjects navigate in the VE. For this reason, after the initial training, subjects underwent an evaluation session, in which they were asked to select a series of commands after determined cues. The cue consisted of the colour change of an arrow at the top of the screen whose direction indicated which command should the subject select once he/she got the IC state (move forward, turn right or turn left). The initial colour was red, meaning that subjects should not yet change to the IC state. After a random time (2 – 5 s), the arrow colour changed to green, indicating that subjects should carry out the MI to change to the IC state.

The two main parameters measured in the NC state were true positive (TP) and false positive (FP) rates, which corresponded to an activation in the correct or incorrect moment (after the cue or before it), respectively. The nomenclature was taken from [13]. In the IC, the main measure is the performance error.

Each subject underwent a first run, which took approximately 10 minutes, so that they could become familiar with the paradigm, and the system could be calibrated. The calibration consisted of the choice of the selection, pause and reset time and the selection and stop threshold adequate to each subject. The same day they participated in two evaluation runs. In a separate day they had other three runs. Each run consisted of 21 trials (7 of each command, randomly distributed).

Free Navigation Sessions. After the system was evaluated, subjects participated in the sessions which represented the main aim of this study. They were asked to move freely through the VE to reach an objective: an avatar placed 24 m (Manhattan or grid-line distance) away from the subject's initial position. The objective of this test was to check the usefulness of the system, simulating the behaviour in a real environment. In these sessions, the only cue the system gave to the subjects was the relative position of the avatar, indicated by an arrow pointing in the direction of the objective in case it was not in the subjects' field of view. As users were not supposed to follow the same route, a command could be considered neither wrong nor correct, so the measures that were taken correspond to the number of commands and the time needed to reach the avatar.

3 Results

3.1 Evaluation Sessions

In the evaluation sessions, two points of view were taken into account. First of all, several parameters were measured for each subject, according to their performance in the NC state. In Table 1 the number of FP activations is shown for each subject and run. The parameter FP (%) was computed over 105 trials. In the two last columns, we show the mean time subjects used to change to the IC state when they did it after (Mean TP T_{NC}) and before the cue (Mean FP T_{NC}).

Table 1. Results for the NC state during the evaluation sessions

Subject	FP						FP (%)	Mean TP T_{NC} ± SD (s)	Mean FP T_{NC} ± SD (s)
	Run					All runs			
	1	2	3	4	5				
S1	0	0	0	1	0	1	0.95	6.86 ± 0.97	1.69 ± 0.00
S2	1	0	0	1	0	2	1.69	7.02 ± 1.22	1.14 ± 0.89
S3	2	0	0	2	0	4	3.81	3.41 ± 0.35	2.46 ± 0.52
All	3	0	0	4	0	7	2.22	5.78 ± 0.56	1.97 ± 0.63

These times were measured from the beginning of the trial in the FP and from the system cue in the TP. These times include the selection time chosen by each subject in the calibration session: 1.3 s, 1 s and 1.2 s for S1, S2 and S3 respectively.

Secondly, subjects' performance in the IC state is shown in Table 2. These measures do not consider if the change from NC to IC was due to a TP or a FP. They represent the subjects' capability to select the commands they were asked. We show the number of errors they made in each run and for each kind of command they were asked, as well as the total number of errors. The parameter Error (%) is computed over 35 trials for each command (105 trials for all of them). The last two columns present the time that subjects needed to select a command starting: i) at the moment the bar pointed to this one (Mean block T_{IC}) and ii:) when the IC interface was shown (Mean total T_{IC}).

Table 2. Results in the IC state in the evaluation sessions

Subject	Command	Errors						Error (%)	Mean block $T_{IC} \pm$ SD (s)	Mean total $T_{IC} \pm$ SD (s)
		Run					All runs			
		1	2	3	4	5				
S1	Forward	0	0	0	0	1	1	2.86	4.46 ± 0.77	5.91 ± 1.95
	Right	0	0	0	0	0	0	0	4.69 ± 0.8	12.04 ± 2.93
	Left	1	0	0	1	1	3	8.57	4.41 ± 0.83	13.86 ± 1.38
	All	1	0	0	1	2	4	3.80	4.52 ± 0.45	10.55 ± 1.42
S2	Forward	0	0	0	0	0	0	0	3.35 ± 0.74	11.8 ± 3.92
	Right	1	0	0	1	0	2	5.71	3.2 ± 0.6	17.7 ± 7.6
	Left	2	0	1	0	1	4	11.42	4.07 ± 0.76	18.8 ± 4.48
	All	3	0	1	1	1	6	5.71	3.58 ± 0.4	15.97 ± 3.16
S3	Forward	0	0	0	0	0	0	0	3.95 ± 0.68	6.7 ± 1.62
	Right	2	0	0	1	0	3	8.57	3.18 ± 0.48	7.63 ± 1.17
	Left	2	1	0	3	0	6	17.1	3.27 ± 0.48	10.86 ± 0.76
	All	4	1	0	4	0	9	8.57	3.5 ± 0.32	8.06 ± 0.86
All	All	8	1	1	6	3	19	6.03	3.88 ± 0.23	11.56 ± 1.24

3.2 Free Navigation Sessions

The results obtained in the navigation sessions are shown in Table 3, and the different paths subjects followed in Figure 2. The table presents the number of commands of each type that subjects selected. Mean T_{NC} and Mean total T_{IC} show the mean time they needed to get the IC state and to select a command respectively. Finally the total time used to reach the avatar is presented (T_{TOT}).

Table 3. Results in the free navigation sessions

Subject	Forward	Right	Left	Total	Mean $T_{NC} \pm$ SD (s)	Mean total T_{IC} (s)	T_{TOT} (s)
S1	27	7	6	40	4.62 ± 0.86	5.96	638
S2	24	5	4	33	4.41 ± 1.23	8.98	619
S3	25	3	2	30	3.86 ± 0.83	4.74	420

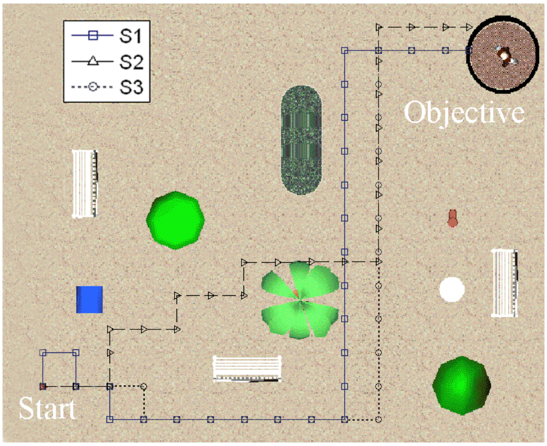


Fig. 2. Paths covered by the three subjects

4 Discussion and Conclusion

Three trained subjects participated in the study. Firstly they underwent two training sessions for calibration purposes. Afterwards they were able to remarkably control a virtual wheelchair within the first two sessions using the proposed paradigm. This paradigm combines the main advantage of a system with several classes (i.e. to have different control commands to ease the way of interaction) with the high classification accuracy which provides a system with only two mental tasks. We have presented an asynchronous BCI, in the sense that it is the subject who gives the cue to the system when he/she wants to achieve control.

Although the number of subjects is not high enough to deduce strong conclusions, the results of the evaluation sessions show that with a very low FP rate (2.22%), subjects could select the commands they wanted with an error rate close to 6%.

Another important fact is that the mean time needed to get the IC state and to select a command among three (4.33 and 6.56 s respectively, averaged over the three subjects during the navigation session) is quite low to consider the paradigm as useful. In these sessions subjects showed more motivation, and maybe for this reason, they achieved an error rate lower than 1%.

There is a remarkable difference in the time that S3 needed to reach the avatar in front of S2 and S1. It can be explained because S3 chose the shortest route (in term of number of commands) and had a lower T_{NC} and T_{IC} time.

It must be mentioned that from a total of 103 commands (for all subjects), only one was considered incorrect by subjects (the first advance of S1), and only one collision occurred (the fourth advance of S2). The average selection time in these sessions was noticeably lower than in the evaluation sessions because of the higher appearance rate of the command forward, which, due to its position in the IC interface, needed less time to be selected.

The recent work of our group is aimed at extending the study to a continuous movement, instead of the discrete advances and turns presented in this paper.

Acknowledgments. This work was partially supported by the Innovation, Science and Enterprise Council of the Junta de Andalucía (Spain), project P07-TIC-03310.

References

1. Kübler, A., Müller, K.-R.: An Introduction to Brain-Computer Interfacing. In: Dornhege, G., de Millán, J.R., Hinterberger, T., McFarland, D.J., Müller, K.-R. (eds.) Toward Brain-Computer Interfacing, pp. 1–25. The MIT Press, Massachusetts (2007)
2. Neuper, C., Pfurtscheller, G.: Motor imagery and ERD. In: Pfurtscheller, G., Lopes da Silva, F.H. (eds.) Event-Related Desynchronization. Handbook of Electroencephalography and Clinical NeuroPhysiology. Revised Series, vol. 6, pp. 303–325. Elseiver Science B.V., Amsterdam (1999)
3. Galán, F., Nuttin, M., Vanhooydonck, D., Lew, E., Ferrez, P.W., Philips, J., de Millán, J.R.: Continuous brain-actuated control of an intelligent wheelchair by human EEG. In: Proceedings of the 4th International Brain-Computer Interface Workshop and Training Course, TU Graz/Büroservice, Graz, pp. 315–320 (2008)
4. Leeb, R., Settgast, V., Fellner, D., Pfurtscheller, G.: Self-paced exploration of the Austrian National Library through thought. International Journal of Bioelectromagnetism 9(4), 237–244 (2007)
5. Leeb, R., Friedman, D., Müller-Putz, G.R., Scherer, R., Slater, M., Pfurtscheller, G.: Self-Paced (Asynchronous) BCI Control of a Wheelchair in Virtual Environments: A Case Study with a Tetraplegic. Computational Intelligence and Neuroscience. Article ID 79642 (2007)
6. Tsui, C.S.L., Gan, J.Q.: Asynchronous BCI Control of a Robot Simulator with Supervised Online Training. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 125–134. Springer, Heidelberg (2007)
7. Scherer, R., Lee, F., Schlögl, A., Leeb, R., Bischof, H., Pfurtscheller, G.: Towards self-paced Brain-Computer Communication: Navigation through virtual worlds. IEEE Transactions on Biomedical Engineering 55(2), 675–682 (2008)
8. Kronegg, J., Chanel, G., Voloshynovskiy, S., Pun, T.: EEG-based synchronized Brain-computer interfaces: A model for optimizing the number of mental tasks. IEEE Transactions on Neural Systems and Rehabilitation Engineering 15(1), 50–58 (2001)
9. Obermaier, B., Neuper, C., Guger, C., Pfurtscheller, G.: Information transfer rate in a five-classes brain-computer interface. IEEE Transactions on Neural Systems and Rehabilitation Engineering 9(3), 283–289 (2001)
10. Ron-Angevin, R., Díaz-Estrella, A., Velasco-Álvarez, F.: A two-class Brain-Computer Interface to freely navigate through virtual worlds. Biomedizinische Technik (to appear)
11. Guger, C., Schlögl, A., Neuper, C., Walterspacher, D., Stein, T., Pfurtscheller, G.: Rapid prototyping of an EEG-based brain-computer interface (BCI). IEEE Transactions on Neural Systems and Rehabilitation Engineering 11(2), 145–147 (2003)
12. Guger, C., Edlinger, G., Harkam, W., Niedermayer, I., Pfurtscheller, G.: How many people are able to operate an EEG-based brain-computer interface (BCI)? IEEE Transactions on Neural Systems and Rehabilitation Engineering 9(1), 49–58 (2001)
13. Schlögl, A., Kronegg, J., Huggins, J.E., Mason, S.G.: Evaluation Criteria for BCI Research. In: Dornhege, G., de Millán, J.R., Hinterberger, T., McFarland, D.J., Müller, K.-R. (eds.) Toward Brain-Computer Interfacing, pp. 327–342. The MIT Press, Massachusetts (2007)

Impact of Frequency Selection on LCD Screens for SSVEP Based Brain-Computer Interfaces

Ivan Volosyak, Hubert Cecotti, and Axel Gräser

Institute of Automation (IAT) - University of Bremen
Otto-Hahn-Allee, NW1, 28359 Bremen, Germany
http://www.iat.uni-bremen.de/

Abstract. In this work, the high impact of appropriate selection of visual stimuli on liquid crystal displays (LCDs) used for the brain-computer interfaces (BCIs) based on the Steady-State Visual Evoked Potentials (SSVEPs) has been confirmed. The number of suitable frequencies on the standard LCD monitor is limited due to the vertical refresh rate of 60Hz and the number of simultaneously used stimuli. Two sets of frequencies have been compared among each other during the on-line spelling task with the Bremen-BCI system in the study with 10 healthy subjects. This work is meaningful for the practical design of LCD based BCIs. In this study, appropriate selection of visual stimuli results in a 40% change in the BCI literacy under otherwise equal conditions.

Keywords: BCI (Brain-Computer Interface), SSVEP (Steady-State Visual Evoked Potential), frequency selection, LCD, visual stimuli.

1 Introduction

A brain-computer interface (BCI) is a technical system that acquires and analyzes brain neural activity patterns in real time to translate them into control commands for computers or external devices [1,2,3]. Therefore, because a BCI requires no user movement, they are particularly suited for disable or elderly persons who are unable to communicate through any classical muscular control. Steady-state visual evoked potentials are the continuous brain responses elicited at the visual and parietal cortical area under visual stimulation with a specific constant frequency; they can be used for BCI and are described as reliable in the literature [4,6,5,7]. Wang et al. [8] reviewed the recent progress of SSVEP based BCIs.

Current developments have lead to an increase of portability and practicability of BCIs. One challenge is to build a portable all-in-one SSVEP based BCI system that operates on a standard notebook. In this case LCD screen will be used as both the graphical user interface (GUI), and the source of flickering stimuli. This solution would be more convenient for the BCI users as they would not need to shift their gaze too much.

Our recent study [9] evaluated the demographics of BCI literacy, it was carried out on the international trade fair CeBit 2008 with over 106 participants.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 706–713, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The performance of the volunteers was slightly below expected when compared with previous results using the newly developed Minimum Energy Combination method published in [6]. During the study we noticed that some subjects were able to navigate the cursor over the virtual keyboard of Bremen-BCI speller, however they were unable to select the desired letter (therefore they were counted as BCI-illiterate). In this study the BCI system's visual stimuli have been changed from conventional LEDs to flickering boxes on the LCD screen of a laptop PC. This paper deals with the high impact of appropriate frequency selection due to specific characteristics of LCD screens. To do this two sets of frequencies have been compared. The original frequency set described e.g. in [10] and a new frequency set which is hypothesized to be more suitable for the LCD screen (the selection of frequencies is described in the next section):

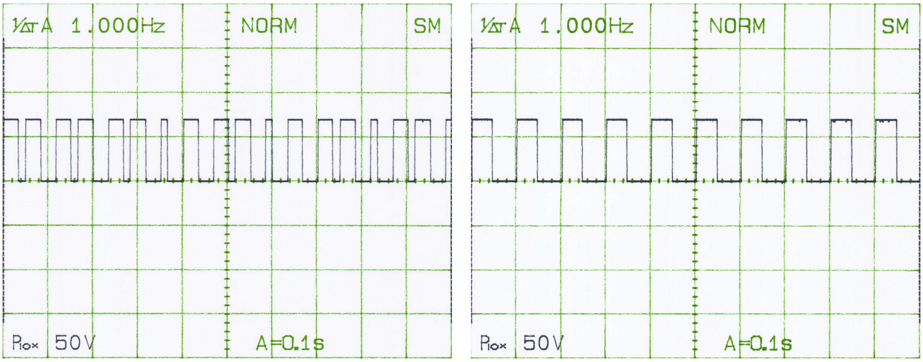
- 1. frequency set: 13.00Hz, 14.00Hz, 15.00Hz, 16.00Hz and 17.00Hz
- 2. frequency set: 7.50Hz, 8.57Hz, 10.00Hz, 12.00Hz and 6.67Hz

The paper is organized as follows: The second section presents typical problems of producing of visual stimuli on LCD screen for SSVEP based BCIs. The experimental protocol, experiments, and subjects are presented in the third section. The results are presented in the fourth section, followed by discussion and conclusion in the final section.

2 SSVEP Stimuli on LCD Screen

The SSVEP based BCI system requires the user attention to be focused on an oscillating visual stimulus. The stimuli are usually flickering lights oscillating at different frequencies. The SSVEP response to visual stimulation flickering with a frequency f_0 is well modeled as a superposition of a number of sinusoids with frequencies f_0, $2f_0$,. . . , i.e., the fundamental stimulation frequency and its harmonics.

The choice of frequencies which obtain the best response has been largely discussed in the literature. An SSVEP response is possible with a large number of frequencies, ranging from 1Hz to 100Hz, with resonance peaks at 10, 20, 40 and 80Hz [4]. The spectrum differences of three kinds of flickers and the differences in SSVEPs evoked by three different stimulators, the light-emitting diode, the cathode ray tube of a desktop monitor and the liquid crystal display of a laptop screen have been investigated in [11]. The amplitude of SSVEP response evoked by LEDs is significantly larger than that evoked by stimuli on the monitors. However, this flicker requires the custom made array of LEDs as visual stimulator with an additional hardware device for generating of frequencies; the BCI user has to switch his attention between the LEDs and the feedback on the computer screen. The use of the LCD screen on conventional PC compatible notebooks for creating the visual stimuli for SSVEP based BCIs will allow the development of cheap and portable BCI systems. This is a long-awaited step in moving from demonstration systems toward practical BCI systems. This promises the inclusion of a much larger audience for this kind of BCI.



(a) 17Hz (b) 10Hz

Fig. 1. Oscilloscope plots for different frequencies on a LCD screen

The creation of flickering boxes on a LCD screen requires the sequential rendering of two different colors at the stimulus frequency. This seems to be a trivial and very simple task. However, the appropriate selection of flickering frequencies, correct software realization under an operating system, for example the widely used Microsoft Windows, that does not run in real-time, can be a challenge. In addition, the desktop windows manager of Windows Vista determines visualization effects (for example the Aero effect, which is generally turned on by default installation), which provides smoother picture on the screen, but can also impair the generated frequencies. Therefore, the stability of the frequencies has to be checked precisely, preferably with an external device, in order to check the assumption that the SSVEP deteriorations result from the differences of the flickering signals. In this work, the frequencies on the LCD screen were checked with a purpose-built hardware tool which consisted of a photo transistor BP103 (the sensor) and an amplifier BC547. Digital justification of the signal fronts was done with two elements 4093N, for easy evaluation of the frequencies. Figure 1(a) presents the observed oscilloscope signal for stimulating frequency 17Hz. Although the number of observed signal periods is correct, there are some artefacts and the signal is not homogeneous. Unstable signals such as these may still produce an effective SSVEP response, but generally a better SSVEP response is generated with a stimulus at a constant frequency.

It is thought that in order to produce stable frequencies on the LCD screen the frequencies should be chosen as divisors of the standard vertical refresh rate of 60Hz (periods equivalent to 2, 3,. . . frames on the LCD screen). This ensure that only whole frames are seen on the screen. Based on the above, suitable frequencies are: 12.00, 10.00, 8.57, 7.50 and 6.67Hz for the alpha band, 20.00 and 15.00Hz for the beta band. Figure 1(b) confirms this hypothesis with the clear and stable signal for the flickering frequency of 10Hz under otherwise equal conditions.

The classical two dimensional BCI control requires five classes: four classes are dedicated to the directions (up, down, left and right) and one class for action (select). Gao et al. [5] gives many reasons for choosing the low-frequency band (6-15Hz) for SSVEP visual stimulation. There are 7 frequencies, 6.00, 6.67, 7.50, 8.57, 10.00, 12.00, 15.00Hz, which belong to this frequency band. After removing two boundary values, the frequencies 6.67, 7.50, 8.57, 10.00 and 12.00Hz were chosen for the experiments. In [12], the impact of harmonic frequency components for SSVEP-based communication was analyzed in detail. They recommended SSVEP processing under consideration of first three harmonics of stimulus frequency i.e. f_0, $2f_0$, $3f_0$. However, only the first two harmonics of suggested frequencies can be used for SSVEP detection in this case; under consideration of three harmonics the frequencies f_0 = 10.00Hz and 6.67Hz cannot be used simultaneously, because second harmonic of 10.00Hz is equal to the third harmonic of 6.67Hz.

The assumption about the high impact of appropriate selection of visual stimuli on liquid crystal displays (LCDs) used for the brain-computer interfaces (BCIs) based on the Steady-State Visual Evoked Potentials (SSVEPs) was successfully validated in experiments.

3 Experiments

The experiments were carried out in normal office room in the Institute of Automation at the University of Bremen. This is worth noting that this is different to the usual EEG recording conditions which is usually an electrically shielded room with low background noise and luminance.

3.1 Subjects

Experiments were performed on 10 healthy volunteer subjects. Every subject uses daily a computer screen at work. Eight of the subjects were male and two were female, all subjects had normal or corrected-to-normal vision. Table 1 presents the demographic distribution of the subjects. Subjects did not receive any financial reward for participating in this study.

Table 1. Demographic distribution of subjects

Subject number	1	2	3	4	5	6	7	8	9	10
Age	30	29	24	27	24	26	25	28	31	28
Gender	M	M	M	M	M	M	F	M	F	M

3.2 Materials and Hardware

Subjects were seated approximately 60 cm from the LCD screen (15.4” with resolution of 1280 x 800 pixels) of the notebook running the BCI software. The graphical user interface of the used SSVEP based Bremen-BCI speller is shown in the Fig. 2. The notebook has an Intel Core 2 Duo (2x1.50GHz) processor

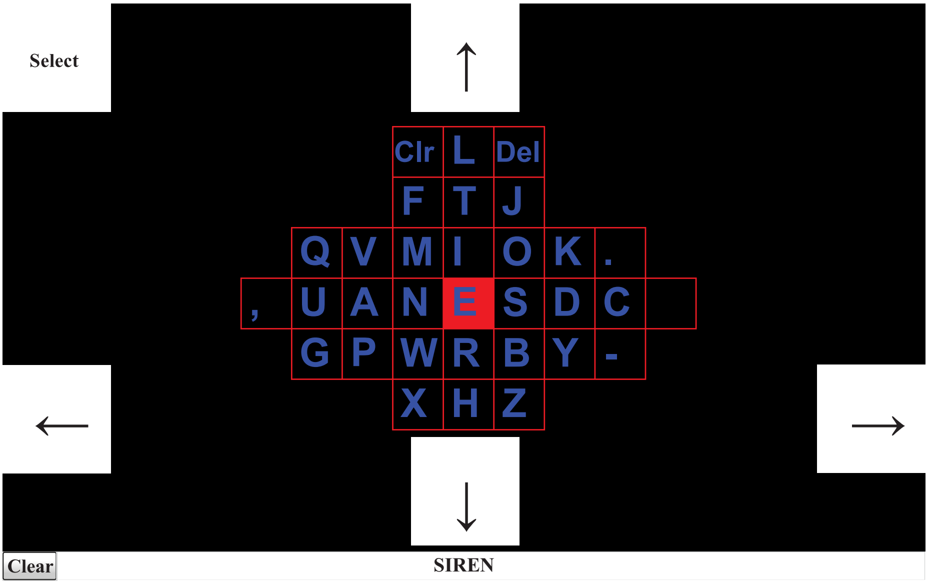


Fig. 2. GUI of the SSVEP based Bremen-BCI speller

running Microsoft Windows Vista. The EEG data were recorded from the surface of the scalp via 8 standard Ag/AgCl EEG electrodes. They were placed on sites $P_Z, PO_3, PO_4, O_Z, O_9, O_{10}$; AF_Z was used for ground and C_Z was used for the reference electrode on the international system of electrode placement. Standard abrasive electrolytic electrode gel was applied between the electrodes and the skin in order to bring impedances below $5k\Omega$. An EEG amplifier g.USBamp (Guger Technologies, Graz, Austria) was used and the sampling frequency was 128Hz. During the EEG acquisition, an analog bandpass filter between 2 and 30Hz, and a notch filter around 50Hz (mains frequency in Europe) were applied directly in the amplifier.

The virtual keyboard with 32 characters (letters and special symbols) is located in the middle of the screen. The five white boxes (150x150 pixels) at outer edges and upper left corner of the screen are flickering as described above (see Fig. 1) and correspond to the commands "left", "right", "up", "down", and "select" respectively. At the beginning of each spelling trial, the cursor is located in the middle of the virtual keyboard over the 'E' character. By focusing on one of five oscillating boxes subjects navigate the cursor to the desired letter. Audio feedback follows every recognized command. The spelling is made by activating the command "select" for the chosen letter. After every selection the cursor automatically moves back to the initial letter 'E'. Therefore, at least nine commands are needed for spelling the word "SIREN" ("right", "select", "up", "select", "down", "select", "select", "left", "select"). The word SIREN was chosen as the shortest word which requires all moving commands in order to be spelled. In the case of a spelling error the last letter or the whole spelled text can be erased by selecting the special characters 'Del' or 'Clr' respectively. The current status of the spelling (the text that the subject has already spelled) is shown in the line at the bottom on the screen, see the text "SIREN" in the Fig. 2. The special letter arrangement takes into account the probabilities of occurrence of each letter in English, it is not possible to move over the layout boundaries, e.g. from the letter 'H' to the letter 'L' by choosing the "down" command. The signal-to-noise ratio (SNR) was calculated on-line every 13 samples (ca. 100ms) on the basis of the last 2s of acquired data (time segment length). After every command detection an idle time of 2s was applied in order to prevent repeated selection of the last spelled command. More details about the used SSVEP detection method can be found in [6].

3.3 Experimental Protocol

After completing the consent form, subjects were prepared for EEG recording as described above. A short familiarization run was carried out in order to introduce the speller application to the subjects. The task was to spell the word "SIREN" with the SSVEP based Bremen-BCI system. The first run was done with the frequency set 1, which was used during the CeBit study and the second run with the new frequencies, frequency set 2. Before every spelling run, the SNR threshold was manually adjusted for every subject. All data collected during the experiment were stored anonymously.

4 Results

The resulting spelling times, number of commands with corresponding accuracies and information transfer rates (calculated as described in [3]) for each set of frequencies are reported in Table 2. As already mentioned, at least nine commands are needed for spelling the word "SIREN". A larger number of commands indicates wrong detections. Either of moving commands (up, down, left or right) or selection commands, which could be corrected using the special 'Del' character. Among 10 subjects, four subjects were not able to use the BCI with the first set of frequencies. For these subjects Table 2 contains null values. With the new set of frequencies that have been suggested, all the subjects were able to cope with the spelling task. Furthermore, the new frequencies show an improvement of the ITR and the accuracy for almost all of subjects (except for subject 7, which achieved the overall highest ITR with the first frequency set).

This improvement could be due to the stability of the stimuli frequencies but it could be also due to the fact that they are simply different. The type of flicker used, as well as the frequency used is also very important. Subject 2 had extensive experience with the Bremen-BCI speller based on the LEDs. In this prior work, this subject always achieved very high ITR, comparable with the results of subject 7. For this reason it was completely unexpected that subject 2 was unable to accomplish the spelling task with the stimuli on the LCD screen

Table 2. Experimental Results

	1. Frequency Set 13.00, 14.00, 15.00, 16.00, 17.00Hz				**2. Frequency Set** 7.50, 8.57, 10.00, 12.00, 6.67Hz			
Subject #	**Com.** [-]	**Acc.** [%]	**Time** [s]	**ITR** [bpm]	**Com.** [-]	**Acc.** [%]	**Time** [s]	**ITR** [bpm]
1	11	90.91	79.32	17.04	9	100.00	44.79	27.99
2	0	0.00	0.00	0.00	9	100.00	135.48	9.25
3	0	0.00	0.00	0.00	9	100.00	46.11	27.19
4	0	0.00	0.00	0.00	9	100.00	39.10	32.07
5	9	100.00	65.31	19.20	9	100.00	60.63	20.68
6	9	100.00	126.65	9.90	16	68.75	103.69	14.85
7	9	100.00	21.33	58.78	9	100.00	34.23	36.63
8	9	100.00	45.20	27.74	9	100.00	41.74	30.04
9	0	0.00	0.00	0.00	15	93.33	55.76	34.06
10	13	84.62	66.42	22.31	11	90.91	51.70	26.14
Min	0	0.00	0.00	0.00	9	68.75	34.23	9.25
Max	13	100.00	126.65	58.78	16	100.00	135.48	36.63
Mean	6.00	57.55	40.42	**15.50**	10.50	95.30	61.32	**25.89**
S.D	5.31	49.78	43.67	18.49	2.72	9.90	32.55	8.63

using the first frequency set. This proves that the problem was due to the display (the signal form, stability of the frequencies, the contrast and the luminance of visual stimuli are different) and not from the signal processing. However, judicious choice of frequencies allows this subject to use the BCI speller with the LCD display. For some subjects, the quality of the stimuli defines the access to the SSVEP based BCIs.

5 Discussion and Conclusion

The experiments conducted in this paper have shown the high impact of judicious frequency selection on LCD screens for SSVEP based BCIs. This work is meaningful for the practical design of LCD based BCI applications. Appropriate selection of visual stimuli results in a 40% change in the BCI literacy under otherwise equal conditions (see Table 2).

The subjects in this study spelled only one word "SIREN" with two different sets of frequencies. Nevertheless the results can be treated as representative and they are in line with results of our new study with 37 naive subjects (without any SSVEP-BCI experience), including eight handicapped users, performed on the international rehabilitation fair RehaCare2008 in Düsseldorf, Germany, with the new frequency set proposed in this paper.

Our future work will address an extensive analysis of all collected data (during this experiment and during the fairs CeBit 2008, RehaCare 2008), statistical evaluation of results, demographical analyses, comparisons between healthy and people with disabilities etc. in order to facilitate BCIs that can adapt to each user, ideally with little or no expert help.

Acknowledgment. This research was supported by a Marie Curie European Transfer of Knowledge project BrainRobot, MTKD-CT-2004-014211, within the 6th European Community Framework Program and by a Marie Curie European Re-Integration Grant RehaBCI, PERG02-GA-2007-224753, within the 7th European Community Framework Program.

References

1. Berger, T.W., Chapin, J.K., Gerhardt, G.A., McFarland, D.J., et al.: Brain-Computer Interfaces: An international assessment of research and development trends. Springer, Heidelberg (2008)
2. Dornhege, G., Sejnowski, T.J.: Toward Brain Computer Interfacing. MIT Press, Cambridge (2007)
3. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. J. Clin. Neurophysiol. 113(6), 767–791 (2002)
4. Herrmann, C.S.: Human EEG responses to 1-100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. Exp. Brain Res. 137(3-4), 346–353 (2001)
5. Gao, X., Xu, D., Cheng, M., Gao, S.: A BCI-based environmental controller for the motion-disabled. J. Neural Syst. Rehabil. Eng. 11(2), 137–140 (2003)
6. Friman, O., Volosyak, I., Gräser, A.: Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces. J. of Biomed. Eng. 54(4), 742–750 (2007)
7. Middendorf, M., McMillan, G., Calhoun, G., Jones, K.S.: Brain-Computer Interfaces based on the Steady-State Visual Evoked response. J. Rehab Eng. 8(2), 211–214 (2000)
8. Wang, Y., Gao, X., Hong, B., Jia, C., Gao, S.: Brain-Computer Interfaces based on visual evoked potentials. J. Med. Biol. Mag. 27(5), 64–71 (2008)
9. Allison, B., Volosyak, I., Lüth, T., Valbuena, D., et al.: BCI Demographics: How many (and what kinds of) people can use an SSVEP BCI? In: Proceedings of the 4th International Brain-computer Interface Workshop and Training Course, Graz, Austria, pp. 333–338 (2008)
10. Valbuena, D., Cyriacks, M., Friman, O., Volosyak, I., Gräser, A.: Brain-Computer Interface for high-level control of rehabilitation robotic systems. In: Proc. IEEE ICORR 2007, Noordwijk, The Netherlands, pp. 619–625 (2007)
11. Wu, Z., Lai, Y., Xia, Y., Wu, D., Yao, D.: Stimulator selection in SSVEP-based BCI. J. Medical Engineering and Physics 30(8), 1079–1088 (2008)
12. Müller-Putz, G.R., Scherer, R., Brauneis, C., Pfurtscheller, G.: Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. J. Neural Eng. 2(4), 123–130 (2005)

Multiobjective Evolutionary Algorithms: Applications in Real Problems*

Antonio Berlanga[1], Jesús García Herrero[1], and José Manuel Molina[1]

[1] Departament of Computer Science, Universidad Carlos III de Madrid, Spain
{antonio.berlanga,jesus.garcia,josemanuel.molina}@uc3m.es

Abstract. The concept of optimization refers to the process of finding one or more feasible solutions of a problem which corresponds to the extreme values (either maximum or minimum) of one or more objective functions. Initial approaches to optimization were focused on the case of solving problems involving only one objective. However, as most real-world optimization problems involve many objectives the research on this area has rapidly broaden this attention to encompass what has been called multi-objective optimization.

Keywords: Multiobjective Optimization, Real World Problem.

1 Introduction

When moving from the single-objective approach to the multi-objective one the question of objective trade-offs arises. This is caused by the fact that a solution that might be optimal with respect to one objective would be less-than-optimal for the rest. Therefore the globally optimal solution to a given problem would be a compromise between feasible ones of the different objectives.

This leads us to the dilemma that hovers over all multi-objective optimization algorithms: how to select a solution from the set of feasible ones. This selection implies the use of high-level pondering criteria. An ideal strategy for solving this issue can be devised as: (1) find a set of partially optimal solutions using different selection criteria, and then; (2) choose among the elements of the set using a high-level knowledge.

Classical approaches to this class of problems are based on preference-based search procedures. In these procedures fitness values produced by each objective are scaled via a relative preference vector. This kind of process is repeated on a point-by-point basis generating a (hopefully) better solution after each iteration. This field has been extensively studied in last years by Evolutionary Computation (ECs) researchers [1], analyzing the properties and advantages for different families of algorithms. Two basic types of approaches can be distinguished, on the one hand the aggregation of all objectives into a single scalar function, applying then standard optimization methods, and on the other hand the use of specific multi-objective evolutionary algorithms

---

* This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, SINPROB, CAM MADRINET S-0505/TIC/0255 and DPS2008-07029-C02-02.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 714–719, 2009.
© Springer-Verlag Berlin Heidelberg 2009

(MOEAs), exploiting the concept of multi-objective domination. In this second case, the algorithms preserving dominance are intended to maintain diverse population and provide a representation of the whole Pareto-optimal front after a single optimization run. These methods apply operations to preserve the population diversity, such as niching or clustering, and all non-dominated solutions are given similar emphasis with respect to dominated solutions, avoiding their collapse in a single point of the front.

2 Multiobjective Problem Definition

In general terms, the multi-objective optimization problem can be formally stated as:

$$\begin{array}{rrr} \text{minimize or maximize} & f_m(\mathbf{x}), & m=1,\ldots,M; \\ \text{suject to} & \mathrm{g}_\mathrm{j}(\mathbf{x})\geq 0, & j=1,\ldots,J; \\ & h_k(\mathbf{x})=0, & k=1,\ldots K; \\ & x_i^l \leq x_i \leq x_i^u, & i=1,\ldots,n. \end{array} \tag{1}$$

Where $\mathbf{x}\in\Re^n$ is the vector of variables to be optimized, each bounded by $\mathbf{x}_i^l$ and $\mathbf{x}_i^u$; $f_m(\mathbf{x})$ are the objective functions being minimized or maximized and $g_j(\mathbf{x})$ and $h_k(\mathbf{x})$ are the inequality and equality constraints, respectively. We can refrase any maximization objective as a minimization one (and vice versa). Thanks to this and because of the fact that we are not going to deal with constraints we can rewrite as:

$$\text{minimize} \quad \mathbf{z}_m = f_m(\mathbf{x}), \quad m=1,\ldots,M; \tag{2}$$

Here the vector z_m designates the quality of a given solution and is known as the objective vector. In order to rank the quality of a solution with respect with the others the concept of dominance has been introduced.

Definition: A solution $\mathbf{x}^i$ is said to dominate a solution $\mathbf{x}^{i*}$ ($\mathbf{x}^i \succ \mathbf{x}^{i*}$) if both of the following conditions hold true:

1. Solution $\mathbf{x}^i$ is not worse than $\mathbf{x}^{i*}$ for all objectives: $\forall j\in\{1,2,\ldots,M\}: f_j(\mathbf{x}^i)\leq f_j(\mathbf{x}^{i*})$
2. Solution $\mathbf{x}^i$ is strictly better than $\mathbf{x}^{i*}$ for at least one objective: $\exists j\in\{1,2,\ldots,M\}: f_j(\mathbf{x}^i)> f_j(\mathbf{x}^{i*})$.

An intuitive consequence of the above definition is that if a solution $\mathbf{x}^i$ dominates a solution $\mathbf{x}^{i*}$ then its understood that $\mathbf{x}^i$ is a better solution than $\mathbf{x}^{i*}$. On the same line of thought, the set of non-dominated solutions is defined as the set of solutions that are not dominated by any other. If a set of non-dominated solutions is built upon the whole set of feasible solutions it is called globally Pareto-optimal set.

3 Multiobjective Evolutionary Algorithms (MOEAs)

The algorithms which search solutions to simultaneously optimize several objectives have recently received a considerable attention. The key of Evolutionary Computation [2] success in this field is the potentially capability to search and handle a population of solutions forming a representative sample of the Pareto-optimal set [3].

This set is defined making use of the domination concept in multi-objective optimization problems. When there are several objectives involved, different solutions cannot be directly compared and ranked to determine the best one as the single-valued function case, but the stronger concept of domination appears. A solution **x** is dominated by solution, **y**, if is better than simultaneously in all objective functions considered. In any other case they could not be strictly compared to select the best one (unless a numeric criterion for trade-off is explicitly defined). Taking into account this concept of domination, a Pareto-optimal set, P, is defined as the set such there does not exist any solution in the search space dominating any member in P.

The MOEAs have the double goal of guiding the search towards the global Pareto-optimal set and at the same time cover as many solutions as possible. This implies the enlargement of population size and besides specific procedures in the genetic operators to guarantee a well-distributed sample of the front. The textbook by Deb [4] presents a wide analysis of the most outstanding multi-objective evolutionary algorithms. There are several tens of different approaches, depending on the types of operators for selection, mutation and use of domination. Basically, they can be divided into two groups, non-elitist and elitist algorithms. The non-elitist group includes first approaches, such as VEGA (Vector Evaluated GA) [5]), which divides population in subsets assigned to different objectives, or most recent MOGA (Multi-objective GA) [6], NSGA (Non-dominated Sorting GA) [7] and NPGA (Niched-Pareto GA) [8], PESA (Pareto Envelope-Based Selection Algorithm) [9]. The last algorithms apply non-dominated classification in the GA population, with variations in the type of selection operators. The elitist approaches to MOEA are the most recent techniques, and explicitly use operators to preserve the elites of population in the offspring generation of to assure that fitness does not decrease. Among them, we can mention the NSGA-II (Non-dominated Sorting GA-II) [10], SPEA (Strength Pareto EA) [11], SPEA2 [12], PAES (Pareto-Archived Evolution Strategy) [13], [9], PESA-II [14], CAEP (Cultural Algorithms with Evolutionary Programming) [15], etc. These algorithms exploit elitism by explicitly keeping a population with non-dominated solutions until now, so the selection operators are based on comparisons with them. A thorough survey regarding the above models and many others can be found in [16] and elsewhere.

4 Applications in Real Problems

In a real problem, we define a partial cost for objective and situation $\mathbf{c}_{ij}$: $c_{ij}(\vec{x}) = R(f_{ij}(\vec{x}) - s_{ij})/s_{ij}$, being s_{ij} the expected value for i-th objective under j-th situation; $f_{ij}(.)$ is the corresponding performance function evaluated on input $\vec{x}$;

and R(.) is the ramp function to only take into account excesses over s_{ij}. Considering this formulation with normalized costs $c_{ij}(\vec{x})$, some aggregation options are:

A. Weighted sum: It is the most direct evaluation is the weighting sum on partial objective functions:

$$\text{fitness}_{\text{SUM}} = \sum_{i=1}^{Nm} \sum_{j=1}^{Ns} w_{ij} c_{ij}(\vec{x}) \tag{3}$$

Optimization of weighted sums converge to particular solutions of Pareto front, corresponding to the tangential points in the directions defined by the weights vectors w_i [17]. All potential non-dominated solutions can be obtained in the case that the Pareto front has a convex shape. In other case, local minima appear in the front, and some points may be left unreachable with this procedure.

B. Weighted m-distance: the limitations of weighted linear combination may be overcome by means of a generalization with a weighted m-metric computed over objective functions:

$$\text{fitness}_{\text{m-dis tan ce}} = \left[\sum_{i=1}^{Nm} \sum_{j=1}^{Ns} \left(w_{ij} c_{ij}(\vec{x}) \right)^{m} \right]^{\frac{1}{m}} \tag{4}$$

In that case, any point in the front is potentially reachable, independently of its convexity, with an optimizer by choosing the appropriate value for parameter m.

C. Maximum metric: in the limit of previous case, when $m \rightarrow \infty$, we have the Tchebycheff (or maximum) metric:

$$\text{fitness}_{\max} = \max_{i,j} \left\{ w_{ij} c_{ij}(\vec{x}) \right\} \tag{5}$$

A theoretical result [18] shows that any point in the Pareto front can be reached by solving the weighted Tchebycheff problem with the appropriate weight vector, independently of the front shape.

D. WCES [19]: the aggregation proposed for this type of problems is a variation of maximum operation, taking into account that evaluated functions (objectives 1…Nm) will have a similar behaviour among the design scenarios (1…Ns), varying only the particular values and shapes in every one.

$$\text{fitness}_{\text{WCES}} = \sum_{i=1}^{N_m} \left(\max_{j \in \{1,\ldots,N_s\}} \left\{ c_{ij}(\vec{x}) \right\} \right) \tag{6}$$

E. Multi-objective functions: Conversely, the application of multi-objective algorithms to this problem, formulated with normalized costs, is:

$$\vec{f}_{MO}=\left[c_{11}(\vec{x}),c_{21}(\vec{x}),\ldots,c_{N_m1}(\vec{x}),\ c_{12}(\vec{x}),c_{22}(\vec{x}),\ldots,c_{N_m2}(\vec{x}),\ \ldots c_{1N_s}(\vec{x}),c_{2N_s}(\vec{x}),\ldots,c_{N_mN_s}(\vec{x})\right] \quad (7)$$

and the optimization address the NmNs objectives at the same time.

F. Aggregated multi-objective: Between the pure multiobjective approach and the scalar aggregated function, an intermediate approach has been considered, an aggregated multi-objective problem:

$$\vec{f}_{AG\text{-}MO}=\begin{bmatrix} \max\limits_{j\in\{1,\ldots,N_s\}} \left\{c_{1j}(\vec{x})\right\}, & \max\limits_{j\in\{1,\ldots,N_s\}} \left\{c_{2j}(\vec{x})\right\}, & \ldots & \max\limits_{j\in\{1,\ldots,N_s\}} \left\{c_{N_mj}(\vec{x})\right\} \end{bmatrix} \quad (8)$$

in this case the optimization would address only Nm objectives, the worst cases for each magnitude in a similar way as WCES aggregation.

References

1. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
2. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, Inc., Oxford (1996)
3. Deb, K.: Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In: Evolutionary Algorithms in Engineering and Computer Science, ch. 8. John Wiley & Sons Ltd., Chichester (1999)
4. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
5. Shaffer, J.H.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: Proc. of the First International Conference on Genetic Algorithms, pp. 93–100 (1985)
6. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multibojective optimization: Formulation, discusión and generalization. In: Forrest, S. (ed.) Proceedings of the Fifth Int Conf on Genetic Algorithms, pp. 416–423. Morgan Kauffman, San Mateo (1993)
7. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionay Computation 2(3), 221–248 (1994)
8. Horn, J., Nafpliotsis, N., Goldberg, D.E.: A niched pareto genetic algorithm for multiobjective optimization. In: Proc. of the First IEEE Conf on Evolutionary Computation. IEEE World Congress on Computational Computation, Piscataway, NJ, vol. I, pp. 82–87. IEEE Press, Los Alamitos (1994)
9. Corne, D.W., Knowles, J.D., Oates, M.J.: The pareto envelope-based selection algorithm for multiobjective optimisation. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 839–848. Springer, Heidelberg (2000)
10. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on Evolutionary Computation 6(2), 182–197

11. Zitzler, E., Thiele, L.: An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical report, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology, ETH (1998)
12. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland (May 2001)
13. Knowles, J.D., Corne, D.W.: The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation. In: Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999), vol. 1, pp. 98–105 (1999)
14. Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J.: PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp. 283–290. Morgan Kaufmann Publishers, San Francisco (2001)
15. Becerra, R.L., Coello, C.A.: A Cultural Algorithm for Solving the Job-Shop Scheduling Problem. In: Jin, Y. (ed.) Knowledge Incorporation in Evolutionary Computation. Studies in Fuzziness and Soft Computing, vol. 167, pp. 37–55. Springer, Heidelberg
16. Tan, K.C., Khor, E.F., Lee, T.H.: Multiobjective Evolutionary Algorithms and Applications. Series: Advanced Information and Knowledge Processing. Springer, United Kingdom (2005)
17. Deb, K.: Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In: Evolutionary Algorithms in Engineering and Computer Science, ch. 8. John Wiley & Sons Ltd., Chichester (1999)
18. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Publishers, Boston (1999)
19. Herrero, J.G., Berlanga, A., Lopez, J.M.M.: Effective Evolutionary Algorithms for Many-Specifications Attainment: Application to Air Traffic Control Tracking Filters. IEEE Transactions on Evolutionary Computation 13(1), 151–168 (2009)

Evolutionary Genetic Algorithms in a Constraint Satisfaction Problem: Puzzle Eternity II

Jorge Muñoz, German Gutierrez, and Araceli Sanchis

University Carlos III of Madrid
Avda. de la Universidad 30, 28911 Leganés, Spain
{jmfuente,ggutierr,masm}@inf.uc3m.es

Abstract. This paper evaluates a genetic algorithm and a multiobjective evolutionary algorithm in a constraint satisfaction problem (CSP). The problem that has been chosen is the Eternity II puzzle (E2), an edge-matching puzzle. The objective is to analyze the results and the convergence of both algorithms in a problem that is not purely multi-objective but that can be split into multiple related objectives. For the genetic algorithm two different fitness functions will be used, the first one as the score of the puzzle and the second one as a combination of the multiobjective algorithm objectives.

1 Introduction

The Eternity II puzzle [1] is a constraint satisfaction problem created as a challenge, with a prize, to know if someone is able to solve it. The difficulty lies on the huge search space of the problem, there is not any algorithm that can find a solution in a reasonable time. It is an edge-matching puzzle with 256 tiles that are differents one to each other. The tiles have a color pattern in each of its edges and there are not two tiles with the same color pattern in all their edges. All the tiles must be placed in a 16×16 board, and can be rotated 90°, 180° or 270° before being placed. There are tiles with a grey pattern in their edges that must be placed in the corners and the borders of the board. These grey edges must be placed next to the border. There is also known the cell and position of one of the tiles but we will omited this information because it is irrelevant for the objectives of this paper.

If we use the information about the tiles that have to be placed into the corners and the border of the board, then the search space of this problem is approximately $4^{196} \cdot 196! \simeq 10^{485}$. A more accurate approximation can be calculated with the Equation 1 where n is the number of tiles.

$$\begin{aligned} S &= 4! + (4 \cdot (\sqrt{n} - 2))! + 4^{(\sqrt{n}-2)^2} \cdot ((\sqrt{n} - 2)^2)! \\ &\simeq 4^{(\sqrt{n}-2)^2} \cdot ((\sqrt{n} - 2)^2)! \end{aligned} \tag{1}$$

Our objective is not to solve the puzzle, but satisfy the greatest number of constraints with evolutionary algorithms in the minimum possible time. A

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 720–727, 2009.
© Springer-Verlag Berlin Heidelberg 2009

constraint is the join of two adjacent edges of tiles, if the adjacent edges match then the constraint is satisfied. We do not bear in mind the edges that must be placed in the border of the board, only the inner ones among tiles. We will called the number of constraints satisfied the score. In a squared board of $m \times m$ cells the maximum score that can be reached is obtained by Equation 2. For the Eternity II the maximum score is 480.

$$\text{score} = m \cdot (m-1) + (m-1) \cdot m = 2 \cdot m \cdot (m-1) \tag{2}$$

The edge-matching puzzles are NP-complete problems [2], specifically they are constraint satisfaction problems (CSPs) [3]. When the problem is computationally intractable and we want to satisfy the maximum number of constraints we call it a MAXSAT [4] problem and here is where an evolutionary algorithm can be used [5].

In the rest of the document it will be shown some related work of evolutionary algorithms in combinatorial problems (*Section 2*), a description of the search algorithms used (*Section 3*), the experimental results of the evolutionary algorithms with different parameters (*Section 4*) and, finally, the conclusions and future works (*Section 5*).

2 Related Work

Constraint satisfaction problems are solved through two different techniques: inference and search; although sometimes a combination of both is used [6,3]. The inference modifies the problem to create another one that is easier to solve, normally this change is made with domain information. The main algorithms in inference are the consistency algorithms, also known as constraint propagation [3]. In search the most known algorithm is backtracking [6], this algorithm essentially performs a depth-first search of the space of potential solutions. The lack of the search algorithms is that they do not use the domain information to improve the search and they repeat the same mistakes, that is, they repeat lot of times searches that do not lead to a solution and could be avoided with some domain information.

The exhaustive search is the unique algorithm that grants to find a solution if it exists, but in problems with big search space this algorithm is useless because the time required to finish. So in this sort of problems, like some MAXSAT, a local search is performed instead of an exhaustive search. The local search algorithms starts with a set of assigned variables and tries to allocate the rest of variables. When it is reached a situation where a constraint is violated the algorithm modify the values of the variables that break the constraint. The best known results in the Eternity II were obtained with an algorithm of these features that uses an hibridization of constraint propagation and very large neighborhood stochastic local search [7], it satisfies 458 of the 480 constraints.

Another way to find solutions in a MAXSAT problem is through the use of evolutionary algorithms [8]. Some of the algorithms in the literature are SAWEA [9,10], RFEA [11,12], FlipGa [13], ASAP [14] and genetic algorithm [5], although a clasic algorithm as FC-CBJ [15] outperforms these kind of evolutionary algorithms [16].

3 Search Algorithms

Besides the genetic algorithm and the multiobjective evolutionary algorithm, we have used an exhaustive search algorithm and random search. The exhaustive search is a backtracking algorithm that places the tiles in the board but with an improvement. When the maximal deep of search is reached, instead of do the backtracking in that moment, the algorithm continues browsing the board and placing tiles in the right way with the cells where any tile can be placed left in blank. When it finishes the backtracking is performed.

Next, it will be seen a brief description of the evolutionary algorithms and the two fitness functions used in the genetic algorithms.

3.1 Genetic Algorithm (GA)

In the experiments we have used a simple genetic algorithm with elitism and tournament selection. Codification, crossover, mutation and fitness function will be described below.

The representation of a solution is a bidimensional matrix where each cell contains a tile and its orientation. This representations keeps the spacial relations among the adjacent tiles between generations, so specific crossover operator and mutation have to be used.

The crossover operator for the coding is based on the regions exchange. Two parents are selected, then a region of random size in a random point is chosen. Then the outer cells of the region are copied into the offspring from one of the parents, and the inner from the other one. In this operation it has to bear in mind that there can not be duplicated or missed tiles, this must be avoided. An example of the crossover operator is shown in Fig. 1.

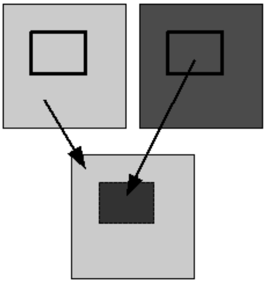
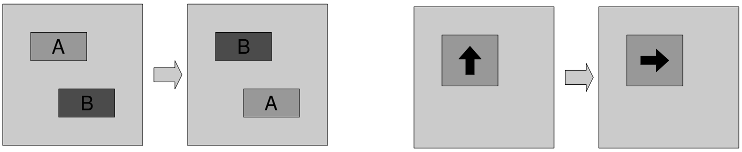


Fig. 1. A simple example of crossover

The mutation operators are two, when a mutation is going to be applied only one of the operators is chosen randomly. One mutation operator is the region exchange (Fig. 2(a)), this operator exchanges two regions of the same size that are not overlap. The second mutation operator is the region rotation (Fig. 2(b)), this operator rotates a squared region.



(a) Exchange (b) Rotation

Fig. 2. Examples of mutations

For crossover and mutation operators, the height and width of the regions are chosen randomly between two bounds. The minimum size of the region for crossover is 2 and the maximum 8, and for mutation the bounds are 1 and 10.

In this algorithm two fitness functions have been used, both of them will try to be minimized. The maximum value of this fitness functions is 1 and the minimum is 0, where 0 also means that the puzzle is solved. The first one, called the *normal fitness*, it is one minus the normalized score (see Equation 3). The second fitness is a combination of the three objectives in the multiobjevite algorithm, so it is called the *combined fitness*. The equation to calculate this *combined fitness* is shown in Equation 4 where k is the number of objectives, $objective_i$ is the value of the objective i and $max\ objective_i$ is the maximum value of the objective i (in Section 3.2 the objectives will be described).

$$\text{normal fitness} = 1 - \frac{\text{score}}{480} \tag{3}$$

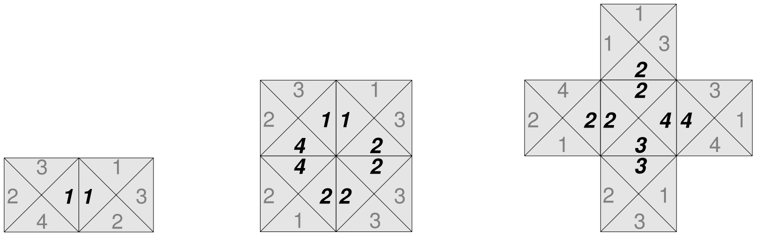
$$\text{combined fitness} = 1 - \left(\frac{1}{k} \cdot \sum_{i=1}^{k} \frac{\text{objective}_i}{\text{max objective}_i} \right) \tag{4}$$

3.2 MultiObjective Evolutionary Algorithm (MOEA)

The multiobjective evolutionary algorithm has been developed from the concepts used in the NSGA-II algorithm [17]. But three modifications have been done to improve the performance in the problem. The chromosome, crossover operator and mutation operator are the same as in the genetic algorithms we saw before.

The first improvement is how the dominance is calculated. Instead of use fronts of dominance, the dominance of one individual is calculated by counting the number of individuals that dominate that individual. This is faster than calculate the fronts of dominance.

The second improvement is how the distance among individuals is calculated. Only the distances among the individuals of the pareto front are calculated. This distance is through the diversity in the values of the objectives of the individuals. If the same objective in both individuals has different values the distance between that individuals is increased one unit, thus the maximum distance between two individuals is the number of objectives and the minimum is 0.



(a) Objective 1 (b) Objective 2 (c) Objective 3

Fig. 3. The three objectives of the MOEA

The third improvement is the number of elitism individuals that are copied from one generation to the next one. Only a fixed number of individuals in the pareto front are copied. The first ones that are copied are those ones which have more distance with the rest of individuals.

The objectives that have to be maximized for the problem are three:

- *Objective 1*: The score of the puzzle explained in *Section 1* (Fig. 3(a)). The maximum value of this objective is 480 and the minimum is 0.
- *Objective 2*: The number of square regions of four tiles (2 × 2 regions) that match in their adjacent inner sides (Fig 3(b)). The maximum value is 225 and the minimum is 0.
- *Objective 3*: The number of tiles that have their four sides matched with the adjacent tiles (Fig. 3(c)). The maximum value is 256 and the minimum is 0.

4 Experimental Results

In the experimentation it has been launched a random algorithm (RAND), the exhaustive search (ES) explained in the beginning of Section 3, a genetic

Table 1. Results

Experiment	Algorithm	Cross.	Mut.	Elit.	Fitness	Max.	Min.	Mean	Std. Dev.
RAND	RAND	-	-	-	-	49	45	46.5	± 1.02
ES	ES	-	-	-	-	369	357	363.7	± 3.66
GA1	GA	9000	999	1	norm	316	286	295.9	± 7.76
GA2	GA	5000	4999	1	norm	160	151	157.3	± 2.53
GA3	GA	7500	2500	0	norm	130	75	106.7	± 18.48
GA1C	GA	9000	999	1	comb	355	337	347.9	± 6.5
GA2C	GA	5000	4999	1	comb	357	146	210.4	± 90.03
GA3C	GA	7500	2500	0	comb	366	348	357.5	± 5.48
MOEA1	MOEA	9000	990	10	-	367	345	353.7	± 6.56
MOEA2	MOEA	5000	4990	10	-	371	350	361.7	± 5.1

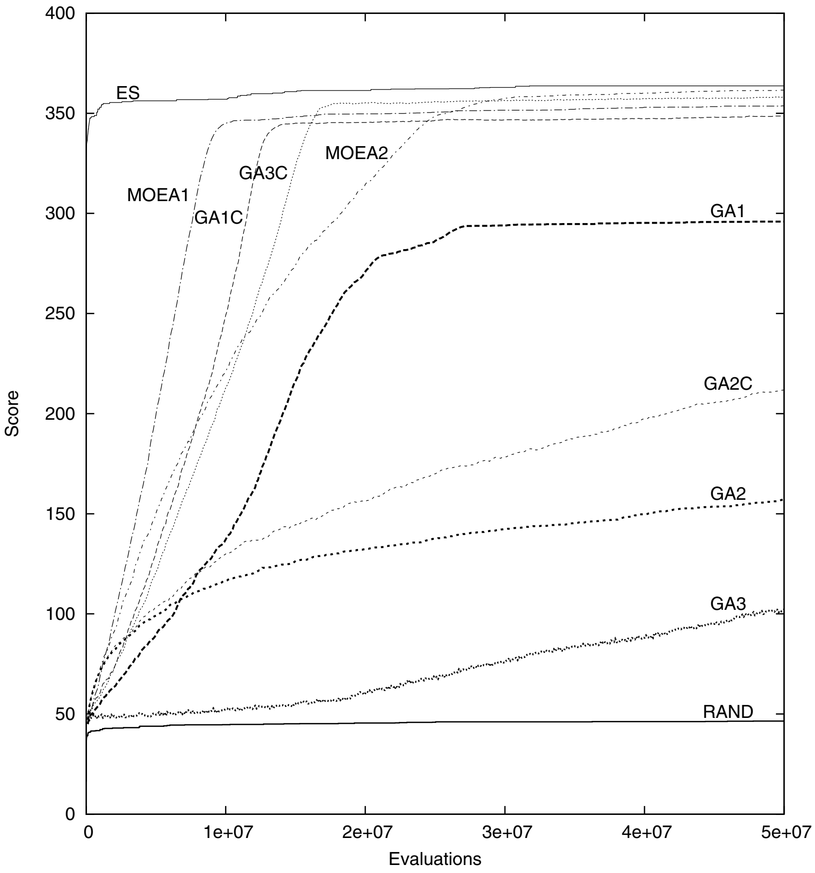


Fig. 4. Evolution of the score in the experiments

algorithm (GA) and a multiobjective evolutionary algorithm (MOEA). Both evolutionary algorithms have been executed with different values for crossover, mutation and elitism, but the population has been 10000 individuals for all experiments and the tournament is always of size 3. The results of the experiments are shown in Table 1. First column (Experiment) shows the experiment name. Second column (Alg.) is the algorithm used in that experiment. Next three columns are the number of individuals that are generated for the next generation by crossover (Cross.), mutation (Mut.) and elitism (Elit.) respectively. The next column (Fitness) is the sort of fitness that have been used in the genetic algorithm: normal fitness (norm) or combined fitness (comb). In the next three columns are shown the minimum (Min.), maximum (Max.) and mean (Mean) value of the bests scores of the puzzles in the experiments. For these columns the higher value is the better is, with a maximum value of 480. The last column (Std. Dev.) is the standard deviation of the experiments. Empty cells ('-') mean that parameter is not used in the algorithm. Each experiment was executed 10 times and each time it was running until $5 \cdot 10^7$ evaluations, a evaluation means each time the fitness of an individual is obtained.

Fig. 4 shows the evolution of best average score of the experiments. The x-axis is the number of evaluations and the y-axis is the average best score. The name of the experiment is shown next to its line, just top-left of the line.

5 Conclusions and Future Works

The best result was obtained in the *MOEA2* experiment with a score of 371 over 480, but the best average score was obtained with the exhaustive search (*ES*), 363.7 over 480.

The genetic algorithm with the normal fitness got better results with a higher crossover rate and the experiment *GA1* got the best results among then. There also notice that the experiment without elitism (*GA3*) got very bad results, it looks like it was slower to converge. In the genetic algorithm with combined fitness the results were better than with normal fitness and converged faster. But unlike before, here the experiment without elitism (*GA3C*) got the best results and converged almost as fast as the experiment with the higest crossover rate (*GA1C*). The difference between use a normal fitness or combined one without elitism is incredible huge, the algorithm goes from a results near the random search (*RAND*) to results very close the better ones (*ES*). If we compare the experiments *GA2* and *GA2C*, where the only difference is that the second one uses a combined fitness, the results were better in *GA2C* but the standar deviation increased a lot.

Between the two multiobjective experiments, *MOEA1* converged faster than *MOEA2* but this second one got better results, closer to the exhaustive search.

In future works we will compare these results with more evolutionary algorithms like SAWEA or FlipGA and with more specific algorithms for CSPs like FC-CBJ.

Acknowledgment

This work was supported in part by the Carlos III University of Madrid under grant PIF UC3M01-0809 and by the Ministry of Science and Innovation under project TRA2007-67374-C02-02.

References

1. Tomy: Eternity II (official site) (November 2008), `http://www.eternityii.com`
2. Demaine, E., Demaine, M.: Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Complexity. Graphs and Combinatorics 23, 195–208 (2007)
3. Dechter, R.: Constraint Processing. Morgan Kaufmann, San Francisco (2003)
4. Raman, V., Ravikumar, B., Rao, S.S.: A simplified NP-complete MAXSAT problem. Information Processing Letters 65(1), 1–6 (1998)

5. Rana, S., Whitley, D.: Genetic algorithm behavior in the MAXSAT domain. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 785–794. Springer, Heidelberg (1998)
6. Kumar, V.: Algorithms for Constraint-Satisfaction Problems: A Survey. AI Magazine 13(1), 32–44 (1992)
7. Pierre Schaus, Y.D.: Hybridization of CP and VLNS for Eternity II. JFPC (2008)
8. Gottlieb, J., Marchiori, E., Rossi, C.: Evolutionary Algorithms for the Satisfiability Problem. Evolutionary Computation 10(1), 35–50 (2002)
9. Bäck, T., Eiben, A., Vink, M.: A Superior Evolutionary Algorithm for 3-SAT. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 125–136. Springer, Heidelberg (1998)
10. Eiben, A., van der Hauw, J.: Solving 3-SAT with adaptive Genetic Algorithms. In: Proceedings of the 4th IEEE Conference on Evolutionary Computation, pp. 81–86 (1997)
11. Gottlieb, J., Voss, N.: Improving the performance of evolutionary algorithms for the satisfiability problem by refining functions. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 755–764. Springer, Heidelberg (1998)
12. Gottlieb, J., Voss: Adaptive fitness functions for the satisfiability problem. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 621–630. Springer, Heidelberg (2000)
13. Marchiori, E., Rossi, C.: A flipping genetic algorithm for hard 3-SAT problems. In: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 1, pp. 459–465 (1999)
14. Rossi, C., Marchiori, E., Kok, J.N.: An adaptive evolutionary algorithm for the satisfiability problem. In: Proceedings of the 2000 ACM symposium on Applied computing, vol. 1, pp. 463–469 (2000)
15. Haralick, R.M., Elliott, G.L.: Increasing Tree Search Efficiency for Constraint Satisfaction Problems. Artificial Intelligence 14(3), 263–313 (1980)
16. Craenen, B., Eiben, A., van Hemert, J.: Comparing evolutionary algorithms on binary constraint satisfaction problems. IEEE Transactions on Evolutionary Computation 7(5), 424–444 (2003)
17. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester (2001)

Multiobjective Algorithms Hybridization to Optimize Broadcasting Parameters in Mobile Ad-Hoc Networks

Sandra García, Cristóbal Luque, Alejandro Cervantes, and Inés M. Galván

Computer Science Departament, Carlos III University of Madrid
Avda. Universidad 30, 28911 Leganes, Spain
{sgrodrig,cluque,acervant,igalvan}@inf.uc3m.es

Abstract. The aim os this paper is to study the hybridization of two multi-objective algorithms in the context of a real problem, the MANETs problem. The algorithms studied are Particle Swarm Optimization (MOPSO) and a new multiobjective algorithm based in the combination of NSGA-II with Evolution Strategies (ESN). This work analyzes the improvement produced by hybridization over the Pareto's fronts compared with the non-hybridized algorithms. The purpose of this work is to validate how hybridization of two evolutionary algorithms of different families may help to solve certain problems together in the context of MANETs problem. The hybridization used for this work consists on a sequential execution of the two algorithms and using the final population of the first algorithm as initial population of the second one.

1 Introduction

Nowadays, many real-world optimization problems require compromise between several objectives. Mathematically, this is modelled as trying to simultaneously optimize several fitness functions depending on the same set of parameters.

Whenever there is a chance of combination of these objectives in a single objective function, these problems can be solved using conventional single-objective optimization methods. However, quite often there is no easy, or proper, way to combine the problem's objectives; this happens whenever objectives are conflicting, that is, improving some of the objectives means decreasing some of the rest. In these cases we need to solve a Multiobjective Optimization problem. The solution to this kind of problems is known as the pareto-optimal solution [2].

Multiobjective Evolutionary Algorithms (MOEAs) have been proved to be specially suited for this kind of optimization task. Some of the better known are SPEA[3] , PAES [4] or NSGA-II [5]. Hybrid metaheuristics are currently a subject of great interest for the community. Some combinations of techniques have been quite successful, such as combination of Local Search (LS) heuristics (Hill-descent, Simulated Annealing, Tabu Search) or even combinations of a LS technique with Evolutionary-based techniques. There are also some tools that may help performing this kind of experiments: for instance, Paradiseo[6] or Team[7].

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 728–735, 2009.
© Springer-Verlag Berlin Heidelberg 2009

In this paper we consider the problem of broadcasting on a particular sub-class of MANETs called Metropolitan MANETs (Mobile Ad-hoc Networks). This problem has been already studied using Multiobjective Evolutionary Algorithms[8].

The purpose of this work is to study how hybridization of two evolutionary algorithms of different families may help to improve the results obtained by individual algorithm for the MANETs problem. The evolutionary multi-objective algorithms combined in this work are MOPSO (Multi-Objective Particle Swarm Optimization) and ESN (Evolution Stratety with NSGAII) algorithms[9]. The first one is an adapted version of Particle Swarm Optimization to multi-objective optimization problems. And the second one is a multi-objective version for the Evolutionary Strategy algorithm.

2 MANETs Problem Definition

According to IETF (Internet Engineering Task Force), MANETs are autoconfiguration structures where *stations* (also called *terminals*) are temporally connected without a pre-existing infrastructure or a centralized administration [10]. MANETs are wireless networks where one station can communicate directly with the other stations.

In the last years, MANETs have increased its interest, as path to the 3G networks: a) they allow to extend the coverage, b) they don't need new infrastructure. In this way, a terminal user can use another users' terminals as part of a multistep path toward the kernel of the network.

In this paper we consider the problem of broadcasting on a particular sub-class of MANETs called Metropolitan MANETs, which span from shopping malls to metropolitan areas. Instead of providing a generic set of parameters that performs well on average situations, the goal is to optimize the broadcast messages. Optimizing a broadcasting strategy is a multiobjective problem where multiple functions have to be satisfied at the same time.

For this work, the broadcasting strategy considered for optimization is DFCN [11]. The DFCN strategy defines five parameters (minGain, lowerBoundRAD, upperBoundRAD, proD and safeDensity) to determine the following three objective values:

- Minimizing the makespan, or time required to perform the broadcast.
- Maximizing the network coverage, or number of stations reached.
- Minimizing the bandwidth used for broadcast messages and retries.

These three objectives define the objective space, and are considered three independent fitness functions. The purpose of optimization is to tune the input parameters of DFCN in order to find the three-dimensional Pareto-optimal front defined in objective space. The front is made by all the feasible non-dominated values for the three objective functions.

3 Multi-Objective Algorithms

The hybridization studied in this work consists on just sequentially executing the two algorithms and using the output population of the first algorithm to initialize the population of the second one. Thus, one algorithm is executed during a number of generations and, afterwards, the other algorithm is executed using as initial population the result obtained by the first one. This mix of populations requires certain considerations because individuals of the algorithms use different information. For MOPSO, individuals represent a particle with a position in the swarm and a velocity. However, ESN works with individuals that have assigned a variance value, which is updated during the evolution and it is used to generate new individuals. Thus, depending on the order in which the algorithms are executed, the following considerations are taken into account:

- MOPSO+ESN: When the population obtained by MOPSO is used to initialize ESN, the position of particles are the individuals for ESN and the variance value for all individuals in the population is fixed to 0.1.
- ESN+MOPSO: When the first algorithm used is ESN and the population obtained after a number of generations is used to initialize the swarm, the position of particles is initialized using the solutions of ESN and the velocity is determined randomly from the interval [-1.0,1.0]

3.1 Multi-Objective Particle Swarm Optimization: MOPSO

MOPSO [1] is an adapted version of Particle Swarm Optimization (PSO) to multi-objective optimization problems. MOPSO combines PSO [12] with the archiving strategy of PAES [4]. In this work, we use the version available in the EMOO repository [13].

The algorithm uses an external repository that stores non-dominated solutions in the swarm. Each iteration, for each particle, a leader is selected. The mechanism of leader selection is as follows: the fitness space is divided in hypercubic sectors, and one of the positions in the repository is randomly selected using a roulette algorithm that favors the sectors that are less populated. The velocity and position of particles are updated using the standard equations of PSO but using the leader particle selected from the repository. The fitness of the new position is calculated. This new position is compared with solutions already in the repository, and is inserted if it is a non-dominated solution. The maximum number of particles in the repository is fixed, so when this limit is reached, before the insertion, a particle from the most-populated sector of the repository is removed. If the new solution dominates some of the solutions in the repository, those solutions are removed.

3.2 Evolution Stratety with NSGAII: ESN

The ESN algorithm is based on the hybridization of Evolution Strategies and NSGAII. The algorithm uses the standard Evolution Strategies' steps [14], replacing the selection process by the NSGAII [5] selection process. In our implementation of Evolution Strategies, each of the μ individuals in the population produces one offspring only by mutation. From the resulting population of $2 \cdot \mu$ individuals, the best μ are selected for the next generation.

In the reproduction phase, for each individual the variance value is updated. Each parent generates one child using the new variance and a Gaussian distribution. After reproduction, the population is sorted using the NSGAII mechanism. This process creates the front F_1 with all the non-dominated solutions in the population. Then, those individuals are discarded and new fronts are created using the rest of the population. The worst μ individuals are deleted using the following criterion: an individual is better than another when it belongs to a front with a lower index; if two individuals are in the same front, the one with the greater crowding distance is better.

3.3 Results Evaluation: Metrics for Multiobjective Problems

Solutions resulting from different executions of the multiobjective algorithms must be compared using quantitative metrics that measure the success of the algorithms toward the MO problem objectives: distance to the “true” Pareto-optimal front, distribution of solutions over the obtained front, and spread.

It is generally admitted that there is no single metric that can be used to evaluate those objectives simultaneously; this is specially true when the best Pareto-optimal front is not known. A detailed description of available metrics can be found in [15].

We have chosen the following metrics to compare solutions (fronts) obtained by the algorithms:

- **Set Coverage (SC).** Coverage of a set of points A over a set of points B ($SC(A, B)$) is defined as the fraction of the points in set B that are weakly dominated by a point in set A. This measure has to be calculated in both directions (A vs. B and B vs. A) because $SC(A, B) \neq 1 - SC(B, A)$.
- **Hypervolume (HV).** This metric, also called size of the dominated space, is the volume enclosed by the union of the of the points in the set. The volume dominated by any point is calculated as the volume of the hypercube defined by each point. The value of this metric is usually calculated after normalization of the points in the solutions.

Set Coverage is a measure of how a set dominates the other in terms of number of points. However, it does not take into account the actual distance between the points in both sets; that is, “how much” a point in a set dominates the rest. It requires two sets of points (two fronts) for comparison, so it cannot be used to assign a performance measure to a single front except if the “true” Pareto-optimal front is known and used as reference.

Hypervolume assigns a quantitative value to a given set (where greater values mean better performance) but it cannot be used to derive dominance of a set over the other and its actual value depends on the normalization used.

4 Experiments

In this section we present the utility of algorithm hybridization proposed in this article in order to solve the aforementioned problem, tuning of the parameteres of the DFCN broadcasting protocol for MANETs.

All the experiments are performed using *Madhoc* [16], a Metropolitan MANET simulator. It aims at providing a tool for simulating different level services based on distinct technologies on MANETs for different environments. The simulator provides several different models of network structure, of which we have used the Mall environment, that simulates a commercial shopping center.

4.1 Experimental Settings

Experiments have been performed using different combinations of the order of execution of the algorithms and the number of evaluations performed by each of the algorithms.

Moreover, we have also executed experiments using only MOPSO and ESN, and we have calculated their metrics in order to compare their results with the other obtained by the hybridizations. The obtained fronts have been evaluated using the metrics explained in section 3.3 and their results are shown in tables 1,2,3 and 4.

All of these experiments used a total of 25000 evaluations. The population size for both algorithms was 100 individuals/particles. The initial variance for ESN was 0.1, and the value of the constant Δ was set to 0.7.

MOPSO used a swarm size of 100 particles, and fixed PSO parameters: $w = 0.4$, $r_1 = r_2 = 2.0$, $X = 0.4$. We used 30 divisions of the adaptive grid as suggested in [1]. The number of non-dominated points in the solution was limited to 100.

In tables presented in this section, ESN-X + MOPSO-Y means an experiment of X evaluations of the ESN algorithm, followed by Y evaluations of the MOPSO algorithm (and thus $X + Y = 25000$). A generation of 100 individuals takes 100 evaluations (one for each indivivual). So, X evaluations represents $X/100$ generations. In the same way for MOPSO, Y evaluations represents $Y/100$ iterations.

For each configuration, five experiments were performed. Tables in this section shown the mean value of the metrics for those experiments.

4.2 Experimental Results

Table 1 presents the Hypervolume for the different combinations of algorithms. As we can see, all the solutions have very similar values for this metric. The best result is attained by ESN-25000, but it is not much better than the other results. Thus, we can conclude that this metric is not enough to compare the quality of the hybrid algorithms, so we need to study the Set Coverage.

Set Coverage is studied in tables 2, 3 and 4. Table 2 shows the dominance that each hybridization obtain over the solutions that the MOPSO and ESN non-hybridized algorithms produces for this problem. Table 3 shows the dominance

Table 1. Hypervolume

Algorithm	Hypervolume	Variance
ESN-8000 + MOPSO-17000	0.7123	0.0000
ESN-12500 + MOPSO-12500	0.7119	0.0000
ESN-17000 + MOPSO-8000	0.7108	0.0000
MOPSO-8000 + ESN-17000	0.7143	0.0000
MOPSO-12500 + ESN-12500	0.7116	0.0000
MOPSO-17000 + ESN-8000	0.7080	0.0000
MOPSO-25000	0.7080	0.0000
ESN-25000	0.7195	0.0002

Table 2. Set Coverage

Set Coverage	MOPSO-25000	Variance	ESN-25000	Variance
ESN-8000 + MOPSO-17000	0.4267	0.0082	0.3461	0.0044
ESN-12500 + MOPSO-12500	0.4444	0.0132	0.3576	0.0062
ESN-17000 + MOPSO-8000	0.4420	0.0104	0.3774	0.0057
MOPSO-8000 + ESN-17000	0.4351	0.0087	0.3614	0.0083
MOPSO-12500 + ESN-12500	0.4168	0.0081	0.3362	0.0030
MOPSO-17000 + ESN-8000	0.4034	0.0094	0.3052	0.0063
MOPSO-25000	-	-	0.3548	0.0092
ESN-25000	0.4581	0.0153	-	-

Table 3. Set Coverage

Set Coverage	ESN-8000 + MOPSO-17000	Variance	ESN-12500 + MOPSO-12500	Variance	ESN-17000 + MOPSO-8000	Variance
MOPSO-25000	0.3552	0.0124	0.3436	0.0093	0.3368	0.0101
ESN-25000	0.4464	0.0115	0.4332	0.0102	0.4348	0.0152

Table 4. Set Coverage

Set Coverage	MOPSO-8000 + ESN-17000	Variance	MOPSO-12500 + ESN-12500	Variance	MOPSO-17000 + ESN-8000	Variance
MOPSO-25000	0.3370	0.0065	0.3636	0.0095	0.3708	0.0124
ESN-25000	0.4726	0.0196	0.4608	0.0130	0.5164	0.0145

of the non-hybridized algorithms over the ESN+MOPSO hybridized algorithms, and table 4 over the MOPSO+ESN algorithms.

Next, we analyze the Set Coverage obtained by the execution using first Evolution Strategy with NSGA-II as a first part and Particle Swarm Optimization as a second part of the hybridization. If we compare the percent of dominant points of these first three approaches versus MOPSO, we can see how these hybridizations improve the Set Coverage obtained by the solutions of "MOPSO-25000". But this does not happen with “ESN-25000” because their Pareto fronts are better (they have a greater Set Coverage) than the solutions obtained by the hybrid algorithm, as can be seen in tables 3 and 4. In conclusion, we cannot say that the hybridization in which ESN starts the execution improves the solutions obtained by the executions of the non-hybridized algorithms.

Continuing with the Set Coverage analysis, the experiments which use MOPSO as the first algorithm and ESN as the second one, we can see that the more ESN evaluations we perform, the better Set Coverage it obtains over the non-hybridized ESN. This also happens when we compare the hybridized experiments with non-hybridized MOPSO: it means that using more ESN evaluations produces better

Set Coverage over non-hybridized MOPSO. We think this is due to fact that the ESN algorithm has better performance in this kind of problem and helps MOPSO to find better fronts.

For the experiments with hybridized algorithms, we usually find better fronts in terms of Set Coverage when ESN is used first; and when MOPSO is used first, the results are better as long as MOPSO has less evaluations.

5 Conclusions

A mobile ad-hoc network (MANETs) is defined as a self-configuring network of mobile routers (and associated hosts). Due to the fact that network topology varies dynamically, the operations of broadcasting are of utter importance for the existence and the performance of the network and can benefit from parameter optimization on the broadcasting algorithm used in the network.

We have defined the optimization problem for the broadcasting strategy in MANETs as a multiobjective problem where a single solution is defined as a set of values for those parameters, and the objectives that must be simultaneously achieved are: Maximum Coverage, Minimum Bandwith usage and Minimum makespan. Experiments are conducted by simulation of the broadcasting protocol in a predefined network environment.

This paper applies the hybridization of two different MOEAs (MOPSO and ESN) to find the Pareto-optimal front, that is, the set of non-dominated solutions for this problem. In order to study the benefits of the hybridization, experiments using the non-hybridized algorithms, MOPSO and ESN, were also carried out. All experiments were compared using two standard metrics for evaluation of the solutions to multiobjective problems: Set Coverage and Hypervolume.

Results show that both algorithms and their different combinations are able to find non-dominated fronts that approximate the “true” Pareto-optimal front with adequate accuracy and distribution of the solutions.

When comparing the hybridizations, in terms of Hypervolume we cannot decide what performance is the best one, because its results are very similar. In terms of Set Coverage, we have proved that hybridization is a good tool to improve the results of MOPSO, but ESN without any hybridization obtains better fronts than other combinations of MOPSO and ESN.

Acknowledgments. This article has been financed by the Spanish founded research MEC projects OPLINK::UC3M, Ref:TIN2005-08818- C04-02 and MSTAR::UC3M, Ref:TIN2008-06491-C04-03.

References

1. Coello, C.A., Toscano, G., Salazar, M.: Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(3), 256–279 (2004)
2. Goldberg, D.: Genetic algorithms in Sarch, Optimization, and Machine Learnig. Addison Wesley Publishing Company, Reading (1989)

3. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization, EUROGEN, pp. 95–100 (2001)
4. Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the pareto archived evolution strategy. Evol. Comput. 8(2), 149–172 (2000)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
6. Talbi, E., Cahon, S., Melab, N.: ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics. Journal of Heuristics, 357–380 (2004)
7. León, C., Miranda, G., Segura, C.: A Parallel Plugin-Based Framework for Multiobjective Optimization. In: International Symposium on Distributed Computing and Artificial Intelligence, vol. 50/2009, pp. 142–151 (2008)
8. Alba, E., Dorronsoro, B., Luna, F., Nebro, A.J., Bouvry, P., Hogie, L.: A Cellular Multi-Objective Genetic Algorithm for Optimal Broadcasting Strategy in Metropolitan MANETs. Computer Communications 30(4), 685–697 (2007)
9. Pérez Pérez, R., Luque, C., Cervantes, A., Isasi, P.: Multiobjective Algorithms to Optimize Broadcasting Parameters in Mobile Ad-hoc Networks. In: IEEE Congress on Evolutionary Computation, pp. 3142–3149 (2007)
10. IETF Secretariat: IEFT MANET Charter (2007), `http://www.ietf.org/html.charters/manet-charter.html`
11. Hogie, L., Guinand, F., Bouvry, P.: A Heuristic for Efficient Broadcasting in the Metropolitan Ad Hoc Network. In: 8th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems, pp. 727–733 (2004)
12. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
13. Coello, C.A., et al.: EMOO Repository, `http://www.lania.mx/ccoello/EMOO`
14. Bäck, T., Schwefel, H.P.: Evolutionary Algorithms: Some Very Old Strategies for Optimization and Adaptation. New Computing Techniques in Physics Research II. In: Proc. Second Int'l Workshop Software Engineering, Artificial Intelligence, and Expert Systems for High Energy and Nuclear Physics, pp. 247–254 (1992)
15. Zitzler, E., Thiele, L.: An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach, Technical Report 43, Gloriastrasse 35, CH- 8092 Zurich, Switzerland, `http://citeseer.ist.psu.edu/article/zitzler99evolutionary.html` (1998)
16. Hogie, L., Guinand, F., Bouvry, P.: The madhoc metropolitan adhoc network simulator. Technical report, Université du Luxembourg and Université du Havre, France (March 2006), `http://wwwlih.univ-lehavre.fr/hogie/madhoc/`

Application Synthesis for MPSoCs Implementation Using Multiobjective Optimization

Marcus Vinícius Carvalho da Silva[1], Nadia Nedjah[1], and Luiza de Macedo Mourelle[2]

[1] Department of Electronics Engineering and Telecommunication
[2] Department of System Engineering and Computation,
Engineering Faculty, State University of Rio de Janeiro
`{marcus,nadia,ldmm}@eng.uerj.br`

Abstract. Network-on-chip (NoC) are considered the next generation of communication infrastructure for a multiprocessors system-on-chip (MPSoCs). In the platform-based methodology, an application is implemented by a set of collaborating intellectual properties (IPs) blocks. In this paper, we use NSGA-II and microGA to yield efficient topological pre-selected sets IPs on the tiles of a mesh-based NoC. Each IP is associated with a processor that best implements its functionality. The IP mapping optimization is driven by the area occupied, execution time and power consumption.

1 Introduction

Normally, a NoC is designed to establish the communication between the processes of an application parallel implementation as a MPSoCs. The application, usually, consists of a limited number of tasks that are implemented by a set of IP blocks. Different applications may have a similar, or even the same, set of tasks. An IP block can implement more than a single task of the application. For instance, a processor IP block can execute many tasks as a general processor does but a multiplier IP block for floating point numbers can only multiply floating point numbers. The number of IP blocks designers, as well as the number of available IP blocks, is growing up fast.

In order to yield an efficient NoC-based design for a given application, it is necessary to choose the adequate minimal set of IP blocks. With the increase of IP blocks available, this task is becoming harder and harder. Besides IP blocks carefully assignment, it is also necessary to map the blocks onto the NoC available infra-structure, which consists of a set of *cores* communicating through *switches*. A bad mapping can degrade the NoC performance. Different optimization criteria can be pursued depending on how much information details is available about the application and IP blocks.

Usually, the application is viewed as a graph of tasks called *task graph* (TG). The IP blocks features can be obtained from their companion documentation. The IP assignment and IP mapping are key research problems for efficient NoC-based designs. These two problems are *NP*-hard problems and can be solved using multi-objective optimizations.

In this paper, we propose a multi-objective evolutionary-based decision support system to help NoC designers. For this purpose, we propose a structured representation of

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 736–743, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the TG and an IP repository that will feed data into the system. We used two MOEAs: NSGA-II [2] and microGA [1]. Both of these algorithms were modified according to some prescribed NoC design constraints.

The problem of mapping IP blocks into a NoC physical structure have been addressed in some previous studies [6] [7] [4]. However, some of these works did not take into account of the multi-objective nature of these problems and adopted a single objective optimization approach.

A NoC platform consisting of architecture and design methodology, which scales from a few dozens to several hundreds or even thousands of resources. As mentioned before, a resource may be a processor core, DSP core, an FPGA block, a dedicated hardware block, mixed signal block, memory block of any kind such as RAM, ROM or CAM or even a combination of these blocks. A NoC consists of set of *resources* and *switches*. Resources and switches are connected by *links*. The pair (R, S) forms a *tile*. The simplest way to connect the available resources and switches is arranging them as a mesh so these are able to communicate with each other by sending messages via an available path. A switch is able to buffer and route messages between resources. Each switch is connected to up to four other neighboring switches through input and output channels. While a channel is sending data another channel can buffer incoming data.

The rest of the paper is organized as follows: First, in Section 2, we describe a structured TG and IP repository model based on the E3S data. After that, in Section 3, we introduce the mapping problem in NoC-based environments. Then, in Section 4, we sketch the two MOEAs used in this work, individual representations and objective functions for the optimization stage. Later, in Section 6, we show some experimental result. Last but not least, in Section 7, we draw some conclusions and outline new directions for future work.

2 Task Graph and IP Repository Models

In order to formulate the IP mapping problem, it is necessary to introduce a formal definition of an application first. An application can be viewed as a set of tasks that can be executed sequentially or in parallel. It can be represented by a directed graph of tasks, called *task graph*. A *Task Graph* (TG) $G = G(T, D)$ is a directed graph where each node represents a computational module in the application referred to as task $t_i \in T$. Each directed arc $d_{i,j} \in D$, between tasks t_i and t_j, characterizes either data or control dependencies. Each task t_i is annotated with relevant information, such as a unique identifier and type of processing element (PE) in the network. Each $d_{i,j}$ is associated with a value $V(d_{i,j})$, which represents the volume of bits exchanged during the communication between tasks t_i and t_j. Once the IP assignment has been completed, each task is associated with an IP identifier. The result of this stage is a graph of IPs representing the PEs responsible of executing the application.

An *Application Characterization Graph* (APG) $G = G(C, A)$ is a directed graph, where each vertex $c_i \in C$ represents a selected IP/core and each directed arc $a_{i,j}$ characterizes the communication process from core c_i to core c_j. Each $a_{i,j}$ can be tagged with IP/application specific information, such as communication rate, communication bandwidth or a weight representing communication cost. A TG is based on application features only while an APG is based on application and IP features, providing us with a much more realistic representation of the an application in runtime on a NoC. In order to be able to bind application and IP features, at least one common feature is required in both of the IP and TG models.

The E3S (0.9) Benchmark Suite [3] contains the characteristics of 17 embedded processors. These processors are characterized by the measured execution times of 47 different type of tasks, power consumption derived from processor data-sheets, and additional information, such as die size, price, clock frequency and power consumption during idle state. In addition, E3S contains task graphs of common tasks in auto-industry, networking, telecommunication and office automation. Each one of the nodes of these task graphs is associated with a task type. A task type is a processor instruction or a set of instructions, e.g., FFT, inverse FFT, floating point operation, OSPF/Dijkstra, etc. If a given processor is able to execute a given type of instruction, so that processor is a candidate to receive a resource in the NoC structure and would be responsible for the execution of one or more tasks. Here, we represent TGs using XML code. A TG is divided in three major elements: *taskgraph*, *nodes* and *edges*. Each *node* has two main attributes: an unique identifier (*id*) and a task type (*type*), chosen among the 47 different types of tasks present in the E3S. Each *edge* has four main attributes: an unique identifier (*id*), the *id* of its source node (*src*), the *id* of its target node (*tgt*) and an attribute representing the communication cost imposed (*cost*).

The IP repository is divided into two major elements: the *repository* and the *ips* elements. The *repository* is the IP repository itself. Recall that the repository contains different non general purpose embedded processors and each processor implements up to 47 different types of operations. Not all 47 different types of operations are available in all processors. Each type of operation available in each processor is represented by an *ip* element. Each *ip* is identified by a unique identifier and other attributes such as task type, processor identifier, power consumption, execution time and hardware area.

3 The IP Mapping Problem

The platform-based design methodology for SoC encourages the reuse of components to increase re-usability and to reduce the time-to-market of new designs. The designer of NoC-based systems faces two main problems: selecting the adequate set of IPs that optimize the execution of a given application and finding the best physical mapping of these IPs into the NoC structure.

The main objective of the IP assignment stage is to select, from the IP repository, a set of IPs that minimize the NoC consumption of power, area occupied and execution time. At this stage, no information about physical allocation of IPs is available so optimization must be done based on TG and IP information only. So, the result of this step is the set of IPs that maximizes the NoC performance. The TG is then annotated and an APG is produced, wherein each node has an IP associated with it.

In this paper, given an application, described by its APG, the problem that we are concerned with is determining how to topologically map the selected IPs onto the MPSoS structure, such that the objectives of interest are optimized. Some of these objectives are: latency requirements, power consumption of communication, total area occupied and thermal behavior.

4 Multi-Objective Evolutionary Approach

The search space for a ”good” IP mapping for a given application is defined by the possible combinations of IP/tile available in the NoC structure. Assuming that the mesh-based NoC structure has $N \times N$ titles and there are at most N^2 IPs to map, we have a domain size of $N^2!$. Among the huge number of solutions, it is possible to find many equally good solutions. In huge non-continuous search space, deterministic approaches do not deal very well with MOPs. The domination concept introduced by Pareto is necessary to classify solutions. In order to deal with such a big search space and trade-offs offered by different solutions in a reasonable time, a multi-objective evolutionary approach is adopted.

The core of the proposed tool offers the utilization of two well-known and well-tested MOEAs: NSGA-II [2] and microGA [1]. Both adopt the domination concept with a ranking schema for classification. The ranking process separates solutions in *Pareto fronts* where each front corresponds to a given rank. Solutions from rank *one*, which is the *Pareto-optimal* front) are equally good and better than any other solution from Pareto fronts of higher ranks.

4.1 Representation and Genetic Operators

The individual representation includes, for each gene, the task identifier, the processor identifier and the tile indicates location on which a gene is mapped. On a $N \times N$ regular mesh, the tiles are numbered successively from top-left to bottom-right, row by row. The row of the i^{th} tile is given by $\lceil i/N \rceil$, and the corresponding column by i mod N. The crossover and mutation operators were adapted to the fact that the set of selected IPs can not be changed as we have to adhere to the set of prescribed IP assignments. For this purpose, we propose a crossover operator that acts like a shift register, shifting the genes around a random crossover point and thus generating a new solution, but with the same set of IPs. This behavior does not contrast with the biological inspiration of evolutionary algorithms, observing that certain species can reproduce through parthenogenesis, a process in which only one individual is necessary to generate an offspring. The mutation operator performs an *inner swap mutation*, where each gene receives a random mutation probability, which is compared against the system mutation probability. The genes with mutation probability higher than the system’s are swapped with another random gene of the same individual, instead of selecting a random IP from the repository. This way, it is possible to explore the allocation space preserving any optimization done in the IP assignment stage.

5 Objective Function

During the evolutionary process, the fitness of the individuals with respect to each one of the selected objectives (i.e. *area*, *time*, and *power*) must be efficiently computed.

In order to compute the area required by a given mapping, it is necessary to know the area needed for the selected processors and that required by the used links and switches. As a processor can be responsible for more than one task, each APG node must be visited in order to check the processor identification in the appropriate XML element. Grouping the nodes with the same *processorID* attribute allows us to implement this verification. The total number of links and switches can be obtained through the consideration of all communication paths between exploited tiles. Note that a given IP mapping may not use all the available tiles, links and switches. Also, observe that a portion of a path may be re-used in several communication paths. In this work, we adopted a fixed route strategy wherein data emanating from tile i is sent first horizontally to the left or right side of the corresponding switch, depending on the target tile position, say j, with respect to i in the NoC mesh, until it reaches the column of tile j, then, it is sent up or down, also depending on the position of tile j with respect to tile i until it reaches the row of the target tile. Each communication path between tiles is stored in the routing table. The number of links in the aforementioned route can be computed as described in Equation 1. This is also represents the distance between tiles i and j and called the *Manhattan distance*.

$$nLinks(i,j) = |\lceil i/N \rceil - \lceil j/N \rceil| + |i \bmod N - j \bmod N| \tag{1}$$

In the purpose of computing efficiently the area required by all used links and switches, an APG can be associated with a so-called *routing table* whose entries describe appropriately the links and switches necessary to reach a tile from another. The number of hops between tiles along a given path leads to the number of links between those tiles, and incrementing that number by 1 yields the number of traversed switches. The area is computed summing up the areas required by the implementation of all distinct processors, switches and links.

Equation 2 describes the computation involved to obtain the total area for the implementation a given IP mapping M, wherein function $Proc(.)$ provides the set of distinct processors used in APG$_M$ and $area_p$ is the required area for processor p, function $Links(.)$ gives the number of distinct links used in APG$_M$, A_l is the area of any given link and A_s is the area of any given switch.

$$Area(M) = \sum_{p \in Proc(APG_M)} area_p + (A_l + A_s) \times Links(APG_M) + A_s \tag{2}$$

To compute the execution time of a given mapping, we consider the execution time of each task of the critical path, their schedule and the additional time due to data transportation through links and switches along the communication path. The critical path can be found visiting all nodes of all possible paths in the task graph and recording the largest execution time of the so-called critical path. The execution time of each task is defined by the *taskTime* attribute in TG. Links and switches can be counted using the routing table. We identified three situations that can degrade the implementation performance, increasing the execution time of the application:

1. *Parallel tasks mapped into the same tile:* A TG can be viewed as a sequence of horizontal levels, wherein tasks of the same level may be executed in parallel, allowing

for a reduction of the overall execution time. When parallel tasks are assigned in the same processor, which also means that these occupy the same tile of the NoC, they cannot be executed in parallel.

2. *Parallel tasks with partially shared communication path:* When a task in a tile must send data to supposedly parallel tasks in different tiles through the same *initial* link, data to both tiles cannot be sent at the same time.
3. *Parallel tasks with common target using the same communication path:* When several tasks need to send data to a common target task, one or more shared links along the partially shared path would be needed simultaneously. The data from both tasks must then be pipelined and so will not arrive at the same time to the target task.

Equation 3 is computed using a recursive function that implements a depth-first search, wherein function $Paths(.)$ provides all possible paths of a given APG and tm_t is the required time for task *t*. After finding the including the total execution time of the tasks that are traversed by the critical path, the time of parallel tasks executed in the same processor need to be accumulated too. This is done by function *SameProcSameLevel*(.). The delay due to data pipelining for tasks on the same level is added by *SameSourceCommonPath*(.). Last but not least, the delay due to pipelining data that are emanating at the same time from several distinct tasks yet for the same target task is accounted for by function *DiffSrcSameTgt*(.).

$$Time(M) = \max_{r \in Paths(TG_M)} \left(\sum_{t \in r} tm_t + f_1(r) + f_2(r) + f_3 \right) \tag{3}$$

Function f_1 compares tasks of a given same level that are implemented by the same processor and returns the additional delay introduced in the execution of those tasks. Function f_2 computes the additional time due to parallel tasks that have data dependencies on tasks mapped in the same source tile and yet these share a common initial link in the communication path. Function $f_3$computes the additional time due to the fact that parallel tasks producing data for the same target task need to use simultaneously at least a common link along the communication path.

The total power consumption of an application NoC-based implementation consists of the power consumption of the processors while processing the computation performed by each IP and that due to the data transportation between the tiles. The former can be computed summing up attribute *taskPower* of all nodes of the APG and the latter is the power consumption due to communication between the application tasks through links and switches. The power consumption due to the computational activity is simply obtained summing up atribute *taskPower* of all nodes in the APG and is as described in Equation 4.

$$Power_p(M) = \sum_{t \in APG_M} power_t \tag{4}$$

An energy model for one bit consumption is used to compute the total energy consumption for the whole communication involved during the execution of an application on the NoC platform. The bit energy (E_{bit}), can be obtained as in Equation 5:

$$E_{bit} = E_{S_{bit}} + E_{L_{bit}} \tag{5}$$

wherein $E_{S_{bit}}$ and $E_{L_{bit}}$ represent the energy consumed by the switch and link tying the two neighboring tiles, respectively.

The total power consumption of sending one bit of data from tile i to tile j can be calculated considering the number of switches and links the bit passes through on its way along the path, as shown in Equation 6.

$$E_{bit}^{i,j} = nLinks(i,j) \times E_{L_{bit}} + (nLinks(i,j) + 1) \times E_{S_{bit}} \tag{6}$$

wherein function $nLinks(.)$ provides the number of traversed links (and switches too) considering the routing strategy used in this work and described earlier in this section. The function is is defined in Equation 1.

Recall that the application TG gives the communication volume ($V(t,t')$) in terms of number of bits sent from the task t to task t' passing through a direct arc $d_{t,t'}$. Assuming that the tasks t and t' have been mapped onto tiles i and j respectively, the communication volume of bits between tiles i and j is then $V(i,j) = V(d_{t,t'})$. The communication between tiles i and j may consist of a single link $l_{i,j}$ or by a sequence of $m > 1$ links $l_{i,x_0}, l_{x_0,x_1}, l_{x_1,x_2}, \ldots, l_{x_{m-1},j}$. The total network communication power consumption for a given mapping M is given in Equation 7, wherein $Targets_t$ provides all tasks that have a direct dependency on data resulted from task t and $Tile_t$ yields the tile number into which task t is mapped.

$$Power_c(M) = \sum_{t \in APG_M, \forall t' \in Targets_t} V(d_{t,t'}) \times E_{bit}^{Tile_t, Tile_{t'}} \tag{7}$$

6 Results

Many simulations were performed to find out the setting up of the parameters used in NSGA-II and micro-GA. For the former, we used a population size of 600, mutation probability of 0.01, crossover probability of 0.8 and tournament size of 50 and run the algorithm of 100 generations. For the latter, we used population memory size of 1000,

Table 1. Performance results obtained by both NSGA-II for the image processing application

		microGA	NSGA-II
# solutions		3.538	1.992
# Tiles	Minimal	5	6
	Maximal	14	14
	Average	9,5	11,3
Averages	Power$^{(1)}$	10,6461	7,4509
	Area$^{(2)}$	32,6802	30,3599
	Time$^{(3)}$	122.170,04	3.614,3
Minimal	Power$^{(1)}$	3,8250	3,8250
	Area$^{(2)}$	11,3484	14,5684
	Time$^{(3)}$	594,8192	589,6462

^{1}(W), ^{2}($\times 10^{-6}$m^2), ^{3}($\times 10^{-3}$s), ^{4}(hh:mm:ss,ms).

replaceable fraction of 0.7, non-replaceable fraction of 0.3, micro population of 4 individuals, mutation probability of 0.02, crossover probability of 0.09, tournament size of 2, external memory of 200, nominal convergence of 4, replacement cycle of 100, bisection of 5, and run the algorithm for 3000 generations. We also adopted a limit ofa maximum of 2 tasks per processor. The performance of the multi-objective evolutionary processes performed by NSGA-II and microGA were tested using the application provided in the E3S benchmark suite. However, for space limitation, we report the performance results for only one application used for image processing [5]. The task graph of the application for 4 segments has 14 tasks. The obtained results by both algorithms are exposed in Table 1.

7 Conclusions

In this paper, we propose a decision support system based on MOEAs to help NoC designers allocate a prescribed set of IPs into a NoC physical structure. The use of two different MOEAs consolidates the obtained results. Structured and intelligible representations of a NoC, a TG and of a repository of IPs were used and these can be easily extended to different NoC applications. Despite of the fact that we have adopted E3S Benchmark Suite [3] as our repository of IPs, any other repository could be used and modeled using XML, making this tool compatible with different repositories. The proposed *shift crossover* and *inner swap mutation* genetic operators can be used in any optimization problem where no lost of data from a individual is accepted. Future work can be two-fold: adopting a dynamic topology strategy to try to evolve the most adequate topology for a given application and exploring the use of different objectives based on different repositories.

References

1. Coello Coello, C.A., Toscano Pulido, G.: A micro-genetic algorithm for multiobjective optimization. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 126–138. Springer, Heidelberg (2001)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE-EC 6, 182–197 (2002)
3. Dick, R.P., Rhodes, D.L., Wolf, W.: TGFF: Task Graphs For Free. In: Proceedings of the 6th International Workshop on Hardware/Software Co-design, Seattle, Washington, USA, March 1998, pp. 97–101. IEEE Computer Society, Los Alamitos (1998)
4. Jena, R.K., Sharma, G.K.: A multi-objective evolutionary algorithm based optimization model for network-on-chip synthesis. In: ITNG, pp. 977–982. IEEE Computer Society, Los Alamitos (2007)
5. Marcon, C.A.M.: Modelos para o Mapeamento de Aplicações em Infra-Estruturas de Comunicação Intrachip. Ph.d., Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil (2005)
6. Murali, S., De Micheli, G.: Bandwidth-constrained mapping of cores onto NoC architectures. In: DATE, pp. 896–903. IEEE Computer Society, Los Alamitos (2004)
7. Zhou, W., Zhang, Y., Mao, Z.: Pareto based multi-objective mapping IP cores onto NoC architectures. In: APCCAS, pp. 331–334. IEEE, Los Alamitos (2006)

Multi Objective Optimization Algorithm Based on Neural Networks Inversion

Sara Carcangiu, Alessandra Fanni, and Augusto Montisci

Electrical and Electronic Engineering Dept., University of Cagliari
Piazza d'Armi 1, 09123 Cagliari, Italy
`{s.carcangiu,fanni,amontisci}@diee.unica.it`

Abstract. A new neural network–based multiobjective optimization approach is presented, which performs an approximation of the direct problem by means of a neural network, and solves the inverse problem inverting the neural network itself, namely by imposing the value of the desired objective functions and by searching the corresponding value of the design parameters. The search for the Pareto front can be performed directly in the objectives space, rather than in the design parameters, allowing both to uniformly sample the Pareto front, and to limit the computational load. Inverting a neural network corresponds to find the intersection of non convex domains. The proposed inversion algorithm allows to exploit the algorithms available for linear domains, by iteratively evaluating linear approximations of non linear domains, increasing the convergence property. To demonstrate the procedure and the performance of the neural network–based approach, the problem of optimal configuration of an electromagnetic device is selected for analysis and discussion.

Keywords: Multiobjective Optimization, Neural Network Inversion, Neural Approximation Model.

1 Introduction

A Multi Objectives optimization Problem (MOP) is characterized by a vector of mutually conflicting objective functions and can be formalized as follows:

$$\min_{\underline{x}\in\Omega} F(\underline{x}); \quad \Omega = \left\{ \underline{x} \in R^n \middle| G(\underline{x}) \le 0; H(\underline{x}) = 0 \right\} \tag{1}$$

where $\underline{x}$ is the vector of the input variables (design variables), and F is the vector of objective functions ($F_1,\ldots,F_K$). Here, $G(\underline{x})$ and $H(\underline{x})$ represent the problem constraints. A solution $\underline{x}_p$ is a Pareto optimal solution if no objective function can be improved without worsening at least another objective function. Such solution is not unique, and the set of the Pareto optimal solutions are know as the Pareto front.

During the last years, several approaches have been proposed to solve the MOPs, most of which belonging to the class of stochastic methods [1-6]. Despite of the significant progress obtained by the researchers, there are still many open problems to be solved, mainly due to the time consuming numerical procedures (e.g., Finite Elements

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 744–751, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Methods - FEM) needed to evaluate the objective functions, especially for real world design problems.

A different approach consists of applying the optimization procedure, rather than directly to the numerically evaluated objective functions, to their approximation models, and in particular to Neural Networks (NN) models. The inputs of the NN models are the design parameters and the corresponding outputs are the objective functions to be optimized. These models can then be used in place of numerical analysis during optimization [5, 6]. Alternatively, prefixed values of the objective functions can be imposed and the corresponding design parameters can be found by inverting the neural models [7-9].

In this paper, an original approach to search for the Pareto front directly in the objectives space will be presented, which is based on the inversion of the neural network models. As will be detailed in the following, the problem of inverting a NN corresponds to find the intersection of two non convex domains. Several papers in literature deal with projection algorithms for finding intersection points between convex and non convex sets [7, 10-12]. Nevertheless, the proposed projecting algorithms do not assure the convergence. In this paper a new algorithm is presented, which increases robustness and limits the computational cost. Moreover, it extends the capability of the algorithm in [7] to handle multi objectives, rather than mono objective optimization problems.

In order to evaluate the capability of the proposed approach, the method has been used to find the optimal design of a magnetic pole.

2 Neural Network Inversion Algorithm

The proposed approach consists of inverting a MultiLayer Perceptron (MLP) previously trained to approximate the solutions of the direct problem. The standard MLP neural network architecture is adopted, with a single hidden layer. The aim of our procedure is to determine the inputs that correspond to a prefixed output. The inversion of the neural network consists of finding a solution of the non linear neural network equations system [7]:

$$a)\underline{\underline{W}}_1 \cdot \underline{x} + \underline{b}_1 = \underline{y} \quad ; \quad b)\underline{\underline{W}}_2 \cdot \underline{h} + \underline{b}_2 = \underline{u} \quad ; \quad c)\underline{h} = \sigma(\underline{y}) \tag{2}$$

where $\underline{x}$ is the input of the neural network, $\sigma(\cdot)$ is the hidden neurons sigmoidal activation function, $\underline{u}$ is the output, $\underline{y}$ and $\underline{h}$ are the input and the output of the hidden layer respectively (see Fig. 1). The activation function of the output layer is linear. The coefficients of the equations (2.a) and (2.b) are equal to the connections weights $\underline{\underline{W}}_1$ and $\underline{\underline{W}}_2$, and the biases $\underline{b}_1$ and $\underline{b}_2$ of the neural network.

Since non linear equations may have multiple solutions, it seems that neural networks have a built-in non-invertible character [13]. Nevertheless, if the target $\underline{u}^*$ of the network can be specified with a prefixed tolerance $\underline{\varepsilon}$, an iterative procedure can be developed that is able to find (if it exists) a solution whose value differs from the specified target less than the imposed tolerance. If this solution does not exist, the tolerance $\underline{\varepsilon}$ can be relaxed until a solution is found.

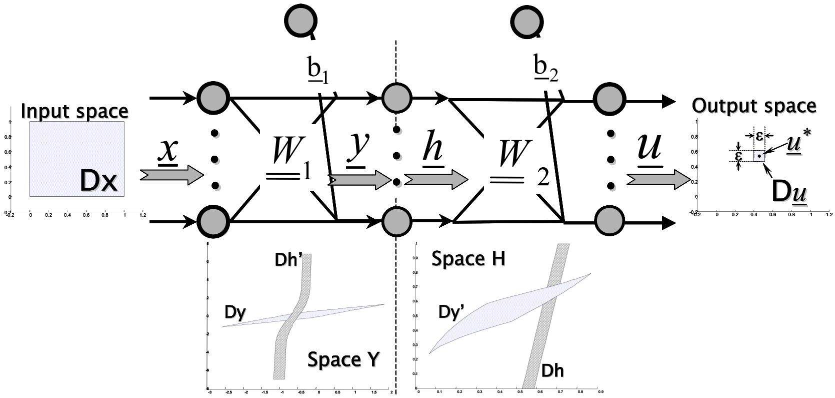


Fig. 1. MLP network structure: D_x is the input domain, D_u is the output domain, D_y and D_h are the domains of $\underline{y}$ and $\underline{h}$ respectively

By means of equation (2.a), the input domain D_x can be linearly projected in the space **Y** where the vector $\underline{y}$ is defined, obtaining the domain D_y.

This means that, in order to match the constraints of the input, the vector $\underline{y}$ has to belong to D_y. On the other hand, by means of equation (2.b), the output domain D_u can be linearly projected in the space **H** where the vector $\underline{h}$ is defined, obtaining the domain D_h. In order to match the constraints of the output, the vector $\underline{h}$ has to belong to D_h. The domain D_h takes into account both the domain D_u and the bounds of the hidden layer activation function. All of these constraints are linear, so the domain D_h can be expressed as linear inequalities.

The equation (2.c) states a biunivocal relationship between $\underline{y}$ and $\underline{h}$, so that the domain D_h can be back-propagated in the space **Y** throughout the hidden layer obtaining the domain D'_h which is no longer linear due to the nonlinear activation function of the hidden layer. A point, which belongs to the intersection between the two domains D_y and D'_h, matches at the same time the constraints of the input and the constraints of the output, and then it is a solution of the inversion problem.

2.1 Projection Algorithm

In the present section, an iterative procedure is proposed whose goal is to find a point that belongs to the intersection I_{yh} between the nonlinear domain D'_h and the linear domain D_y.

Starting from a point $\underline{y}_0$ external to a linear domain, it is easy to project it on the linear domain, following the shortest linear path between the point and the domain. Projecting is more difficult in the case of nonlinear domains, because the shortest path cannot be derived on the basis of the normal to the domain facets. In fact, whereas the normal of a linear facet has the same direction in the whole space, this is not true for a nonlinear facet.

In the present case, projecting a point on the nonlinear domain D'_h is a difficult task, whereas no difficult would exist if the projection was performed in the space **H**, where the domain is linear.

In this paper, in order to exploit the algorithms available for linear domains, the generic nonlinear facet of the domain D'_h:

$$\underline{w}^T \cdot \sigma(\underline{y}) \leq c$$

is approximated with a hyperplane by substituting the sigmoidal functions with their first-order approximation:

$$\underline{w}^T \cdot \left[\sigma(\underline{y}_0) + \sigma'(\underline{y}_0) \cdot (\underline{y} - \underline{y}_0)\right] \leq c \qquad (3)$$

where $\sigma'(.)$ is the first order derivative of the sigmoid. By means of (3) the nonlinear domain D'_h is substituted by the linear domain L_h and $\underline{y}_0$ is projected on the intersection of two linear domains. As the goodness of such projection depends on the precision of the approximation (3), which deteriorates when the distance between the starting and the projected points increases, a new linear approximation of the domain D'_h has to be iteratively evaluated, and a new projection performed, starting from the projected point, until the projection falls within I_{yh}. Fig. **2** shows the iterative procedure for a neural network with a 2-2-1 structure.

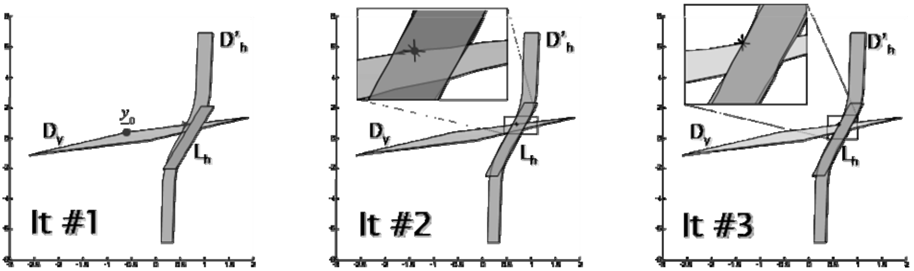


Fig. 2. Procedure for intersection search

Starting from a generic point $\underline{y}_0$ of the space **Y**, e.g., a point belonging to D_y, firstly a first-order approximation of D'_h is calculated, taking the point $\underline{y}_0$ as reference point. The domain D'_h is substituted with the domain L_h and $\underline{y}_0$ is projected on the intersection between D_y and L_h. If such point belongs to I_{yh}, the procedure ends, otherwise the obtained point is assumed as starting point in the successive iteration. The intersection between D_y and L_h, could be empty, but this not imply that a solution of the inversion problem does not exist. In this case, the point of D_y nearest to L_h is taken as starting point at the successive iteration.

Remark. In the example the first iteration already allows to reach the intersection I_{yh}, even if the projected point does not lies in the frontier of both the domains D_y and D'_h. This is due to the fact that in Fig. **2** D_y and D'_h have the same dimension of the space **Y** where they are defined, so one expects that the frontier the intersection I_{yh} be part of the frontiers of the two domains D_y and D'_h. On the contrary, in the general case the hidden layer is greater than the input layer, therefore the dimension of the domain D_y is less than that of the space **Y**. On the other hand the desired output could be defined

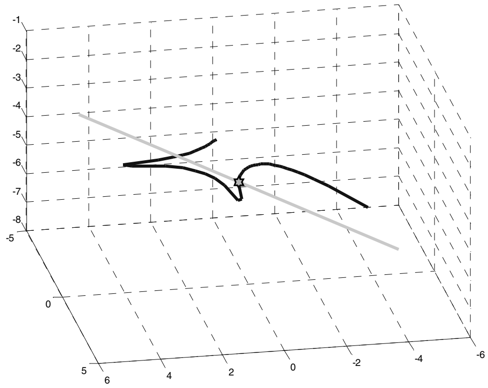


Fig. 3. Intersection between the domains D_y and D'_h for a 1-3-2 MLP network

without tolerance, and this implies that also D'_h could have dimension smaller than the space. In Fig. 3, an example of domains D_y and D'_h of a 1-3-2 MLP network is shown. As it can be noted, the intersection, which consists of only one point, does not fall in the frontiers of the two domains.

3 Using the Inversion Algorithm in the Multiobjective Optimization

The performance of a MOP solver can be evaluated on the basis of different criteria: capability of finding Pareto optimal solutions; capability of uniformly sampling the Pareto front; limited computational cost.

Many strategies have been presented in literature, and in general they consist of defining a criterion to explore the parameters space and to reach the Pareto front. Many independent runs have to be performed in order to obtain a good sampling of the front, and the samples can be scarcely equally-spaced, because the search is performed in the parameters space (Input space) rather than in the objective functions space. Moreover, for each set of design parameters, the corresponding values of the objectives have to be calculated. Usually, each objectives vector evaluation involves a heavy computational load, so that many strategies become unsuitable when the number of objectives is large.

Using a neural network to describe the relationship between parameters and objective functions allows one to obtain two advantages: reducing the computational cost of function calls, which is the main obstacle to use some strategies; exploiting the neural model in order to search the Pareto points directly in the objective space rather than in the parameter space. This last is possible by resorting to the inversion procedure described in Section 2. In Fig. 4, the proposed method is illustrated for a problem with two objectives to maximize. Starting from a feasibly point in the objective space, one can move toward the utopia point until an unfeasible solution is reached, which means that the Pareto front has been reached. Starting from this first Pareto point (0) one can start to sample the Pareto front.

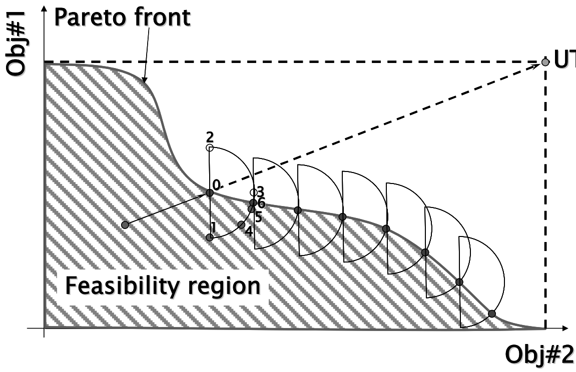


Fig. 4. Sampling of the Pareto front

Firstly, the desired sampling step of the front is set. Then, a circle is ideally drawn with centre placed on the first Pareto point (0) and radius equal to the sampling step. Two points (1) and (2) can be identified on the opposite sides of the circle: one in the feasibility region (1), and one in the non-feasibility region (2). By means of, e.g., a bisection method, the intersection between the circle and the Pareto front (point 6) is reached. The new Pareto point becomes the centre of a new circle and the procedure is iterated until the whole front is reconstructed.

Once the Pareto front is available, the final decision is taken *a posteriori* by the decision maker on the basis of a fitness criterion.

4 Results

In order to show the performance of the proposed optimization algorithm, the method has been used to find the optimal configuration of an electromagnetic device test bed [4]. The benchmark consists in the optimal shape design of a magnetic pole. In Fig. 5, the model of the device is shown. Because of the symmetry with respect to the x-axis, only a half of the magnetic pole rectilinear section has been modeled. The current density is uniform in the winding and is zero elsewhere. The non-linear permeability of the ferromagnetic material is taken into account. As far as the inverse problem is concerned, four design variables $\underline{x} = (x_1, x_2, x_3, x_4)$ (see Fig. 5) are selected.

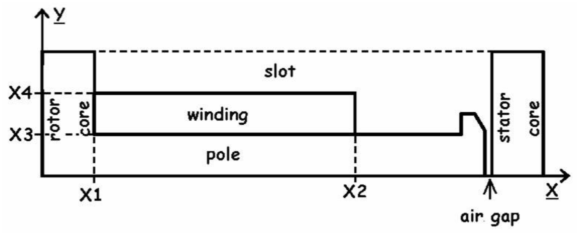


Fig. 5. Model of the magnetic pole

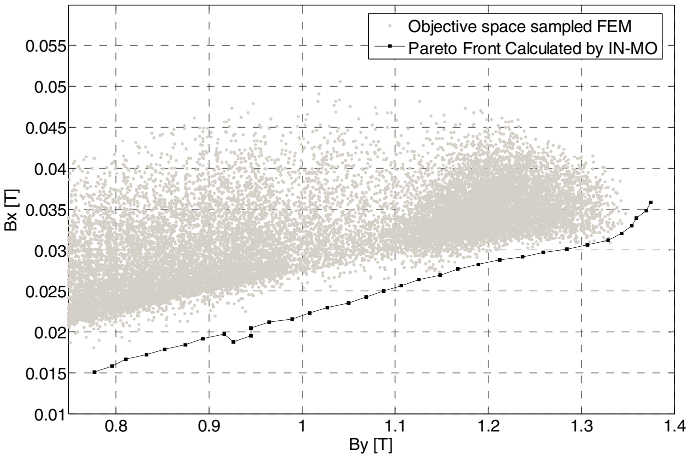


Fig. 6. Magnetic pole: approximated Pareto front

The feasible region of the design variables is defined by the conditions on both geometric congruency and non-saturation of the material.
Two objective functions are defined:

- $F_1(\underline{x})$ is the maximum component of magnetic induction in the x-axis direction along the air gap midline, to be maximized;
- $F_2(\underline{x})$ is the average component of magnetic induction in the y-axis direction in the winding, to be minimized.

A MLP is trained to solve the direct problem and to calculate the two objective functions. The NN model has 4 input neurons, corresponding to the design variables, 40 neurons in the hidden layer, and 2 output neurons, corresponding to the objective functions. The MLP associates the objective function values to each feasible vector $\underline{x}$ by means of a set of examples, which have been randomly sampled in the design parameters space. These training examples are calculated by means of FEM simulations. Fig. 6 shows a random sampling of the objectives space and the Pareto front with 34 sample points obtained by means of the proposed procedure. It has to be highlighted that, the approximation of the Pareto front mostly depends on the accuracy of the neural approximation model, which is known a priori. This is a considerable advantage, especially in engineering problems.

In Table 1, the comparison of the performance of different algorithms, taken from literature, are reported. All the algorithms well fit the Pareto front and, except for

Table 1. Comparison of the performance of different algorithms

	TS-EC [6]	MOTS [6]	MOESTRA[5]	Present algorithm
# iterations	60.000.000	70.000.000	9316	800
CPU Time [sec]	660	900	1107	150

MOESTRA algorithm, the respective samples are uniformly distributed. As can be noted the proposed method over performs the other algorithms in terms of computational cost.

5 Conclusions

A neural network inversion algorithm and its application to a multi objective optimization problem are presented in this paper. The neural model of the relationship among design parameters and objectives has to be firstly crated. Then, inverting the model, it is possible to solve the synthesis problem, i.e., to establish the design parameters that correspond to prefixed values of the objectives. This procedure has been exploited to solve an electromagnetic MOP, performing the search of the Pareto front directly in the objectives space rather than in the parameters space. The procedure showed good performance both in terms of computational cost and quality of the obtained solutions.

References

1. Zitler, E., Deb, K., Thiele, L.: Comparison of Multiobjective evolutionary algorithms: empirical results. Evolutionary Computation 8, 173–195 (2000)
2. Ho, S.L., Yang, S., Ni, G., Lo, E.W.C., Wang, H.C.: A Particle Swarm optimization based method for multiobjective design optimizations. IEEE Trans. on Mag. 41(5), 1756–1759 (2005)
3. Vasconcelos, J.A., Maciel, J.H.R.D., Parreiras, R.O.: Scatter Search Techniques applied to Electromagnetics problems. IEEE Trans. on Mag. 41(5), 1804–1807 (2005)
4. Di Barba, P., Mognaschi, M.E.: Recent experiences of multiobjective optimization in electromagnetics: A comparison of methods. COMPEL 24(3), 921–930 (2005)
5. Carcangiu, S., Di Barba, P., Fanni, A., Mognaschi, M.E., Montisci, A.: Comparison of multi-objective optimisation approaches for inverse magnestostatic problems. COMPEL 26(2), 293–305 (2007)
6. Carcangiu, S., Fanni, A., Montisci, A.: Multiobjective Tabu Search Algorithms for Optimal Design of Electromagnetic Devices. IEEE Trans. on Mag. 44(6), 970–973 (2008)
7. Cherubini, D., Fanni, A., Montisci, A., Testoni, P.: Inversion of MLP Neural Networks for Direct Solution of Inverse Problems. IEEE Trans. on MAG. 41(5), 1784–1787 (2005)
8. Cau, F., Di Mauro, M., Fanni, A., Montisci, A., Testoni, P.: A Neural Networks Inversion–Based Algorithm for Multiobjective Design of a High-Field Superconducting Dipole Magnet. IEEE Transactions on Magnetics 43(4), 1557–1560 (2007)
9. Carcangiu, S., Fanni, A., Montisci, A.: A Multiobjective Strategy-Driven Optimization Procedure for the Design of Electromagnetic Devices. In: Proceedings of the 13th IEEE conference on electromagnetic field computation (2008)
10. Elser, V., Rankenburg, I., Thibault, P.: Searching with iterated maps. Proceedings of the National Academy of Sciences 104(2), 418–423 (2006)
11. Bauschke, H.H., Combettes, P.L., Luke, D.R.: Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization. Journal of the Optical Society of America A 19(7), 1334–1345 (2002)
12. Fienup, J.R.: Phase retrieval algorithms: a comparison. Applied Optics 21, 2758–2769 (1982)
13. Rico Martinez, R., Adomaitis, R.A., Kevrekidis, I.G.: Noninvertibility in neural networks. Computers & Chemical Engineering 24, 2417–2433 (2000)

EMORBFN: An Evolutionary Multiobjetive Optimization Algorithm for RBFN Design

Pedro L. López, Antonio J. Rivera, M. Dolores Pérez-Godoy, María J. del Jesus,
and Cristóbal Carmona

Department of Computer Science, University of Jaén. Jaén (Spain)
pllc0001@estudiantes.ujaen.es,
{arivera,lperez,mjjesus,ccarmona}@ujaen.es

Abstract. In this paper a multiobjective optimization algorithm for the design of Radial Basis Function Networks is proposed. The goal of the design algorithm is to obtain networks with a high tradeoff between accuracy and complexity, overcoming the drawbacks of the traditional single objective evolutionary algorithms. The main features of EMORBFN are a selection mechanism based on NSGA-II and specialized operators. To test the behavior of EMORBFN a similar mono-objective optimization algorithm for Radial Basis Function Network design has been developed. Also C4.5, a Multilayer Perceptron network or an incremental method to design of Radial Basis Function Networks have been included in the comparison. Experimental results on six UCI datasets show that EMORBFN obtains networks with high accuracy and low complexity, outperforming other more mature methods.

Keywords: Evolutionary Multi-objective Optimization, Radial Basis Function Networks, Classification.

1 Introduction

Radial Basis Function Networks (RBFNs) are one of the most important Artificial Neural Network (ANN) paradigms in the machine learning design field. An RBFN is a feed-forward ANN with a single layer of hidden units, called radial basis functions (RBFs). The first research on neural networks based on RBFs [4] was carried out at the end of the eighties. Since then, the overall efficiency of RBFNs has been proved in many areas like pattern classification [5], function approximation [13] and time series prediction [20].

The main features of a RBFN are a simple topological structure, with only one hidden layer, of neurons/RBFs that have a particular locally-tuned response that depends on the center and the width (radius) of each RBF. Also, they have universal approximation capability [13].

The goal of any RBFN design process is to determine centers, widths and the linear output weights connecting the RBFs to the output neuron layer. The most traditional learning procedure has two stages: first, unsupervised learning of centers and widths is used, and finally output weights are established by means of supervised learning. Clustering techniques [14] are commonly used to adjust the centers. Regarding the widths, they may all be given the same value, may reflect the width of the previously

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 752–759, 2009.
© Springer-Verlag Berlin Heidelberg 2009

calculated clusters (i.e., RBFs), or may be established as the average distance between RBFs, among other possibilities. In order to obtain the weights in the second stage, algorithms such as Least Mean Square (LMS) [21] can be used.

Another important paradigm concerning RBFN design is Evolutionary Computation (EC) [3][10], a general stochastic optimization framework inspired by natural evolution. Typically, in this paradigm each individual represents a whole network that is evolved in order to increase its accuracy.

Managing only the accuracy objective to optimize RBFNs, like in traditional evolutionary computation, may lead to obtain nets with a high number of RBFs. This is because it is easier to reduce error with those RBFNs having many RBFs than with those having few neurons. In order to overcome this drawback, Evolutionary Multiobjective Optimization (EMO)[6], can be used. Among the different multiobjective methods, Nondominated Sorting Genetic Algorithm NSGA-II [7] is one of the most efficient and used techniques.

In this paper, a new evolutionary multiobjective optimization algorithm for the design of RBFNs (EMORBFN) applied to classification problems is presented. This algorithm follows the basic lines of a standard evolutionary algorithm but uses a NSGAII-based strategy to select the population of the new generation. In order to promote diversity, new operators to recombine and mutate individuals of the population are introduced. Other important characteristic of EMORBFN is the use of a dissimilarity measure (HVDM)[22] to calculate distances between nominal attributes. To test the performance of the proposed algorithm a traditional monoobjective evolutionary algorithm for the design of RBFNs (EORBFN) has been developed. Also, other well-know classification algorithms such as an RBFN-incremental algorithm, C4.5 or a Multilayer Perceptron Network have been used to compare the results of our proposed method.

The rest of paper is organized as follows: in section 2, a revision of the multiobjective optimization of RBFNs is shown. The EMORBFN algorithm is explained in section 3. The experimentation and the analysis of the results are shown in section 4. Conclusions are outlined in section 5.

2 Evolutionary Design of RBFNs

An RBFN is a feed-forward neural network with three layers: an input layer with n nodes, a hidden layer with m neurons or RBFs, and an output layer with one or several nodes (Figure 1). The m neurons of the hidden layer are activated by a radially-symmetric basis function, $\phi_i : R^n \rightarrow R$, which can be defined in several ways.

From all the possible choices for ϕ_i, the Gaussian function is the most widely used: $\phi_i(\vec{x}) = \phi_i(e^{-(\|\vec{x}-\vec{c}_i\|/d_i)^2})$, where $\vec{c}_i \in R^n$ is the center of basis function ϕ_i, $d_i \in R$ is the width (radius), and $\| \;\|$ is usually the Euclidean norm on R^n. The output of one basis function will be high when the input vector and the center of this basis function are close, always taking into account the value of the radius. The weights w_{ij} show the contribution of an RBF to the respective output node, and therefore output nodes implement the weighted sum of RBF outputs (equation (1)).

$$f_j(\vec{x}) = \sum_{i=1}^{m} w_{ij}\phi_i(\vec{x}) \tag{1}$$

Nowadays, Evolutionary Computation [3] is one of the most important paradigms in the design of RBFN [10][5]. Typically, in these algorithms, the individuals are codified using the Pittsburgh scheme, where an individual represents a whole RBFN. In the other hand, a cooperative-competitive evolutionary approach represents an RBF in each individual [16].

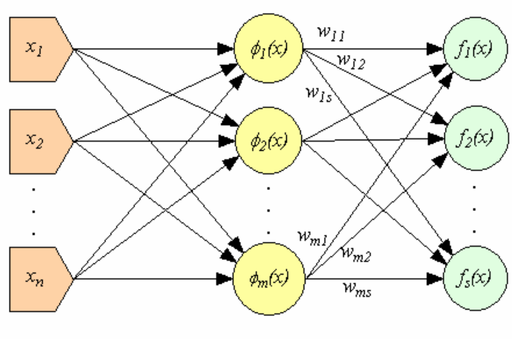


Fig. 1. RBFN Topology

Most of Pittsburgh methods only optimize the accuracy of the individuals/RBFNs without having into account the complexity of the networks obtained. As have been said, managing only the objective of the accuracy may lead to obtaining nets with a high number of RBFs. That's because it is easier to get less error in RBFNs with many RBFs than in RBFNs with few neurons. Evolutionary Multiobjective Optimization [6] can be used in order to optimize several objectives such as the accuracy and complexity (the number of RBFs) of the models.

Standard Multiobjective Evolutionary Algorithms (MOEAs) have been used in the design of RBFNs for estimation or function approximation, such as Multiobjective Genetic Algorithms MOGAs in [8] or [19]. [17] hybrids NSGA-II with rough-sets and surrogate techniques in order to achieve a more efficient multiobjective algorithm. [11] uses spatially distributed surrogates (RBFNs) in a NSGA-II framework. In [12] simple RBFNs (i.e.: with the same width for all RBFNs), obtained by a NSGA-II algorithm, are used as ensembles in the final model proportioned by the algorithm. In a more specific application regarding estimations, [9] uses the NSGA-II in the mineral reduction environment. Also, there are specific MOEAs such as [23] that present a Hierarchical Rank Density Genetic Algorithm (HRDGA) to evolve the neural network's topology and parameters simultaneously.

3 EMORBFN: Evolutionary Multiobjetive Optimization Algorithm for RBFN Design

Basically, EMORBFN is an Evolutionary Multiobjective algorithm for the design of RBFN. It follows the main steps shown in Fig. 2. In step 6, in order to create the next population, the NSGA-II algorithm is used. The objectives taken into account to optimize and sort the individuals are their accuracy and complexity (number or neurons). The implemented codification is real and Pittsburgh-based, where each individual represents a whole network. A dissimilarity measure HVDM distance [22] has been chosen in order to work with nominal attributes. Bellow, the main steps of EMORBFN are explained.

```
1. $P_t$ = Initialization (Data)
2. Training/Evaluation ($P_t$)
3. $R_t$ = Recombination ($P_t$)
4. $M_t$ = Mutation ($P_t$)
5. Training/Evaluation ($R_t$, $M_t$)
6. $P_{t+1}$ = Fill_next_population ($P_t$, $R_t$, $M_t$)
7. If the stop-condition is not verified go to step 2
```

Fig. 2. Main steps of EMORBFN

RBFN initialization.- For each initial network, a random number, m, of neurons (i.e. the size of population) is also randomly allocated among the different classes of the training set. Each RBF center, $\vec{c}_i$, is randomly established to a pattern of the training set, taking into account that the RBFs must be distributed equally among the different classes. The RBF widths, d_i, will be set to half of the average distance between the centers. Finally, the RBF weights, w_{ij} , are set to zero.

Training/evaluation.- During this stage, the well-known LMS algorithm [21] has been used to calculate the RBF weights of each individual. At this moment, the RBFN is evaluated, obtaining its classification error.

Recombination.- With the crossover operator two individuals/RBFNs parents are randomly chosen to obtain one RBFN offspring. The number of RBFs of the new individual will be delimited between a minimum and a maximum value. The minimum value is set to the number of RBFs of the parent with fewer RBFs. The maximum value is set to sum of the number of RBFs of both parents. In order to generate the offspring, RBFs will be chosen in a random way from the parents.

Mutation.- Six mutation operators, usually considered in the specialised bibliography [10] have been implemented. They can be classified as random operators or biased operators. The random operators are:

- *DelRandRBFs*: randomly eliminates a *pm* percent of the total number of RBFs in the RBFN.
- *InsRandRBFs*: randomly adds a *pm* percent of the total number of RBFs in the RBFN.
- *ModCentRBFs*: randomly modifies the center of *pm* percent of the total number of RBFs in the RBFN. The center of the basis function will be modified in a *pr* percent of its width.

- *ModWidtRBFs*: randomly modifies a *pm* percent of the total number of RBFs in the RBFN. The width of the basis function will be modified in a *pr* percent of its width.

Biased operators, which exploit local information, are:

- *DelInfRBFs*: deletes a *pm* percent of the total number of RBFs in the RBFN.
- *InsInfRBFs*: inserts a *pm* percent of the total number of RBFs in the RBFN.

Filling the next population.- In order to create the new population, the basic lines of NSGA-II have been followed. In this way the rank of parents P_t and offspring (Q_t, R_t) is calculated depending on the values of their objectives by means of a dominance count. Next, the all non-dominated fronts $F = \{F_1, F_2, \ldots\}$ are generated, where individuals in front F_1 have less ranking value that individuals in F_2 and so on. The new population is filled inserting the fronts F_1, F_2, … until the maximum number of the individuals of the population is reached. If the number of the individuals of the last front (F_l) is greater than the number of individuals needed to fill P_{t+1}, then crowding distance is calculated for the individuals of F_l. Thus, individuals with large crowding measure will complete P_{t+1}.

4 Experimentation and Results

To test our multiobjective method against a monoobjective based proposal, a typical Evolutionary algorithm for the Optimization of RBFNs, EORBFN has been developed. The design lines of EORBFN are the classical ones for this kind of algorithms [10]. In order to promote an adequate comparison between EMORBFN and EORBBFN, similar operating conditions have been established. In this way EORBFN and EMORBFN have the same pseudocode, codification scheme (Pittsburgh), operators and dissimilarity measure (HVDM). Obviously, the main differences can be found in step 6 when the new population is created. To fill the new population P_{t+1} from the populations of parents (P_t) and offspring (Q_t, R_t) a tournament selection mechanism is considered. The diversity of the population is promoted by using a low value for the tournament size ($k = 3$). The objective considered in this evolutionary algorithm is to minimize the classification error.

In order to increase the efficiency of the evolutionary algorithms, the search space of this method has been restricted. In this experimentation, the search space has been reduced by fixing the complexity (number of RBFs) between a minimum and a maximum number of neurons. The minimum number of neurons has been set to the number of classes for the problem and the maximum to four times this number. The parameter values used for experimentation with EMORBFN and EORBFN are shown in Table 1.

EMORBFN has been compared with other traditional classification methods such as: C4.5 [15], an algorithm that creates classification rules in the form of decision trees from a dataset. MLP-Back [18], an algorithm for Multilayer Perceptron Networks design which uses the Backpropagation algorithm for learning. RBFN-Incr [1], an algorithm for the RBFNs design based on an incremental scheme. The implementation of these algorithms has been obtained from KEEL [1].

Table 1. Experimentations parameters used for EMORBFN and EORBFN

Parameter	Value
Iterations of the main loop	200
Individuals	40
Crossover probability	0.6
Mutation probability	0.1
Pm	0.2
Pr	0.2

Table 2. Results with Hepatitis dataset

Algorithm	*#nodes/ Rules*	*Classification rate (%)*
C4.5	7.1	**89.647**
MLP-Back	30.0	71.817
RBFN-Incr	120.1	76.637
EORBFN	7.5	86.711
EMORBFN	**4.0**	87.598

Table 3. Results with Iris dataset

Algorithm	*#nodes/ rules*	*Classification rate (%)*
C4.5	4.8	94.000
MLP-Back	30.0	64.667
RBFN-Incr	29.8	94.667
EORBFN	10.4	95.067
EMORBFN	**4.0**	**95.789**

Table 4. Results with Pima dataset

Algorithm	*#nodes/ Rules*	*Classification rate (%)*
C4.5	18.3	73.972
MLP-Back	30.0	70.819
RBFN-Incr	671.4	67.728
EORBFN	7.6	75.615
EMORBFN	**5.0**	**75.683**

Table 5. Results with Sonar dataset

Algorithm	*#nodes/ rules*	*Classification rate (%)*
C4.5	14.3	71.071
MLP-Back	30.0	67.762
RBFN-Incr	159.8	74.976
EORBFN	**7.7**	73.305
EMORBFN	8.0	**75.536**

Table 6. Results with Wbcd dataset

Algorithm	*#nodes/ rules*	*Classification rate (%)*
C4.5	12.4	94.995
MLP-Back	30.0	87.722
RBFN-Incr	319.9	94.303
EORBFN	6.2	96.713
EMORBFN	**4.0**	**97.535**

Table 7. Results with Wine dataset

Algorithm	*#nodes/ rules*	*Classification rate (%)*
C4.5	5.1	94.902
MLP-Back	30.0	93.301
RBFN-Incr	125.3	74.739
EORBFN	10.4	95.275
EMORBFN	**3.0**	**95,708**

The collection of data sets used in this section (Hepatitis, Iris, Pima, Sonar, Wbcd, Wine) was obtained from the UCI Repository of Machine Learning Database [2]. The parameters used for C4.5, MLP-Back and RBFN-Incr are the ones proposed by the authors. In order to estimate the precision we use a ten-fold cross validation approach, that is, ten partitions for training and test sets, 90% for training and 10% for testing, where the ten test partitions form the whole set. For each dataset we consider the average results of the ten partitions.

From Table 2 to Table 7 the classification test rate of the methods and their corresponding complexities (number of nodes -RBFs- in the RBFN or number of the rules for C4.5) are shown. For the EMORBFN algorithm, the best accuracy network obtained is chosen.

From these results it can be inferred that EMORBFN is the best method in accuracy in five of the six datasets; only in Hepatitis dataset, C4.5 outperforms EMORBFN in only two points. Despite this fact, EMORBFN obtains the second best result in accuracy for this dataset. Regarding the complexity of the models obtained, EMORBFN achieves the model with the lowest complexity in five of the six datasets. Only EORBFN obtains a model with lower complexity in Sonar Dataset. But again, EMORBFN obtains the second best result in complexity for this dataset.

Only attending to the results obtained by EMORBFN and EORBFN can be observed that: the results of EORBFN regarding accuracy are similar to the EMORBFN results; but studying complexity, EORBFN obtains more complex models that EMORBFN. Also, EORBFN models often reach the maximum limit of RBFs, set in its parameters. This fact confirms the hypothesis that EMORBFN overcomes the drawbacks of EORBFN.

5 Conclusions

In this paper a new evolutionary multiobjective algorithm for RBFN design, EMORBFN, has been presented. EMORBFN implements the basic pseudocode of any evolutionary algorithm but, in order to fill the population of the next generation, a NSGAII-based technique is used. In order to promote diversity, new operators to recombine and mutate individuals of the population are introduced. Another important characteristic of EMORBFN is the use of a dissimilarity measure (HVDM) to calculate distances between nominal attributes.

To analyze the behavior of EMORBFN, a traditional evolutionary monoobjective optimization algorithm (EORBFN) for RBFN design has been developed. EORBFN has the same operating elements that EMORBFN but only optimizes the accuracy of the networks. Also, the results for other well-known classification methods, such as C4.5, MLP-BackPropagation and an incremental RBFN design method have been obtained and used in the comparison.

An analysis of the results shows that EMORBFN mostly outperforms the other proposals, both in accuracy and complexity. EORBFN achieves competitive accuracy results but the models obtained reach the maximum limit of RBFs set in the parameters. This fact confirms our initial hypothesis and demonstrates that EMORBFN overcomes the drawbacks of traditional evolutionary design algorithms.

Acknowledgement

This work has been partially supported by the CICYT Spanish Projects TIN2008-06681-C06-02 and TIN2007-60587.

References

1. Alcalá-Fdez, J., Sánchez, L., García, S., Del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F.: KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. Soft Comput. (2008), doi:10.1007/s00500-008-0323-y
2. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, Irvine (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

3. Bäck, T., Hammel, U., Schwefel, H.: Evolutionary computation: comments on the history and current state. IEEE T. Evolut. Comput. 1, 3–17 (1997)
4. Broomhead, D., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex System 2, 321–355 (1988)
5. Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. IEEE T. Syst. Man Cy. B 35(5), 928–947 (2005)
6. Deb, K.: Multi-objective optimization using evolutionary algorithms, 1st edn. Wiley, Chichester (2001)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation 6(2), 182–197 (2002)
8. González, J., Rojas, I., Ortega, J., Pomares, H., Fernández, F.J., Díaz, A.F.: Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. IEEE T. Neural Networ. 14(6), 1478–1495 (2003)
9. Guillén, A., Pomares, H., Rojas, I., González, J., Herrera, L.J., Rojas, F., Valenzuela, O.: Output value-based initialization for radial basis function neural networks. Neural Process Lett. (2007), doi:10.1007/s11063-007-9039-8
10. Harpham, C., Dawson, C., Brown, M.: A review of genetic algorithms applied to training radial basis function networks. Neural Comput. Appl. 13, 193–201 (2004)
11. Isaacs, A., Ray, T., Smith, W.: An evolutionary algorithm with spatially distributed surrogates for multiobjective optimization. In: Randall, M., Abbass, H.A., Wiles, J. (eds.) ACAL 2007. LNCS (LNAI), vol. 4828, pp. 257–268. Springer, Heidelberg (2007)
12. Kondo, N., Hatanaka, T., Uosaki, K.: Pattern classification via multi-objective evolutionary RBF networks ensemble 2006. SICE-ICASE art. no. 4108811, 137–142 (2006)
13. Park, J., Sandberg, I.: Universal approximation using radial-basis function networks. Neural Comput. 3, 246–257 (1991)
14. Pedrycz, W.: Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE T. Neural Networ. 9(4), 601–612 (1998)
15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kauffman, San Francisco (1993)
16. Rivera, A.J., Rojas, I., Ortega, J., del Jesus, M.J.: A new hybrid methodology for cooperative-coevolutionary optimization of radial basis function networks. Soft Comput. (2007), doi:10.1007/s00500-006-0128-9
17. Santana-Quintero, L.V., Coello, C.A., Hernández-Díaz, A.G.: Hybridizing surrogate techniques, rough sets and evolutionary algorithms to efficiently solve multi-objective optimization problems. In: GECCO 2008, pp. 763–764 (2007)
18. Rojas, R., Feldman, J.: Neural Networks: A Systematic Introduction. Springer, Heidelberg (1996)
19. Teixeira, C.A., Ruano, M.G., Ruano, A.E., Pereira, W.C.A.: A soft-computing methodology for non invasive time-spatial temperature estimation. IEEE T. Bio-Med. Eng. 55(2), 572–580 (2008)
20. Whitehead, B., Choate, T.: Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. IEEE T. Neural Networ. 7(4), 869–880 (1996)
21. Widrow, B., Lehr, M.: 30 Years of adaptive neural networks: perceptron, madaline and backpropagation. Proc. IEEE 78(9), 1415–1442 (1990)
22. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. J. Artif. Intell. Res. 6(1), 1–34 (1997)
23. Yen, G.G.: Multi-Objective evolutionary algorithm for radial basis function neural network design. Studies in Computational Intelligence 16, 221–239 (2006)

Performance Measures for Dynamic Multi-Objective Optimization

Mario Cámara[1], Julio Ortega[1], and Francisco de Toro[2]

[1] Dept. of Computer Technology and Architecture
[2] Dept. of Signal Theory, Telematics and Communications
University of Granada, Spain
`{mcamara,jortega}@atc.ugr.es, ftoro@ugr.es`

Abstract. As most of the performance measures proposed for dynamic optimization algorithms in the literature are only for single objective problems, we propose new measures for dynamic multi-objective problems. Specifically, we give new measures for those problems in which the Pareto fronts are unknown. As these problems are the most common in the industry, our proposed measures constitute an important contribution in order to promote further research on these problems.

1 Introduction

Dynamic optimization problems appear in many different real-world applications with high socio-economic relevance. They are dynamic because there are changes over time in the conditions on which the cost functions depend, in the restrictions that the solutions must meet, etc. For example, in some scheduling problems, such as those appearing in parallel computing servers, the available resources and the volume of tasks to be allocated could vary over time. The shifting of optima with time is an important issue in this kind of problems.

Dynamic multi-objective optimization (DMO) is comprised of both multi-objective problems, where there are more than one objective function to optimise, and non-stationary or dynamic problems, where the restrictions or the objective functions can change over time [1]. In the last years, a growing interest from researchers has focused on solving dynamic problems by using evolutionary algorithms [2,3,4].

In order to be able to assess algorithms by their performance, some measures were given over the last years [2,3,4,5,6,7]. However, most of them were aimed either at dynamic problems having only one objective or at problems where the location of the real Pareto fronts is known beforehand.

In this paper, we present some measures designed for dynamic multi-objective problems. Our measures tackle the difficulty that arise when the location of the real Pareto fronts is unknown, as it happens with most real world DMO problems.

In section 2 a review of the previous work done in dynamic optimization is given. Then, section 3 presents the measures that we propose. Section 4 offers some results using these measures. Finally, conclusions are given in section 5.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 760–767, 2009.
© Springer-Verlag Berlin Heidelberg 2009

2 Previous Work

A very important issue when developing algorithms is to have a suite of performance measures to compare the different algorithms. Despite of the fact that in non-stationary multi-objective optimization is not feasible to have a definitive set of such performance measures [8], some quality indicators have been created with the same goal [9].

However, in addition to performance indicators like those for stationary multi-objective algorithms, dynamic problem optimizers need another type of performance measures. They should point out which algorithm from the available ones best suits the current needs, and if it would be able to deal with the problem at hand on time. This preliminary study should take place before the commercial exploitation of the system begins. Thus, it allows the designer to use *offline* measures instead of *on-line* or *on-the-fly* measures.

Some of the few existing performance measures on this topic are considered in what follows.

Morrison [6] introduces a new measure known as *Collective Mean Fitness* F_C, which is a single value that is designed to provide an aggregate picture of an EAs performance, where the performance information has been collected over a representative sample of the fitness landscape dynamics. *Collective fitness* is defined as the mean best-of-generation values, averaged over a sufficient number of generations, G', required to expose the evolutionary algorithm to a *representative sample* of all possible landscape dynamics, further averaged over multiple runs.

Weicker [5] proposes measures for what he describes as the three different aspects that have to be taken into account when analyzing and comparing dynamic problems algorithms. The first is the *accuracy* measure which should measure the closeness of the current best found solution to the actual best solution. It usually takes values between 0 and 1, being 1 the best accuracy value. Accuracy is defined as:

$$accuracy_{F,EA}^{(t)} = \frac{F(best_{EA}^{(t)}) - min_F^{(t)}}{max_F^{(t)} - min_F^{(t)}} \tag{1}$$

where $best_{EA}^{(t)}$ is the best found solution in the population at time t. The maximum and minimum fitness values in the search space are represented by $max_F^{(t)}$ and $min_F^{(t)}$ respectively. F is the fitness function of the problem.

Weicker also stated that *stability* is an important issue in the context of dynamic optimization. A dynamic algorithm is called stable if changes in the environment do not affect the optimization accuracy severely. Hence, a definition of stability was given as:

$$stab_{F,EA}^{(t)} = \max\left\{0, accuracy_{F,EA}^{(t)} - accuracy_{F,EA}^{(t-1)}\right\} \tag{2}$$

and takes values from 0 to 1. In this case, a value close to 0 means high stability.

A third aspect of interest in dynamic problems is the ability of an algorithm to react to changes. Weicker proposes to check that an algorithm has ε-reactivity at time t using the next equation:

$$react_{F,A,\varepsilon}^{(t)} = \min \left\{ \left\{ t' - t | t < t' \leq maxgen, t' \in \mathbb{N}, \frac{accuracy_{F,EA}^{(t')}}{accuracy_{F,EA}^{(t)}} \geq (1 - \varepsilon) \right\} \cup \{maxgen - t\} \right\} \tag{3}$$

where $maxgen$ is the number of generations. The measure $react_{F,A,\varepsilon}^{(t)}$ evaluates how much time Δt it takes the algorithm to achieve a desired accuracy threshold.

The problem shown by all the measures proposed until now is that they were designed only for dynamic single objective problems and not for multi-objective optimization problems. In [3], we proposed an adaptation of Weicker's measures for the multi-objective problems. Here, we improve and extend those measures.

On the other hand, Li et al [2] also proposed some measures for DMO. The first one is called $rGD(t)$, which is based in the GD measure [10]. The drawback of this measure is that the real Pareto front must be known at any time. Nevertheless, it could be a very good substitute of Weicker's accuracy for multi-objective problems. They also propose the use of the *hypervolume ratio*[1] which is identical to our accuracy proposed at the same time [3]. Finally, they offer an adaptation of the collective mean error based on $rGD(t)$.

3 Improved Measures for Multi-Objective Algorithms

When taking into account measures for dynamic multi-objective problems it is important to distinguish between those problems in which the current real Pareto front is known at every time of the run of the algorithm and those in which the real Pareto front is rather unknown. The latter is the usual case in real world problems, and even in many of the suggested test cases for researching purposes.

All of the already indicated measures that have been previously proposed are either only for single objective problems or for those multi-objective problems in which the real Pareto fronts $F_{real}(t)$ are known at any time.

In the rest of the paper, we will use the following terms and notation to designate the different aspects of each problem, which is strongly influenced from that found in [1]. The terms *decision space* and *objective space* refer, respectively, to the spaces of the search variables and of the objective functions of that search variables. Also, we call the set of non-dominated solutions found at time t as *approximate Pareto optimal solutions* at time t, which is divided into the *approximate decision space* at

---

[1] Hypervolume is a measure defined, by Zitzler et al [9], as the size of the space covered by a given set of solutions from a given reference point. It is similar to the concept of a discrete integral.

time t, $S_P(t)$, and the *approximate objective space* or *approximate Pareto front* at time t, $F_P(t)$. The real Pareto front, that can be known or not at time t, is denoted as $F_{real}(t)$ and is always in the objective space.

In order to deal with real problems where the location of the real Pareto fronts is unknown we propose to redefine the measures previously proposed that rely on the knowledge of the real Pareto fronts. Nevertheless, the only measure that has to be adapted is $acc(t)$ (1) as the other two, $stb(t)$ (4) and $reac_\varepsilon(t)$ (5), rely on the knowledge of the real Pareto fronts only through $acc(t)$ (1).

For completeness purposes we reproduce our definitions for $stb(t)$ (4) and $reac_\varepsilon(t)$ (5):

$$stb(t) = \begin{cases} stb_0(t) & \text{if } stb_0(t) \geq 0 \\ 1 & \text{otherwise} \end{cases} ; \text{ with } stb_0(t) = acc(t) - acc(t-1) \quad (4)$$

$$reac_\varepsilon(t) =$$
$$\min\left\{ \left\{ t' - t | t < t' \leq maxgen, t' \in \mathbf{N}, \quad \frac{acc(t')}{acc(t)} \geq (1-\varepsilon) \right\} \cup \{maxgen - t\} \right\} \quad (5)$$

Nevertheless, we note again that as we are interested in offline measures we can exploit the knowledge of all the approximate Pareto fronts, not just the past ones.

Hence, in order to improve our accuracy definition for those cases in which the real Pareto fronts are unknown, we need to replace the hypervolume of the fronts by other suitable quantities. These quantities could be the maximum and minimum hypervolumes over all the time. However, if the Pareto fronts of the objective space of the problem change, those absolute maximum and minimum measures could be far from the current real Pareto front. Because of this, we propose to adapt the concept of accuracy, and consider it as a measure of the current approximate Pareto front in comparison only with the nearby approximate fronts, both in the past and in the future. This is the concept of accuracy within a *window* or *offset* which was already mentioned by Weicker [5].

A window is a period of time in which the problem becomes *stationary*: the problem does not show clear changes in the approximate Pareto fronts that have been found for those times. Thus, windows mark phases of the problem. Each phase is characterized by the moment in which the change has happened (when the phase starts) and its length in time units. We suggest that the window length should be variable instead of fixed. If the problem under study changes with an exact frequency, this window length should turn out to be equal to the inverse of the frequency. But in order to widen the set of problems that the new measure could analyze, it must be able to cope with variable frequencies.

Therefore, our improved measure has two parts. First, the windows or phases are detected and the lengths corresponding to each phase are calculated. Then, the accuracy measure is calculated for every time step using the relative minimal or maximal hypervolume values within that phase. To calculate the lengths of all the phases we propose the Procedure 1.

Procedure `1. Calculation of lengths`

Input: A set of N hypervolume values for the approximate Pareto fronts.
Output: A set S of the lengths of each of the phases.

```
1  begin
2      for $i = 2$ to $N$ do
3          $\Delta HV_i = HV_i - HV_{i-1}$
4      $length = 1$
5      for $i = 2$ to $N$ do
6          if $\Delta HV_i \geq |\Delta HV_{i-1} + \Delta HV_{i+1}|$ then
7              $S \longleftarrow S \cup length$
8              $length = 1$
9          else
10             $length = length + 1$
11 end
```

The conditions inside the *if* in line 6 of Procedure 1 can be changed to other conditions which may be suitable to detect a change in the fronts. Once the lengths have been obtained the accuracy for every approximate Pareto front is calculated by using (6), which makes use of (7) and (8). In them, HV is the hypervolume of the given set of solutions.

$$acc_{unk}(t) = \begin{cases} acc_{unk}^{maximizing}(t) & \text{if the problem is maximizing} \\ acc_{unk}^{minimizing}(t) & \text{if the problem is minimizing} \end{cases} \tag{6}$$

$$acc_{unk}^{maximizing}(t) = \frac{HV_{min}(Q(t))}{HV(F_p(t))} = \frac{HV(F_p(\min\{Q(t)\}))}{HV(F_p(t))} = \frac{V_{min}^{Q(t)}}{V(t)} \tag{7}$$

$$acc_{unk}^{minimizing}(t) = \frac{HV(F_p(t))}{HV_{max}(Q(t))} = \frac{HV(F_p(t))}{HV(F_p(\max\{Q(t)\}))} = \frac{V(t)}{V_{max}^{Q(t)}} \tag{8}$$

where $Q(t)$ is a set containing the time values for the window in which t takes place, i.e., the surrounding past and future values of t in which the approximate Pareto fronts have not suffered noticeable changes in their hypervolume measures, according to Procedure 1. The cardinality of each $Q(t)$ is the length of the phase that $Q(t)$ represents. Thus, $V_{max}^{Q(t)}$ and $V_{min}^{Q(t)}$ are, respectively, the maximum and minimum hypervolume of the fronts within the phase defined by $Q(t)$.

4 Results

In this section we show how these measures work on a set of approximate Pareto fronts corresponding to the problem *FDA3-mod* [11]. This problem is a modification we proposed for FDA3 [1] in order to avoid some underlying problems in it. In this function, every 5 time steps the solutions are forced to change in the decision and objective spaces, through a controlling parameter τ_i which is constant over the run of the algorithm. Due to space limitations only the first 21 time steps are shown. The problem was solved with our SFGA algorithm [12], a generational MOEA which keeps in the population only the first dominating rank of the non-dominated solutions. In spite of the fact that the lengths of the problem phases were calculated using the Procedure 1, they are not shown in the table because they equal 5 after the changes were forced in the problem, as expected, because of the existence of the parameter $\tau_i = 5$.

Table 1 shows the measures for the found Pareto front for each time step, and the difference in hypervolume between a found Pareto front and the previous one.

Table 1. Performance Measures for FDA3-mod

Time	**Hypervolume**		**Measures**		
	Absolute	**Change**	**Accuracy**	**Stability**	**Reaction time**
t	$HV(t)$	$\vert \Delta HV(t)\vert$	$acc(t)$	$stb(t)$	$reac(t)$
1	3,7213	-	0,984	-	1
2	3,7562	0,035	0,993	0,009	1
3	3,7734	0,017	0,998	0,005	1
4	3,7780	0,005	0,999	0,001	1
5	3,7818	0,004	1,000	0,001	1
6	3,6845	0,097	0,989	1,000	**5**
7	3,7046	0,020	0,995	0,005	1
8	3,7141	0,010	0,998	0,003	1
9	3,7177	0,004	0,998	0,001	1
10	3,7221	0,004	1,000	0,001	1
11	3,6175	0,105	0,986	1,000	**5**
12	3,6482	0,031	0,995	0,008	1
13	3,6571	0,009	0,997	0,002	1
14	3,6627	0,006	0,999	0,002	1
15	3,6681	0,005	1,000	0,001	1
16	3,5648	0,103	0,986	1,000	**5**
17	3,5965	0,032	0,995	0,009	1
18	3,6094	0,013	0,998	0,004	1
19	3,6143	0,005	0,999	0,001	1
20	3,6162	0,002	1,000	0,001	1
21	3,5339	0,082	0,999	1,000	**5**

The importance of the measures presented in this paper is twofold. First, all the data shown in Table 1 has been gathered and calculated without reckoning on the location of the real Pareto fronts. This feature widens immensely the range of problems which could be tackled with these measures. On the other hand, they are the first measures for dynamic and multi-objective problems which specifically address the three different but important aspects of dynamic problems as suggested by Weicker [5]: the accuracy, the stability and the reaction time.

From Table 1 it is clear that the measures proposed by Weicker and adapted in this paper to multi-objective problems are useful when tackling a dynamic problem like *FDA3-mod*. The reason for that is that Procedure 1 was able to detect all the phases that happened in the problem and exactly in the actual time step in which they appeared. Also, very suitable accuracy measures for every approximate Pareto front were found. That was possible because the accuracy was a measure related exclusively to the relative minimum or maximum approximate Pareto front appearing in that phase.

For instance, if we take the approximate Pareto fronts found at times 1 and 16 given in Table 1, they have almost the same accuracy (around 0,985), but their hypervolumes values differ more than in 4%. However, if absolute minimum and maximum hypervolumes had been used in $acc_{unk}(t)$, that differences in the hypervolume values would have been translated into the accuracy measure, yielding that the first approximate Pareto front would be better than the sixteenth in terms of accuracy. This turns out to be incorrect. The true is that the first approximate Pareto front can be worse or better but only in comparison to those other approximate fronts which belong to the first phase of the problem, and the same happens with the sixteenth approximate Pareto front, but in this case, in the fourth phase.

Finally, these proposed measures can be used to make a choice on which algorithm could fit better for a specific application. In that case, tests are done using different algorithms and the accuracy measure is used to choose which one of the tested algorithms shows the best results in optimization terms. Then, the other two measures are very useful in order to make sure that the chosen algorithm is able to cope on time with the given problem.

5 Conclusions

In this paper we have proposed performance measures for dynamic multi-objective optimization in order to address problems where the location of the real Pareto fronts is unknown.

DMO is a hot topic and so this contribution may turn out to be quite useful. We expect that researchers will take advantage of the proposed performance measures for their research. Precisely, our current and future work on parallel evolutionary algorithms for dynamic multi-objective optimization, some of which first results are shown in [11], will take advantage of these measures.

Acknowledgement. Work funded by the projects TIN2007-60587 (Spanish Ministry of Science and Technology) and P07-TIC-02988 (Junta de Andalucía), and by the programme for early stage researchers by the Andalusian government, co-financed with FEDER funds.

References

1. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: Test cases, approximations, and applications. IEEE Trans. on Evolutionary Computation 8, 425–442 (2004)
2. Li, X., Branke, J., Kirley, M.: On performance metrics and particle swarm methods for dynamic multiobjective optimization problems. In: IEEE Congress on Evolutionary Computation, pp. 576–583 (2007)
3. Cámara, M., Ortega, J., de Toro, F.J.: Parallel processing for multi-objective optimization in dynamic environments. In: Proceedings Of The 21st International Parallel And Distributed Processing Symposium, IPDPS 2007 (2007)
4. Alba, E., Saucedo, J.F., Luque, G.: 13. In: A Study of Canonical GAs for NSOPs. Panmictic versus Decentralized Genetic Algorithms for Non-Stationary Problems, pp. 246–260. Springer, Heidelberg (2007)
5. Weicker, K.: Performance measures for dynamic environments. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 64–76. Springer, Heidelberg (2002)
6. Morrison, R.: Performance measurement in dynamic environments. In: Branke, J. (ed.) GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, pp. 5–8 (2003)
7. Hatzakis, I., Wallace, D.: Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: GECCO 2006: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp. 1201–1208. ACM, New York (2006)
8. Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C.M., da Fonseca, V.G.: Why quality assessment of multiobjective optimizers is difficult. In: GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 666–674. Morgan Kaufmann Publishers Inc., San Francisco (2002)
9. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)
10. Van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph.D thesis, Wright-Patterson AFB, OH (1999)
11. Cámara, M., Ortega, J., de Toro, F.: Parallel multi-objective optimization evolutionary algorithms in dynamic environments. In: Lanchares, J., Fernández, F., Risco-Martín, J.L. (eds.) Proceedings of The First International Workshop On Parallel Architectures and Bioinspired Algorithms, vol. 1, pp. 13–20 (2008)
12. de Toro, F., Ortega, J., Ros, E., Mota, S., Paechter, B., Martn, J.M.: PSFGA: Parallel processing and evolutionary computation for multiobjective optimisation. Parallel Computing 30, 721–739 (2004)

Methods for Artificial Evolution of Truly Cooperative Robots

Dario Floreano[1] and Laurent Keller[2]

[1] Laboratory of Intelligent Systems, Ecole Polytechnique Federale de Lausanne
[2] Department of Ecology and Evolution, University of Lausanne
Switzerland
`{Dario.Floreano}@epfl.ch`

Abstract. Cooperation applies the situations where two or more individuals obtain a net benefit by working together. Cooperation is widely spread in nature and takes several forms, ranging from behavioral coordination to sacrifice of one's own life for the benefit of the group. This latter form of cooperation is known as "true cooperation", or "altruism", and is found only in few cases. Truly cooperative robots would be very useful in conditions where unpredictable events may require costly actions by individual robots for the success of the mission. However, the interactions among robots sharing the same environment can affect in unexpected ways the behavior of individual robots, making very difficult the design of rules that produce stable cooperative behavior. It is thus interesting to examine under which conditions truly cooperative behavior evolves in nature and how those conditions can be translated into evolutionary algorithms that are applicable to a wide range of robotic situations.

1 Introduction

Building on earlier intuitions by [3,4] suggested that altruism can evolve if the cooperator is genetically related to the recipient of help. In this case, even if the cooperator cannot propagate its own genes to the next generation, its altruistic act will increase the probability that a large portion of those genes will be propagated through the reproduction of the recipient of the altruistic act. The theory of kin selection [6], which developed from Hamilton's model, predicts that the ratio of altruistic individuals in a population is related to the degree of kinship, or genetic relatedness, among individuals. Another explanation for the evolution of altruistic cooperation is provided by the theory of group selection, which argues that altruistic cooperation may also evolve in groups of genetically unrelated individuals that are selected and reproduced together at a higher rate than the single individuals composing the group [10]. This could happen in situations where the synergetic effect of cooperation by different individuals provides a higher fitness to the group with respect to other competing groups. In those situations, cooperating individuals can be seen as a

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 768–772, 2009.
© Springer-Verlag Berlin Heidelberg 2009

superorganism that becomes the unit of selection. It has been suggested that group selection may be a driving force behind the transition from unicellular to multicellular organisms [7].

In the context of evolution experiments with robots [1,2], the hypotheses of kin selection and of group selection translate into the genetic composition of the group of robots and into the method used to select individuals for reproduction. If we consider only the extreme conditions, individuals in a team could be genetically homogeneous (clones) or heterogeneous (they differ from each other); and the fitness could be computed at the level of the team (in which case, the entire team of individuals is reproduced) or at the level of the individual (in which case, only individuals of the team are selected for reproduction).

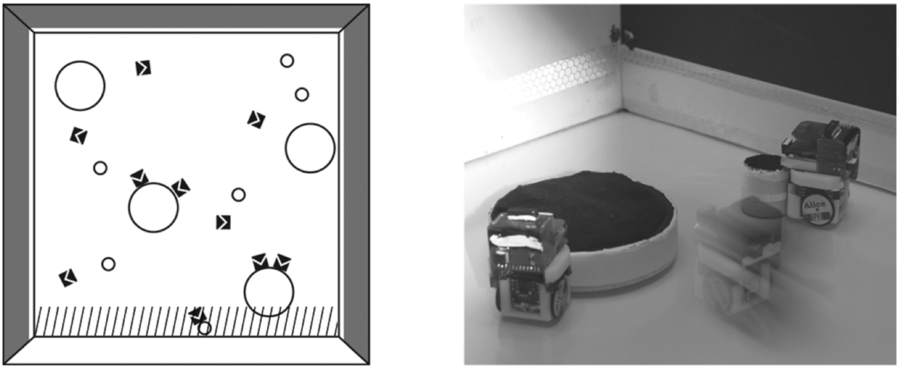


Fig. 1. Left: The experimental setup for the altruistic foraging task. Ten microrobots (black squares with arrows) searched for small and large tokens and transported them to the target area (hatched area at bottom) under the white wall (the other three walls were black). An identical setup was used in the other two experimental conditions, except that the arena contained either only small tokens in the collective task (no cooperation required), or only large tokens in the cooperative task (cooperation required, but no cost to the individual). Right: Three microrobots in the altruistic cooperative foraging task. The robot in the background could transport the small token by itself. The robot at the left could not transport the large token by itself and needed to wait for the arrival of a second robot.

We compared the performance of genetically homogeneous teams evolved with individual-level selection, genetically homogeneous teams evolved with team-level selection, genetically heterogeneous teams evolved with individual-level selection and genetically heterogeneous teams evolved with team-level selection [9]. The experiments were conducted with teams of sugar-cube robots in a foraging task that could be manipulated to offer the opportunity for various degrees of cooperation (figure 1). The genomes encoded the synaptic connections of the robot neurocontrollers. The results indicated that the best combination of genetic composition and level of selection depend on the degree of cooperation requested by the task. When the task does not require costly cooperation, teams of heterogeneous individuals evolved under individual-level selection obtain the best performance. However, when the task involves a cost for the individual, genetic relatedness or team-level selection generate neural controllers with the best performance.

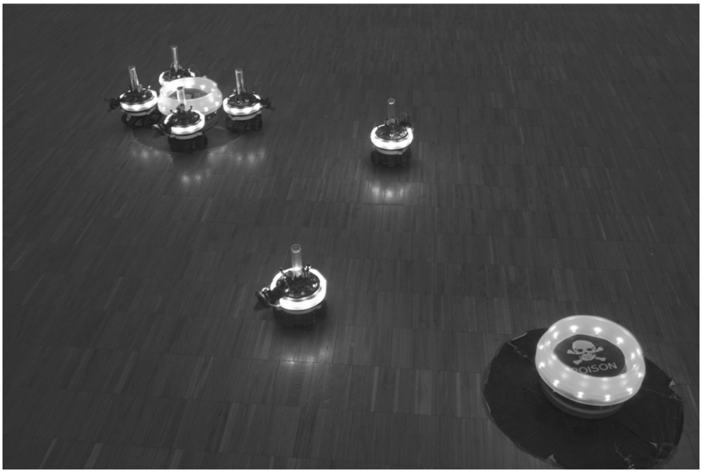


Fig. 2. Robots are positioned in an arena with a food object and a poison object that look the same at distance. Robots can communicate by changing color and by observing the color of the surrounding robots. The picture shows an evolved group of four robots feeding on the food object positioned in the top left corner while they are lighted up in blue color. Two robots in white color are attracted by the blue signal and move away from the poison object positioned in the bottom right corner.

In another set of experiments (figure 2), we compared the four combinations of genetic composition and level of selection with groups of "expressive" robots that are exposed to food and danger sources that cannot be uniquely be identified at distance [2]. In these experiments, the evolutionary emergence of communication of the source type brings an advantage to the group at the expense of the individuals that decide to tell which is the food or poison. The results indicated that communication by color emerged in populations of robots that had been evolved under the condition of genetic relatedness or of team-level selection, but not in populations of robots that were genetically unrelated and evolved under individual-level selection

Finally, we turned to an application scenario where a swarm of Micro Aerial Vehicles (MAVs) are deployed in a catastrophic scenario to locate the rescuers on the ground and establish an ad hoc radio network that allows rescuers to communicate with each other (figure 3). The stringent mission requirements, which ruled out the use of Geo Positioning System and of vision sensors along with the unpredictable location of the rescuers on the ground, made it very difficult to come up with suitable control systems for the micro aerial vehicles. Consequently, we resorted to the evolutionary methods that we designed and evaluated in the previous experiments. In this situation, the fitness was proportional to the quantity of communication packets that could be exchanged among rescuers during a period of 30 minutes, which made it impossible to compute the contribution of individuals in the team. Furthermore, robots could incur a cost by searching for rescuers instead of participating to the maintenance of an already established radio network. Consequently, we resorted to teams of genetically related controllers evolved under team-level selection [5]. The genomes of the robots encoded the synaptic weights of the neurocontrollers of the robots. Evolution experiments were carried out in simulations that modeled both the flight and the radio network dynamics. Evolved teams were capable of rapidly locating a rescuer located at an unknown position and maintaining a reliable voice connection with another rescuer. Current work is aimed at transferring the evolved control systems to a swarm of real MAVs and extending their use to multiple rescuers on the ground.

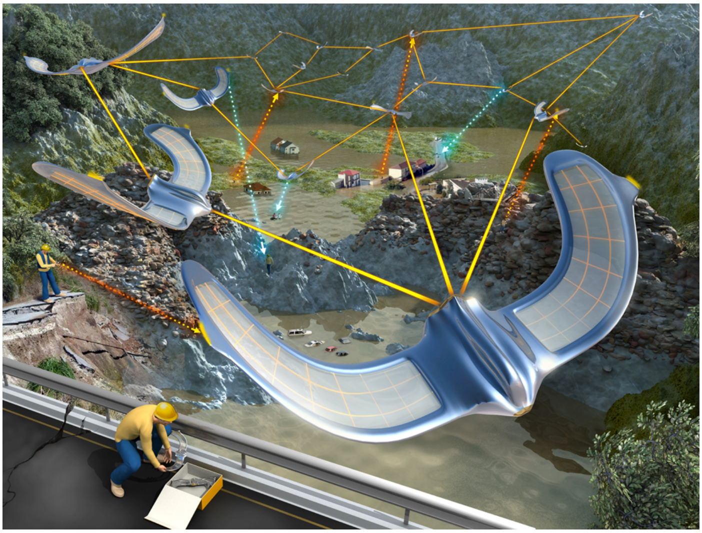


Fig. 3. A team of MAVs equipped with wireless communication devices are deployed by hand in a catastrophic scenario. The robots must locate the rescuers on the ground uniquely by means of radio signal and deploy themselves oin order to establish an ad hoc radio network that allows rescuers to communicate.

Cooperation could be mediated by many other factors both in nature and in robotics. For example, if organisms and robots were capable of recognizing each other and memorizing the effects of their interactions, one could observe the emergence of reciprocal sharing and even of a reputation system. However, these experiments indicate that even robots whose control system cannot recognize other individuals, cannot learn, and cannot memorize events can still engage in sophisticated forms of altruistic cooperation.

Acknowledgments. This work was supported by the Swiss National Science Foundation and by the Science and Technology Division of Armasuisse.

References

1. Floreano, D., Mattiussi, C.: Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies. MIT Press, Cambridge (2008)
2. Floreano, D., Mitri, S., Magnenat, S., Keller, L.: Evolutionary Conditions for the Emergence of Communication in Robots. Current Biology 17, 514–519 (2007)

3. Haldane, J.B.S.: Population genetics. New Biology 18, 34–51 (1955)
4. Hamilton, W.D.: The genetical evolution of social behavior, I and II. Journal of Theoretical Biology 7, 1–52 (1964)
5. Hauert, S., Zufferey, J.-C., Floreano, D.: Evolved swarming without positioning information: an application in aerial communication relay. Autonomous Robots (2009) (in press)
6. Maynard-Smith, J.: Group selection and kin selection. Nature 201, 1145–1147 (1964)
7. Michod, R.E.: Darwinian Dynamics. In: Evolutionary Transitions in Fitness and Individuality. Princeton University Press, Princeton (1999)
8. Nolfi, S., Floreano, D.: Evolutionary Robotics: Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press, Cambridge (2000)
9. Waibel, M., Keller, L., Floreano, D.: Genetic Team Composition and Level of Selection in the Evolution of Multi-Agent Systems. IEEE Transactions on Evolutionary Computation (2009) (in press)
10. Wynne-Edwards, V.C.: Evolution through Group Selection. Blackwell, Palo Alto (1986)

Social Robot Paradigms: An Overview

Sergi del Moral, Diego Pardo, and Cecilio Angulo

Centre d'Estudis Tecnolgics per a la Dependncia,
CETpD Universitat Politcnica de Catalunya, UPC
`{sergi.delmoral,diego.pardo,cecilio.angulo}@upc.edu`
`http://www.upc.com/cetpd`

Abstract. This article overviews the current and diverse paradigms about a social robot. We discuss and contextualize key aspects about the mechanisms used by this type of robot to relate with humans, other robots and its surrounded social environment: interaction, communication, physical and environmental embodiment and autonomy. We enrich current definitions and discuss about the crucial role of the purpose of this type of robots within nowadays society.

Keywords: social robot, socially interactive robot, sociable, interaction, communication, social context.

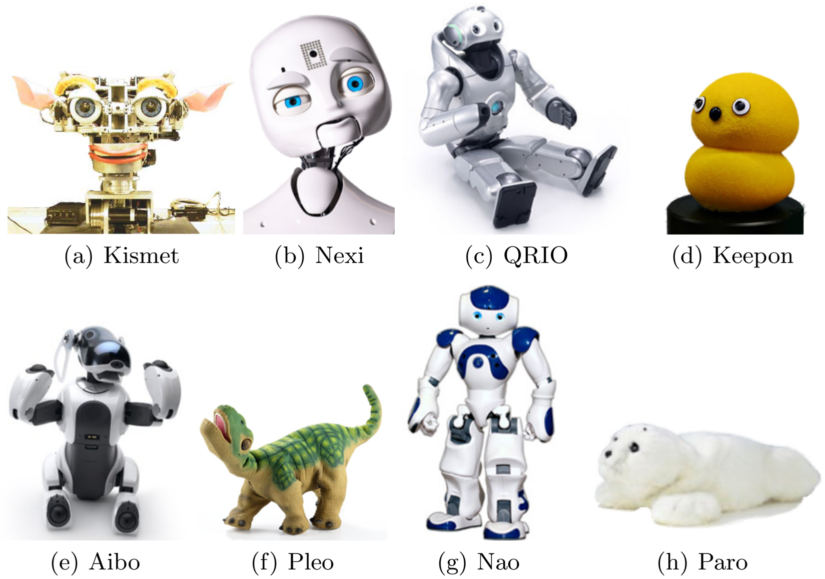
1 Introduction

Human being has been constructing automatic machines during thousands of years [14,31], nevertheless the development of robotics has exploited at the end of the last century. The *United Nations Economic Commission for Europe* (UNECE) together with the *International Federation of Robotics* (IFR) recently presented a report about the state and expectations of research in robotics [34]. They propose three main categories in order to characterize the field: industrial robotics, professional service robotics and personal service robotics.

Industrial robotics (manipulators and productive oriented robots) represents the most traditional area, which is algo the most developed within the field of robotics. This is related with its direct connection with productivity and investment corporations interested in widely demanded commercial products. In the field of professional service robotics (e.g. robotised surgery equipment or autonomous transportation systems) it is important to highlight that they are already present in daily services. Besides, prediction for the near future is an exponential grow in the robotics applications that community will have available for help in its daily work [34].

Finally, personal services robotics (domestic, assistance and entertainment) is an embryonic field and then it still is an open area of research, with big challenges in a variety of aspects. Social robotics [7], also known as the field of expertise in charge of *socially interactive robotics* [13], or *socially intelligent robots* [5], is a very young research field that focuses in the study of those robots where mechanisms that relate with humans, other robots, and its surrounded social environment are specially important [13,15].

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 773–780, 2009.
© Springer-Verlag Berlin Heidelberg 2009



(a) Kismet (b) Nexi (c) QRIO (d) Keepon

(e) Aibo (f) Pleo (g) Nao (h) Paro

Fig. 1. Several labs platforms, prototypes and first constructive attempts of social robots in (a)-(d). And commercial robots with certain interaction capacity in (e)-(h).

Besides a more natural appearance, nowadays robots already include high computational power, friendly interfaces and advanced on-board communication devices that allow them, or at least are expected, to develop effective social behaviors when interacting with people. There are several labs that already have research platforms, prototypes and first constructive attempts of social robots. Kismet [6] and Nexi [25] robots from MIT, the humanoids HPR-2 [19], ASIMO [17] or QRIO [30] from Sony Corp. and the Keepon creature [20] from NICT are some examples of it. Moreover, today exists commercial robots with certain interaction capacity, e.g (Aibo [32], Pleo [29] Nao [1] and Paro [28]). Figure 1 shows the mentioned robots.

Nevertheless, the meaning of what *social robot* means still a diffuse concept. Due to its technological connotation the definition of this term is constantly under debate, and consequently its meaning has evolved over time in parallel to existing technological paradigms. It is essential to arrive, by consensus, to formalize the future lines of actuation of the research area. In this article we outlook the meaning of social robotics and enrich its description, fitting its actions within a model of interaction and highlighting this last as one of the critical point under which the concept of a social robot must be supported.

This paper is composed as follows. Section 2 reviews simple and diverse paradigms about social robotics, whereas in section 3 a discussion is open about them, pointing to an enriched and contemporaneous definition. Finally, section 4 gathers together the conclusions.

2 Paradigms of Social Robot

Defining with precision what should be denoted as a social robot is not a naive task. There are many authors that have gave their point of view but at the present time there is no a general consensus around a formal definition. Nevertheless, there are common points between the diverse paradigms, now we surf over a spectrum of authors that have proposed ideas about this young robotics field.

First use of the term *social robot* was adopted in the nineties to identify those robotic machines that belong, as inidividuals to collectives of robots [4,9,16,22]. Scientific community has now clear that those efforts were overcome by understanding that the social competence is the ability of interaction with a living organism and not in the capacity of interaction within machines [36]. First efforts differentiating this were presented in [10], suggesting that robots must behave conductively. They also establish communication, collaboration and coordination skills as requirements for the robot as mechanism of relation [11]. Unfortunately, interaction and autonomy are not highlighted within their definition: *A physical entity embodied in a complex, dynamic, and social environment sufficiently empowered to behave in a manner conducive to its own goals and those of its community.*

The embodiment is an important feature included in this definition. An individual is embodied in an environment if it is able to perturb or susceptible to be perturbed by the environment [13], therefore definition in [10] lacks in the sense that embodied condition is necessary and not sufficient for a social agent, and it does not imply that it includes social skills. On the other hand the nature of the goals of the robot is what demands indispensable attitudes as interaction and communication.

The embodiment property, used as key attribute for social robots in [11], may be divided in two types: (1) that referring as a requirement over the control process to be implemented inside a physical body [8,11], and (2) that describing the relation between the robot and its surrounding environment [13]. There are authors that believe that a social robot is not required to exist within a physical body [6,13], others [23] agree in setting both restrictions to the requirements on the social robot. Physical embodiment is inherent to the robot condition. Software agents living in other computational devices like computers, mobiles or PDA's are not included in our vision of robots. Therefore is the environmental embodiment is critical for the establishment of satisfactory human-robot interactions within social context [23]. However, it is important to point out that in the terminology employed by [10], a robot is described simply as a physical entity with conductive capacities, those agents that are not robots are being incorrectly included.

Another interesting analysis on the concept of social robot is discussed in [6] where autonomy is presented as a fundamental condition in order to consider a mechanism sociable. This is also mentioned in [3], where restrictions are relaxed by also considering the semi-autonomous systems devices, this is, those that are not technologically capable to act without any human supervision but that are able to take autonomous decisions. Besides autonomy, another condition proposed in [6] to a social robots is that it must be ascribed to a social model by those humans with whom the robot interacts. People associates a social model when interacts with the robot and when tries to understand its messages.

This implies that the human-robot interaction is crucial for the concept of social robots. It seems that the main feature of these robots is the interaction capacity and that, with the goal of establish effective human-robot interactions, it is mandatory to donate the robot with skills like, communication, gestural capacity, identity, etc. As mentioned in [6] the power and quality of the robot interaction depends in how adjustable to the ascribed social model is. As more adaptable the ascribed social model more satisfactory will result the interaction. This launches an interrogant over the availability and effectiveness of the current metrics for the measurement of the interaction, nevertheless but the analysis of this issue is out of the scope of this paper.

In terms of how well the robot supports the social model that is ascribed to it and the complexity of the interaction scenario a differentiation was proposed by Breazeal in [6]. It includes robots known as *socially evocative*, that manipulate the human filings based on the tendency of humans beings of evoking sentiments associated with care, e.g. Tamagotchi [33] and robotic pets. Other category are those robots called *sociable* (e.g. Kismet robot [6]) which are programmed with complex models of social cognitively and then are able of establish pro-actively relations with the intention of satisfy social ambitions. As pointed out in [6] empowering robot skills like communications, cooperation or learning is almost impossible without an anthropomorphic appearance. However, the term anthropomorphism [12,24,27] is not properly used in [6] when referring to both animals and humans.

Other authors like [15] recently propose an open definition: *social robotics is the study of robots that interact and communicate among themselves, with humans, and with the environment, within the social and cultural structure attached to their roles.* Thus they agree with the idea of interaction and communication as the key components in the conformation of a social robot. They go one step forward mentioning that once the robots overcome the critical aspects of interaction, they should be able to socialize within them. Another encounter point mentioned in [15] is that those skills given to the social robot are subject to the social context within it is embedded, materialized in what they have denominate 'a role'. This definition is too generic that englobes the more important points analyzed so far: interaction, communication and social context. It is non worthless that this is referenced in the preface of the inaugural volume of *International Journal of Social Robotics* [15].

Finally, we cite 2004 Bartneck's definition in [3] as one of the most contemporaneous vision of the social robotics context. They characterize this type of robots as those able to interact and communicate with humans, following the rules of behaviors expected by those with whom the robot is designated to interact with. Again, interaction and communications are the main road of the paradigm. In certain sense, this definition is complementary to those reviewed above, giving that the social model that we ascribe to the robot depends on the social context.

3 Discussion

Provided that there is no agreement about the meaning of the term *social*, which is employed in several different contexts [36], the meaning of social robotics is subject to the connotation from where the concept is reviewed, leading to significantly distanced paradigms. An instance of this is evidenced analyzing the original ideas, which were oriented to the concept of a society of robots, its reaches, dangers and laws. In contrast, nowadays social robotics research is oriented to the study of the robots and its capabilities of interaction, communications, comprehension and even relationship with humans [5].

Nevertheless, we think that despite present definitions on social robotics precisely synthesize the current paradigm, they overlook the purpose of the robots itself. For us, the behavior, the personality, and the attitudes of the robot must take into account the needs of the society i.e. the people that conforms it. The motivations that take us to endow social skills within robots are not fed with science fiction, but it is based in the potential roll that this mechanism can play in the future society. We have reached a point in technological developments in which the construction of robots with social behaviors and high interaction competences is possible. This social faculty enable them to perform functions where the relational and affective behavior is crucial, such as caring for dependent people [18], elderly nursing home [2], autism therpy [20] or health service facility [35].

Despite there are many aspects to discuss in order to completely define a social robot, we consider appropriate to enrich (and update) the Bartneck's definition as follows:

> *A Social Robot is an autonomous motion device equipped with sensors, actuators and interfaces (robot) that interacts and communicates with humans following some expected behavior rules, which are founded on the robot physical properties and the environment within it is embeded, mainly taking into account the needs of the people with witch it is meant to interact with*

We think that this definition gathers together the key features of nowadays social robotics. It is also as generic as the premature state of this field requires. Our definition starts by delimiting the type of element that is being used as a social agent: the robot. We delimit the concept of robot as a physical device, equipped with sensors, actuators and at least some sort of interface, that moves. Robot motions can be absolute or relative but it must demonstrate certain level of autonomy. Based on this properties robots are able to interact with people, this is, to receive and interpret oral, gestural, and behavioral information interpreting it correctly in order to act properly. This response must also be coded in such a way that the human understands the robot intentions. A key issue is aggregated in our definition in order to specify that the response (actions) of the robot are expected to be framed by an implicit set of rules. This rules are tacitly established by the human and are founded in, for example, the constructive

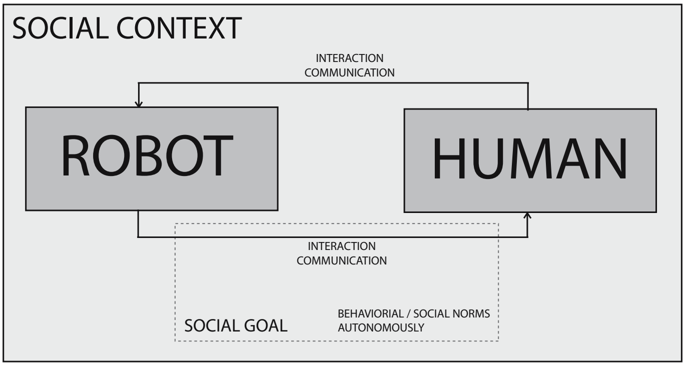


Fig. 2. Social robot scenario

form of the robot. Human interactions with humanoid robots are expected to be different that those with a robotic pet (e.g. a dinosaour or a dog) Humans expect different interaction channels depending on the form of the robot. Similarly, if the robot is embedded in a nursing home, its behaviors should not be the same that those of a robot that interacts in a kindergarten, this is, the behavior rules of the social robot must match environmental requirements and constraints. Finally, we remark that the social model donated to a robot must encapsule a clear purpose, and that that purpose must fulfill needs of those the robot will interact with. Figure 2 presents a diagram graphically describing the mentioned concepts.

4 Conclusions

Defining with precision what should be denoted as social robot is not a naive task. Nowadays the meaning of what it means still a diffuse concept.

We have overviewed actual social robot paradigms. The most important points analyzed so far are interaction, communication and the social context within the social robot is embodied.

Besides the motivations to endow social skills within robots are based in the potential roll that this mechanism can play in the future society, we have reached a point in technological developments in which the construction of robots with social behaviors and high interaction competences is possible. Therefore we propose an up to date social robot definition that not overlook the purpose of the robot itself.

Acknowledgments

This work is supported by the Generalitat de Catalunya and the Fundació Hospital Comarcal Sant Antoni Abat (FHCSAA) and conducted at the Technical Research Centre for Dependency Care and Autonomous Living (CETpD) of the Universitat Politècnica de Catalunya (UPC).

References

1. Aldebaran Robotcis, `http://www.aldebaran-robotics.com/`
2. Baltus, G., Fox, D., Gemperle, F., Goetz, J., et al.: Towards personal service robots for the elderly. In: Proceedings of the Workshop on Interactive Robots and Entertainment, Pittsburgh, PA (2000)
3. Bartneck, C., Forlizzi, J.: A Design-Centred Framework for Social Human-Robot Interaction. In: Ro-Man 2004, Kurashiki, pp. 591–594 (2004)
4. Beckers, R., et al.: From local actions to global tasks: Stigmergy and collective robitics. In: Proceedings of Artificial Live IV (1996)
5. Breazeal, C.: Socially Intelligent Robots. In: Interactions, vol. XII.2 (March/April 2005)
6. Breazeal, C.: Towards sociable robots. Robotics and Autonomous Systems 42, 167–175 (2003)
7. Breazeal, C.: Designing Sociable Robots. MIT Press, Cambridge (2002)
8. Dautenhahn, K.: Embodiment and Interaction in Socially Intelligent Life-Like Agents. In: Nehaniv, C.L. (ed.) CMAA 1998. LNCS, vol. 1562, pp. 102–142. Springer, Heidelberg (1999)
9. Deneubourg, J.-L., et al.: The dynamic of collective sorting robot-like ants and ants-like robots. In: Proceedings of the International Conference on Simulation of Adaptative Behavior (2000)
10. Duffy, B.R., Rooney, C.F.B., O'Hare, G.M.P., O'Donoghue, R.P.S.: What is a Social Robot? In: 10th Irish Conference on Artificial Intelligence & Cognitive Science, September 1–3 (1999)
11. Duffy, B.R.: The Social Robot. Ph.D Thesis, Department of Computer Science, University College Dublin (2000)
12. Duffy, B.R.: Anthropomorphism and the social robot. Robotics and Autonomous Systems 42, 177–190 (2003)
13. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots. Robotics and Autonomous Systems 42, 143–166 (2003)
14. Gates, B.: A robot in every home. Scientific American, 58–65 (January 2007)
15. Ge, S.S., Matarić, M.J.: Preface. International Journal of Social Robotics 10.1007/s12369–008–0010–2 (2008)
16. Goldberg, D., Matarić, M.: Interference as a tool for designing and evaluating multi-robot controllers. In: Proceedings of the AAAI 1997, Providence, RI, pp. 637–642 (1997)
17. Honda Motor Co., `http://wolrd.honda.com/ASIMO/` (visited: 25/03/2009)
18. Huttenrauch, H.: Fetch-and-carry with CERO: Observations from a long-term user study with a service robot. In: Proceedings of the Interncational Workshop on Robot and Human Interactive Communication, Berlin, Germany, pp. 158–163 (2002)
19. Kawada Industries, `http://www.kawada.com.jp/global/ams/hrp_2.htm` (visited: 25/03/2009)
20. Kozima, H., Michalowski, M.P., Nakagawa, C.: Keepon, A Playful Robot for Research, Therapy, and Entertainment. International Journal of Social Robotics 1(1), 3–18 (2009)
21. Kube, C.R., Zhang, H.: Collective Robotics: From Social Insects to Robots. Adaptive Behavior 2(2), 189–219 (1993)
22. Mataric, M.: Issues and approaches in design of collective autonomous agents. Robotics and Autonomous Systems 16, 321–331 (1995)

23. Mataric', M.J.: The Role of Embodiment in Assistive Interactive Robotics for the Elderly. In: Proceedings of the AAAI Fall Symposium on Caring Machines: AI for the Elderly, Arlington, VA, November 4–6 (2005)
24. Hegel, F., Krach, S., Kircher, T., Wrede, B., Sagerer, G.: Understanding Social Robots: A User Study on Anthropomorphism. In: Proceeding of the International Symposium on Robot and Human Interactive Communication (RO-MAN), Munich, Germany (2008)
25. Nexi Robot, MIT Media Lab Web, `http://robotic.media.mit.edu/projects/robots/mds/overview/overview.html` (visited: 11/02/2009)
26. Montemerlo, M., Prieau, J., Thrun, S., Varma, V.: Experiences with a mobile robotics guide for the elderly. In: Proceedings of the AAAI National Conference on Artificial Intelligence, Edmunton, Alberta, pp. 587–592 (2002)
27. Oxford Dictionaries, AskOxford.com, `http://www.askoxford.com` (visited on: 05/02/2009)
28. Paro Robot Project Web, National Institute of Advanced Industrial Science and Technology, `http://www.paro.jp/english/` `http://www.paro.jp/`
29. Pleo World Web, UGobe, `http://www.pleoworld.com/` (visited: 11/02/2009)
30. QRIO Robot, `http://www.sony.net/SonyInfo/News/Press_Archive/200312/03-060E/` (visited: 11/02/2009)
31. Sánchez, F.M., Millán, F., Salvador, J., Palou, J., Rodríguez, F., Esquena, S., Villavicencio, H.: Historia de la robótica: de Arquitas de Tarento al robot Da Vinci (Parte I) (2007)
32. Sony AIBO ERS-7 Enternaiment Robot, `http://support.sony-europe.com/aibo/` (visited: 17/10/2007)
33. Bandai.com - Tamagotchi, `http://www.tamagotchi.com/` (visited: 11/02/2009)
34. United Nations and the International Federation of Robotics. World of Robotics (2002)
35. Wada, K., Shibata, T., Saito, T., Sakamoto, K., Tanie, K.: Psychological and Social Effects of One Year Robot Assisted Activity on Elderly People at a Health Service Facility for the Aged. In: Proceedings of the International Conference on Robotics and Automation (2005)
36. Wikipedia.org., `http://www.wikipedia.org/`, Term: Social. (visited: 6/02/2009)

A Dual Graph Pyramid Approach to Grid-Based and Topological Maps Integration for Mobile Robotics

J.M. Pérez-Lorenzo[1], R. Vázquez-Martín[2], E. Antúnez[3], and A. Bandera[2]

[1]Dpto. Ing. Telecomunicación, Universidad de Jaén (Spain)
[2]Grupo ISIS, Dpto. Tecnología Electrónica, University of Málaga (Spain)
`ajbandera@uma.es`
[3]PRIP, Vienna University of Technology (Austria)

Abstract. A pyramid is a hierarchy of successively reduced graphs which represents the contents of a base graph at multiple levels of abstraction. The efficiency of the pyramid to represent the information is strongly influenced by the graph selected to encode the information within each pyramid level (data structure) and the scheme used to build one graph from the graph below (decimation process). In this paper, the dual graph data structure and the maximal independent edge set (MIES) decimation process are applied in the context of robot navigation. The aim is to integrate the grid-based and the topological paradigms for map building. In this proposal, dual graphs allow to correctly represent the embedding of the topological map into the metric one.

1 Introduction

Autonomous navigation is a fundamental ability for mobile robots which requires the integration of different modules. Among them, self-localization and environment mapping are two essential ones, as they are needed at different levels, from low-level control to higher-level strategic decision making or navigation supervision. It is well known that to guarantee bounded errors on its pose estimates, the robot must rely on sensors which can perceive stable environment features. Thus, if the robot manages a spatially consistent map of the environment, it could apply a map-based localization approach to obtain a correct estimation of its pose [1]. On the other hand, if the robot pose is exactly known, it could build a consistent environment map with the perceived data. The mapping and localization tasks are then *intimately tied together* [2], and they must be concurrently solved. The problem of the simultaneous localization and mapping (SLAM) has been extensively addressed by the robotic community in the last years.

Specifically, two fundamental paradigms have been developed for modeling indoor environments in mobile robotics: the metric and the topological paradigms. Metric approaches are easy to build and they represent accurately the real features of the world [3]. However, they suffer from their huge data load and time

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 781–788, 2009.
© Springer-Verlag Berlin Heidelberg 2009

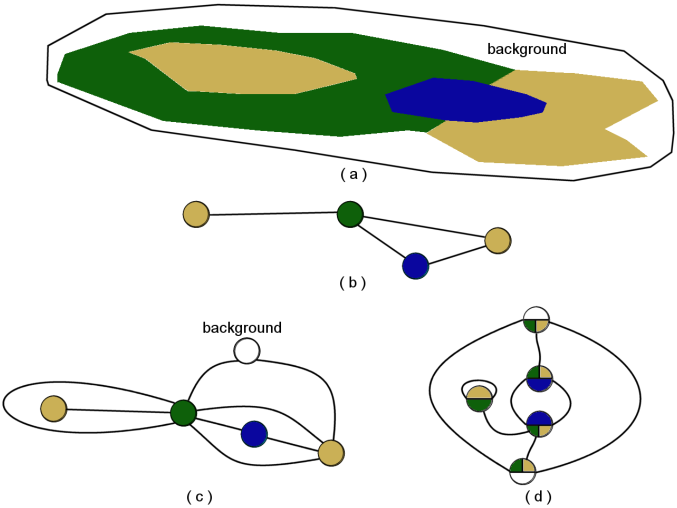


Fig. 1. a) Metric base map (green, blue and brown colored areas represent different surfaces); b) topological representation of a) using a simple graph; and c-d) topological representation of a) using a dual graph

complexity. On the other hand, topological maps are usually more compact, since their resolution is determined by the complexity of the environment. Thus, they allow fast path planning and provide more natural interfaces for human instructions. The principle of topological maps is to split the free-space of the real environment into a small number of regions.

Topological approaches usually represent the environment by using simple graphs [3,4]. Nodes in such graphs correspond to different environment regions and arcs indicate spatial relationships between them. Fig. 1 shows that simple graphs only take into account adjacency relationships, being unable to distinguish from the graph an adjacency relation from an inclusion relation between two regions. Besides, if there is two non-connected boundaries which will allow a robot to cross from one region to another one, the simple graph only joins these nodes by one arc. These limitations can be raised if dual graphs are employed because their structure is adapted to the processed data and they correctly encode the topology in 2D (see Fig. 1). In this paper, the dual graph data structure and the maximal independent edge set (MIES) decimation process are used to integrate the grid-based and the topological paradigms. The use of the dual graph allows to preserve the topology of the metric map and to correctly code the relation of adjacency and inclusion between topological regions.

The rest of this paper is organized as follows: Section 2 briefly reviews the metric and topological strategies for indoor environments mapping. Section 3 presents the proposed method. Experimental results revealing the efficacy of the method are described in Section 4. The paper concludes along with discussions and future work in Section 5.

2 Metric and Topological Paradigms for Mapping

Conventional approaches to SLAM rely on a metric, probabilistic representation of the robot pose and map. These metric approaches attempt to reconstruct the spatial distribution of the perceived environment, commonly in the form of a feature map or an occupancy grid. Although they have been successfully employed to map relatively large-sized environments, the main limitation of these techniques is related to the excessive computational complexity associated to these mapping processes. The classical alternative to metric maps is to model the environment using a topological map. Topological maps attempt to capture the spatial connectivity of the environment by representing it as a graph with arcs connecting the nodes that designate distinctive places in the environment [5]. These maps usually require reduced storage requirements, but such a representation usually lacks the necessary information to localize arbitrarily (can only localize to nodes in the topological graph) and to disambiguate similar topological regions. Besides, while probabilistic methods have been extensively investigated for performing inference over the space of metric maps, it is not the same case for topological maps. As a significant exception, the probabilistic topological maps (PTMs) [6] is a sample-based representation that approximates the posterior distribution over topologies given available sensor measurements. However, PTMs assume that the robot can detect whether it is near or on one of the nodes of the topological map. This can be considered a too restrictive assumption where the diversity of metric information is lost [7].

In order to deal with large, complex environments, the internal representation acquired by the robot can be organized as a hierarchy of maps which represent the whole environment at different levels of abstraction. Typically, these hierarchical representations consists of two layers: a metric map and a higher-level topological map. The hybrid approach usually attaches a local metric map to the nodes of a graph-based environment representation, where arcs represent coordinates transformation between nodes. Thus, the complexity can be bounded within each local map.

3 Hybrid Approach for Mapping Indoor Environments

3.1 Simple and Dual Graphs

Basically, a pyramid is a hierarchical structure which represents the contents of an input graph at multiple levels of abstraction. Each level of this hierarchy is at least defined by a set of vertices V_l connected by a set of edges E_l. These edges define the horizontal relationships of the pyramid and represent the neighbourhood of each vertex at the same level (intra-level edges). Another set of edges define the vertical relationships by connecting vertices between adjacent pyramid levels (inter-level edges). These inter-level edges establish a dependency relationship between each vertex of level l+1 and a set of vertices at level l (reduction window). The vertices belonging to one reduction window are the sons of the vertex which defines it. The value of each parent is computed from the set of values of its sons using a reduction function. Using this general framework, the procedure to build the level G_{l+1} from level G_l consists of three steps:

1. Selection of the vertices of G_{l+1} among V_l : This selection step is a decimation procedure and selected vertices V_{l+1} are called the surviving vertices.
2. Inter-level edges definition: Each vertex of G_l is linked to its parent vertex in G_{l+1}. This step defines a partition of V_l.
3. Intra-level edges definition: The set of edges E_{l+1} is obtained by defining the adjacency relationships between the vertices V_{l+1}.

The parent-son relationship defined by the reduction window may be extended by transitivity down to the base level. The set of sons of one vertex in the base level is named its receptive field.

A simple graph is a non-weighted and undirected graph containing no self-loops. In this hierarchy, the graph edges E_l represent adjacency relationships among pyramidal vertices of the level l. Simple graphs encode the adjacency between two vertices by only one edge, although their receptive fields may share several boundary segments. Therefore, a graph edge may thus encode a non-connected set of boundaries between the associated receptive fields. Moreover, the lack of self-loops in simple graphs does not allow to differentiate an adjacency relationship between two receptive fields from an inclusion relationship. These facts are shown in Fig. 1b, which represents the top of a simple graph pyramid encoding the connected components of Fig. 1a. In a dual graph pyramid, a level consists of a dual pair $(G_l, \tilde{G}_l)$ of planar graphs G_l and $\tilde{G}_l$. If level l defines a partition of the image into a connected subsets of pixels, then the vertices of G_l are the representatives of these subsets and the edges of G_l represent their neighborhood relationships. The edges of $\tilde{G}_l$ represent the boundaries of these connected subsets in level l and the vertices of $\tilde{G}_l$ define meeting points of boundary segments of $\tilde{G}_l$. Fig. 1c represents the top of a dual graph pyramid encoding the connected components of Fig. 1a. Fig. 1d shows the dual graph corresponding to 1c.

Within the dual graph pyramid framework, the set of edges that define the adjacency relationships among pyramidal vertices of the level $l+1$ is generated in two steps. First, the set of edges that connects each non-surviving vertex to its parent is contracted using a contraction kernel. A contraction kernel of a level l is the set of surviving vertices of l and the edges that connect each non-surviving vertex with its parent. The edge contraction operation collapses two adjacent vertices into one vertex, removing the edge between them. This operation may create redundant edges such as empty self-loops or double edges. The removal of these redundant edges constitutes the second step of the creation of the set of edges E_{l+1}. These redundant edges are characterized in the dual of the graph and removed by a set of edge removal kernels [10]. The key idea of the dual graphs is that a contraction in a graph implies a removal in its dual, and vice versa, in order to maintain the duality between the newly generated graphs. Thus, the generation of the edges in level $l + 1$ can be resumed as follows:

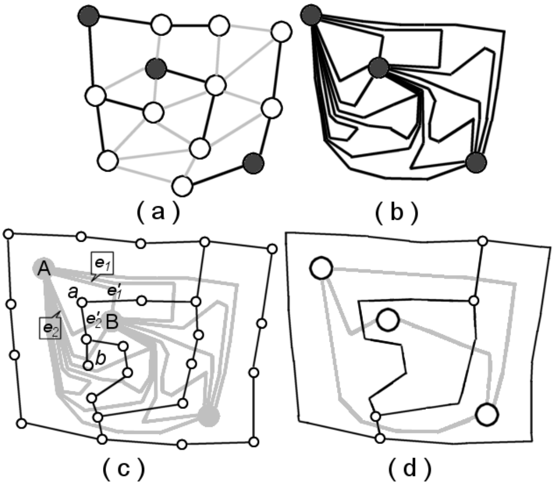


Fig. 2. Contraction and removal kernels: a) contraction kernel composed of three vertices (surviving vertices are marked in black); b) reduction performed by the equivalent contraction kernel in a); c) redundant edges characterization; and d) dual graph pair $(G, \tilde{G})$ after dual decimation step

1. Contraction of edges in G_l which connect non-surviving vertices with their parents. Removal of their corresponding edges in $\tilde{G}_l$. Fig. 2b shows the reduction performed by the contraction kernel in Fig. 2a.
2. Contraction of redundant edges in $\tilde{G}_l$ and removal of their corresponding edges in G_l. In Fig. 2c, the dual vertex a has a face defined by vertices **A** and **B**. The boundary between the regions defined by these vertices is artificially split by this dual vertex. Then, the two dual edges incident to this dual vertex (e'_1 and e'_2) can be contracted. The contraction of these dual edges has to be followed by the removal of one associated edge (e_1 or e_2) in order to maintain the duality between both graphs. In the same way, the dual vertex b encodes an adjacency relationship between two vertices contracted in the same vertex. This relationship can be removed by eliminating this direct self-loop and contracting the associated dual edge.

Using such a reduction scheme each edge in the reduced graph corresponds to one boundary between two regions. Moreover, inclusion relationships may be differentiated from adjacency ones in the dual graph.

3.2 Proposed Mapping Approach

The proposed algorithm is based on a dual graph pyramid that extracts the topological information from the metric map. In this work, the metric map is based on a two-dimensional occupancy grid, as originally proposed by Moravec and Elfes [8]. Thus, each grid cell (x, y) in the map yields the occupancy probability of the corresponding region of the environment. The algorithm works as follows:

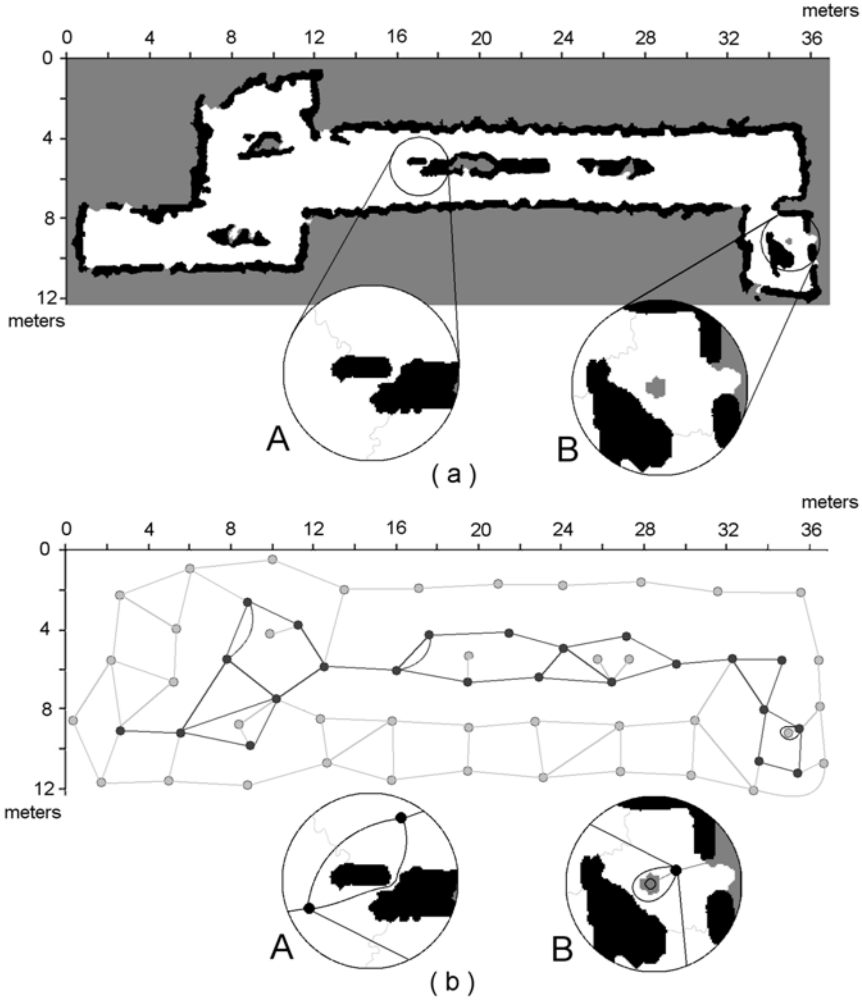
1. Metric map thresholding. Each occupancy value in the metric map is thresholded. Cells whose occupancy value is below threshold U_1 are considered free-space. Cells whose occupancy value are above U_1 and below threshold U_2 are considered non-explored. All other cells are considered occupied. Free-space and non-explored cells are the nodes of the base level of the graph pyramid. Two base nodes are related by an arc if the two corresponding cells are neighbors. To achieve planar embedding of the graph in the metric map we use the 4-vicinity. Each node and arc of the base graph is attributed. Nodes are attributed with a discrete name $w_n(n)$ which can only take two different values: free-space or non-explored. Arcs are attributed with the Euclidean distance $w_a(a)$ between the two metric cells they link.
2. Dual graph pyramid generation. To generate the pyramid, the algorithm proposed by Haxhimusa and Kropatsch [9] is used. Two nodes n and m can be contracted if $w_n(n) == w_n(m)$ and $w_a((n,m))$ is below a threshold U_d.
3. Set arc attributes of G_{l+1}. The attributes of those arcs $a_{l+1} \in G_{l+1}$ are updated with the maximum attribute of the arcs $a_l \in G_l$ that are contracted into a_{l+1}.

Using the dual graph pyramid framework, each node is linked to a connected set of nodes in the base level (receptive field). Besides, each arc between two nodes encodes a unique connected boundary between the associated receptive fields. Finally, the use of self-loops within the hierarchy allows to differentiate adjacency relationships between receptive fields from inclusion relations. Threshold U_d sets the maximum distance between two metric map places integrated into the same topological node. Thus, it defines the resolution of the topological representation.

4 Experimental Results

In order to evaluate the suitableness of the topological representation, Thrun proposes the assessing of three criteria: consistency, losing and efficiency [3]. In our case, the proposed topological map is always consistent with the metric map. In fact, each topological node is linked to a connected set of nodes in the base level. Hence, there exists a direct correspondence between the topological graph and the metric map. However, topological maps lack details and, therefore, paths found in the topological map may not be as optimal as paths found using the metric representation. Finally, when using topological maps, efficiency is traded off with consistency and performance loss.

The algorithm was tested on a Nomad 200 robot as part of a complete system for mapping unknown office environments. In this system, the robot attempts to space the place nodes in the map at equal intervals of 4 meters. Therefore, threshold U_d has been set to this value. Fig. 3a shows the thresholded metric map acquired by the robot. The highest level of the dual graph based topological representation in Fig. 3b shows the position of the places in global coordinates. It can be noted that the structure correctly represents the inclusion of a non-explored node into a free-space one. Besides, there are two situations where two connected nodes are joined by two different arcs. The map generation process



(a)

(b)

Fig. 3. a) Thresholded global grid-map of an office-like indoor environment (white-free space; black- obstacles and grey- non-explored); and b) highest level of the hierarchy of dual graph which encodes the topological representation (black nodes- free space and grey nodes- non-explored)

takes 200 msec. on a 844 MHz Pentium processor. The low computational time allows the on-line generation of the proposed map.

5 Conclusions and Future Work

This paper proposes an integrated method for indoor robot environment mapping. It combines the metric and topological paradigms. Topological maps are generated using a hierarchy of dual graphs that divides the metric map into homogeneous compact regions. The use of the dual graph allows to preserve the topology of the metric map and to correctly code the relation of adjacency and inclusion between topological regions.

Acknowledgments

This work has been partially granted by the Spanish Junta de Andalucía project P07-TIC-03106, the MICIN under project TIN2008-06196 (FEDER funds), and by the Austrian Science Fond under grants P18716-N13 and S9103-N13.

References

1. Borges, G.A., Aldon, M.J.: Optimal mobile robot pose estimation using geometrical maps. IEEE Trans. Robotics Automation 18, 87–94 (2002)
2. Lemaire, T., Berger, C., Jung, I., Lacroix, S.: Vision-based SLAM: stereo and monocular approaches. Int. Journal Computer Vision 74 (2007)
3. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence 99, 21–71 (1998)
4. Zivkovic, Z., Bakker, B., Kröse, B.: Hierarchical map building and planning based on graph partitioning. In: Proc. IEEE Int. Conf. Robotics and Automation (2006)
5. Košecká, J., Li, F., Yang, X.: Global localization and relative positioning based on scale-invariant keypoints. Robotics and Autonomous Systems 52, 27–38 (2007)
6. Ranganathan, A., Menegatti, E., Dellaert, F.: Bayesian inference in the space of topological maps. IEEE Trans. Robotics, 92–107 (2006)
7. Blanco, J.L., Fernández-Madrigal, J.A., González, J.: A new approach for large-scale localization and mapping: hybrid metric-topological SLAM. In: IEEE Int. Conf. Robotics Automation, pp. 2061–2067 (2007)
8. Moravec, H., Elfes, A.: High resolution maps from wide angle sonar. In: Proc. IEEE Int. Conf. Robotics and Automation, pp. 116–121 (1985)
9. Haxhimusa, Y., Kropatsch, W.G.: Segmentation graph hierachies. In: Proc. Joint IAPR Int. Workshop Syntactical Structural Pattern Recognition and Statistical Pattern Recognition, pp. 343–351 (2004)
10. Kropatsch, W.G.: From equivalent weighting functions to equivalent contraction kernels. In: Wenger, E., Dimitrov, L.I. (eds.) Digital Image Processing and Computer Graphics: Applications in Humanities and Natural Sciences, vol. 3346, pp. 310–320 (1998)

Integrating Graph-Based Vision Perception to Spoken Conversation in Human-Robot Interaction

Wendy Aguilar and Luis A. Pineda

Departamento de Ciencias de la Computación,
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Universidad Nacional Autónoma de México
Apdo. Postal 20-726, México, D.F., 01000

Abstract. In this paper we present the integration of graph-based visual perception to spoken conversation in human-robot interaction. The proposed architecture has a dialogue manager as the central component for the multimodal interaction, which directs the robot's behavior in terms of the intentions and actions associated to the conversational situations. We tested this ideas on a mobile robot programmed to act as a visitor's guide to our department of computer science.

1 Introduction

Human-robot interaction can be modeled in terms of flexible protocols focused on the expression and interpretation of intentions in interactive situations, and the execution of actions that satisfy such intentions. Such protocols can involve a range of modalities, like spoken language, pointing actions and vision for the input, for instance, and spoken language, the display of pictures and motor actions of physical devices (like mobile robots) for output. Conversational protocols may range from the very rigid and deterministic schemes involved in menu-based interaction to the rich and flexible patterns of natural language, like the ones exhibited in the so-called practical dialogues [1]. This work presents an integration of a graph-based vision perception module into a service robot with multimodal capabilities, that is based on the specification and interpretation of multimodal dialogue models, as presented in [2].

The proposed architecture was tested on a mobile robot, called Golem, that acts as the guide of a poster session about the research projects in our department of computer science. During the visit, the robot can interact with the user through its visual capabilities and recognize posters within the context of a spoken conversation in Spanish. Once a poster is recognized, the robot is able of explain it in spoken Spanish, with the support of other modalities like texts and graphics which are displayed on a screen. Similar multimodal vision and language projects are presented in [3], [4] and [5]; however, none of these include our notion of dialogue model specification and interpretation and use graph based algorithms for vision recognition.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 789–796, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The paper is organized as follows. Section 2 presents the proposed architecture and explains briefly each module, making emphasis on the integration and implementation of the vision module. Section 3 presents tests and results for the object recognition module and the system as a whole. Finally, the conclusions are presented in section 4.

2 Architecture

In this paper we propose to integrate vision to spoken conversation into a three level agent architecture where vision and language are interpreted through analogous processes (see Fig. 1(a)). The lower level is devoted to the recognition of modality specific information; the intermediate level assigns an interpretation to the images, either visual or linguistic, recovered by the speech and vision recognition systems, in terms of the expected intentions and the potential actions of the agent in the interpretation situation. In the third level, the multimodal conversational protocols (i.e. a set of conversational situations, with their associate intentions and actions) are represented in a modality independent fashion.

In this architecture perceptual modules have two components: 1) A modality specific recognition agent, and 2) a perceptual interpreter agent, see Fig. 1(b). The first component is responsible of sensing a visual or an acoustic signal and create its corresponding uninterpreted modality specific image. The perceptual interpretation agent proper assigns an interpretation to such image in terms of memory, where images with their corresponding interpretation are stored, and the current expected intentions of the agent, as specified in the dialogue model. The final output of the *Perceptual Interpretation Agent* is the most likely intention expected by the computational agent and expressed by the human-user, on the basis of the actual image or linguistic message, and the contents stored in memory.

2.1 Dialogue Manager Agent

The *Dialogue Manager agent* is the central component of this architecture. It is responsible of managing the conversation between the human user and the robot. The dialogue manager is a program interpreter that interprets objects called *dialogue models*. These specify conversational protocols in terms of conversational situations, which in turn are specified in terms of a set of the meaningful intentions that can plausible be expressed by the human-user in the situation (i.e. the expected intentions) with their associated actions. Dialogue models are represented as functional recursive transition networks [2], where nodes symbolize conversational or interactive situations (for example: *listening*, *telling* or *recursive*), and arcs are labeled by the intentions that need to be expressed to reach the corresponding situation (called *input speech acts*), and by the actions that are performed when such situations are reached (called *output rhetorical acts*), see Fig. 1(c). Arcs can be labeled with constants, grounded predicates or functions that have the current situation and the conversational history as their arguments,

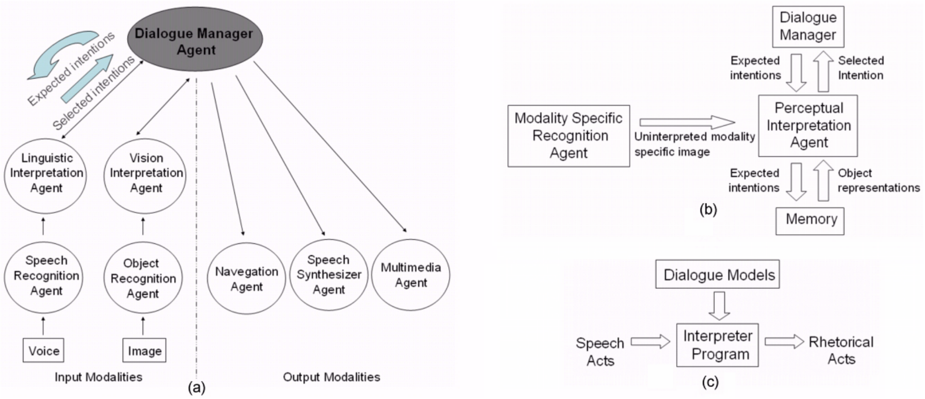


Fig. 1. (a) Agent's architecture, (b) General structure of an input modality module, (c) Dialogue Manager

and are evaluated on the fly when such arcs are traversed. Dialogue models are modular self contained objects, and are handled by the dialogue manger through a stack discipline. The interpretation of dialogue models process runs hand in hand with the interaction of the robot with the world.

2.2 Language Perception Module

The language perception module, in turn, assigns an interpretation to the text produced by the *Speech Recognition Agent*. For this, every expected intention in every interpretation situation has an associated regular expression, which states an ample collection of forms (complete and partial) through which such intention can be expressed by the human user. If the text matches the regular expression, the corresponding intention is selected, and its associated action is performed. In the case that none of the expected intentions can be matched, the system loads and executes a recovery dialogue model which may involve linguistic, visual and even motor behavior. The speech recognizer is an in house system built with *Sphinx*[1] and the Corpus DIMEX100 for Mexican Spanish, [6].

2.3 Vision Perception Module

Vision tasks include face recognition, gesture recognition, vision-based localization, and object recognition, for example. In this paper we focus in this latter task. There are four traditional approximations to object recognition (even though most of the systems use several of these techniques): 1) appearance based methods, 2) grammatical correspondence and graphs, 3) geometric based methods, and 4) local invariant feature correspondence. During the last years the approximation based on invariant local features has been widely accepted, [7][8].

---

[1] http://www.speech.cs.cmu.edu/

Here we explore the use of the combination of SIFT local feature descriptors [9], the Best Bin First (BBF) matching algorithm [10] and the Graph Transformation Matching (GTM) outlier removal algorithm [11], for the interactive object recognition problem on a mobile robot.

First, SIFT interest feature points are detected within an image. Each interest point detected is defined by its pixel position in the 2D image, a scale, an orientation, and a 128 feature vector describing the keypoint's surroundings. Known objects are defined in a database composed of object images and their corresponding keypoint descriptors and labels. The object recognition process consists in finding a correct match between keypoint descriptors of the actual robot view image and the known objects stored in the database. For this, a combination of the BBF and GTM algorithms was used. BBF is an algorithm that efficiently finds an approximate solution to the nearest neighbor search problem. A disadvantage of this algorithm, is that there is no spatial information taken into consideration (other than region similarity) for obtaining the nearest match. This causes the introduction of erroneous matches called outliers. The GTM algorithm was introduced here to reduce these outliers. GTM has a simple but effective conducting principle: iteratively eliminate correspondences

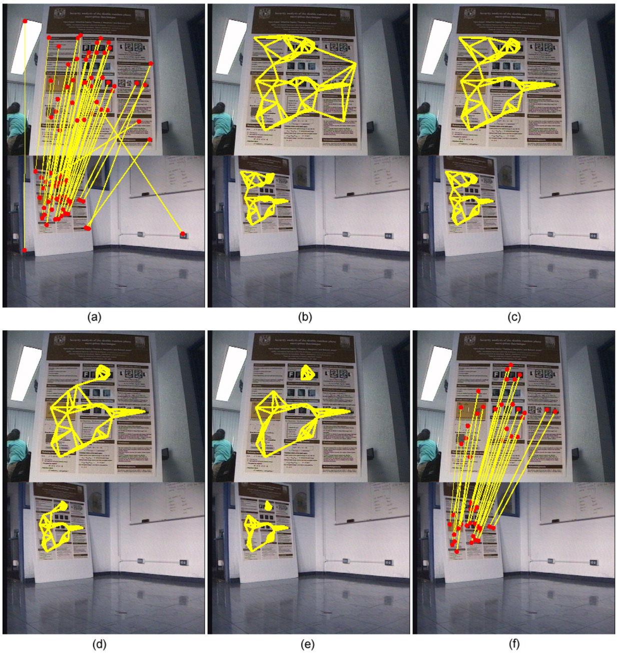


Fig. 2. Example of feature matching results for object recognition: a) BBF initial matching, b) GTM graphs at iteration 1, c) graphs at iteration 6, d) graphs at iteration 12, e) GTM final identical graphs at iteration 17, f) GTM final matching

that disrupt the neighborhood relationships. In order to do so, it constructs a K-nearest-neighbor (K-NN) graph for each view image (based on the space coordinates of detected feature points), and during each iteration it: a) removes the vertex (match) which most disrupt the similarity structure on both graphs, b) reconstructs the corresponding K-NN graphs and repeats this process until both graphs are identical, which means outliers have been removed [11]. Fig. 2.3(a) shows an example of the matching result from the BFF algorithm. Fig. 2.3(b)-(e) shows the graph transformation process through iterations 1, 6, 12 and 17. The final GTM matching (after outliers removal) is shown in Fig. 2.3(f).

As in the case of the language perception module, this module is responsible of trying to figure out what is the most probable object it is seeing, given the known objects knowledge, the actual robot view evidence and the expected intentions in the current dialogue model. This module is again composed of two agents: 1) *Object Recognition Agent*, and 2) *Vision Interpretation Agent*. The *Object Recognition Agent* takes the actual robot view, and describes it in terms of it's SIFT descriptors. The *Vision Interpretation Agent* uses this SIFT descriptors and tries to match them (using the BBF and GTM algorithms) just with the known objects in the list of expected intentions, which highly reduce the search space. The output of the *vision perception module* is the label of the recognized object. This label is passed back to the dialogue manager in order to start a conversation related to what the robot is seeing. If this module could not recognize any object, the dialogue manager starts a recovery conversation protocol.

2.4 Output Modality Agents

In addition to the input modality agents, the system supports a number of output modalities, which are associated to the actions performed in response to the interpretation of the human-user intentions. Actions are defined in multimodal rhetorical structures, which are defined in the dialogue models. Rhetorical structures (e.g. introduction, presentations, explanation, elaboration, etc.) are composed of basic rhetorical acts, that are modality specific actions, like pronouncing an utterance or displaying a text or an image; even the robot's motor behavior is specified as a basic rhetorical act. When a rhetorical act is performed all its basic acts are dispatched sequentially but performed simultaneously. Spoken acts are realized through a speech synthesizer. This agent is implemented using *Festival*[2]. There is also a multimedia agent that is responsible of displaying complementary visual information like text, images, animations or videos to the user. This agent is implemented in *Java* using the *Java Media Framework*[3]. This architecture was implemented using the Open Agent Architecture (OAA)[4] on a Magellan Pro robot - named *Golem*.

---

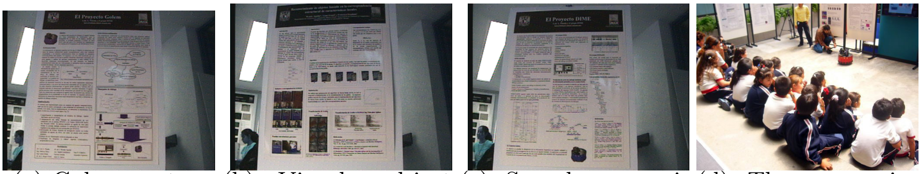
[2] http://www.cstr.ed.ac.uk/projects/festival

[3] http://java.sun.com/javase/technologies/desktop/media/jmf/index.jsp

[4] http://www.ai.sri.com/oaa/

3 Tests and Results

In the present application Golem performs as the guide of a posters session about the research projects in our computer science department. When a visitor arrives, the main dialogue is executed, and the robot starts the conversation by introducing himself and guiding the user through the possible conversational topics. There are two ways of selecting the topic of conversation: 1) ask the robot for a specific project through speech, or 2) show to the robot the poster of the project you are interested in. If the user chooses to interact through vision, then it has to take the selected poster and put it on a position where the robot can see it. For the second approach, an object database needs to be created. A frontal picture for each poster was taken with resolution of 320 × 240 pixels. No considerations were taken about illumination or position. The complete database is composed of 13 poster images, and its corresponding SIFT descriptors and labels. A subset of 3 poster images is shown in Fig. 3(a)-(c).

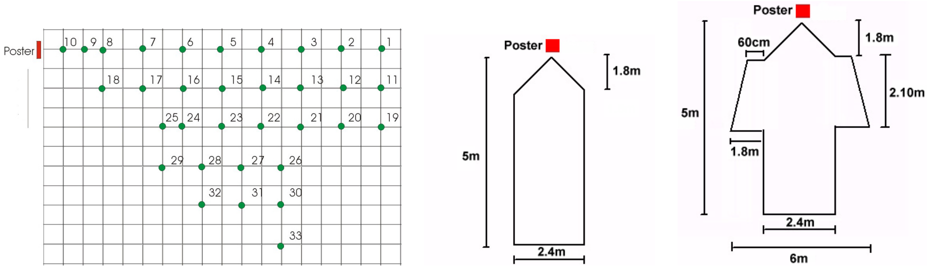


(a) Golem poster (b) Visual object recognition poster (c) Speech recognition project poster (d) The user interacting with the robot

Fig. 3. (a)-(c) A subset of the object image database, (d) Presentation of the robot Golem in a museum

The following is the English translation (from the original in Spanish) of a typical dialogue of the second kind:

GOLEM: *Please, show me the poster that you would like to talk about, and give me a second to recognize it.*
GOLEM: *Hey, it's my poster! Excellent choice.*
GOLEM: *About my poster I can tell you who are my programmers, how I work, or how am I physically. Are you interested in something like this, or do you prefer another poster?*
USER: *Tell me about how you are physically, please.*
GOLEM: *Physically I'am composed of 16 sonars, ... I can also talk to you about who are my programmers or how I work. Are you interested on these topics or do you prefer to talk about another poster?*
USER: *I'd better like to talk about another poster.*
GOLEM: *Ok. Show me the poster you are interested in so I can see it.*
GOLEM: *I cannot recognize this poster. I'm showing you on the screen what my actual view is. Please, move it to a better position.*
GOLEM: *Hey, I can see it now! It's the speech recognition poster ...*



(a) Experiment setting map. The size of each square is 30 × 30cm

(b) Safety zone with just one poster frontal view in the database

(c) Safely zone with 3 poster views: one frontal and two lateral views at 35 degrees

Fig. 4. Object recognition test map and results

A series of systematic experiments were performed to test the robustness of the object recognition module. Fig. 4(a) shows part of the laboratory where the guided visit takes place. All the 13 posters were set at the position marked with a rectangle (one at a time), and the robot was asked to recognize the poster on each of the 33 circles. This was performed with three different illumination conditions: 1) in the morning, 2) in the afternoon and 3) in the night with artificial light. The experiment was repeated changing the orientation of the poster from 0 to 360 degrees of rotation on the same plane. Figure 4(b) shows the results of this experiment when using just one frontal view of the posters in the database. The figure shows the zone where the robot recognized correctly the poster always. This test showed that the farthest recognition distance is 5m and the closest is 60cm. When two more views of the poster were added to the database (taken at 30cm away from the frontal view, one to the left and one to the right), the safety zone grew as shown in Fig. 4(c). About the execution times, the interaction between the user request for the explanation of a specific poster and the robot recognition answer is about 5 seconds.

This application has been demonstrated on more than 50 guided visits to the department, with children and adult users. It has also been taken outside the lab to be tested on talks about this project, where the illumination and sound conditions are natural and uncontrolled. Fig. 3(d) shows a picture of the presentation of this robot at the Museum of Science of Mexico City.

4 Conclusions and Future Work

This paper presented a successful integration of a graph-based perception module into a three level agent architecture centered on the dialogue. The architecture was implemented and tested on a service robot which acts as a visitor's guide of our department. Vision was used for interactive object recognition, where a graph transformation matching algorithm was used for feature matching.

Experiments and results of the object recognition module were presented and a safely recognition zone was defined. This application has been demonstrated on more than 50 guided visits to our department, with children and adult users. As future work we are incorporating new visual tasks, such as gesture recognition and visual-based localization.

References

1. Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A.: An architecture for a generic dialogue shell. Natural Language Engineering 6(34), 213–228 (2000)
2. Pineda, L.A.: Specification and interpretation of multimodal dialogue models. In: Sidorov, G. (ed.) Memorias del Workshop de robots de servicio, MICAI (2008)
3. Wachsmuth, S., Fink, G.A., Kummert, F., Sagerer, G.: Using speech in visual object recognition. In: Mustererkennung 2000, 22. DAGM-Symposium Kiel, Informatik Aktuell, pp. 428–435. Springer, Heidelberg (2000)
4. Saenko, K., Darrell, T.: Towards adaptive object recognition for situated human-computer interaction. In: Proceedings of the 2007 Workshop on Multimodal Interfaces in Semantic Interaction, pp. 43–46 (2007)
5. Rahmadi, K., Altab, H.M., Akio, N., Yoshinori, K.: Object recognition through human-robot interaction by speech. In: 13th IEEE International Workshop on Robot and Human Interactive Communication, RO-MAN, pp. 619–624 (2004)
6. Pineda, L.A., Villasenor, L., Cuétara, J., Castellanos, H., López, I.: Dimex100: A new phonetic and speech corpus for mexican spanish. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) IBERAMIA 2004. LNCS, vol. 3315, pp. 974–983. Springer, Heidelberg (2004)
7. Obdrzálek, J.M.: Object recognition methods based on transformation covariant features. In: XII European Signal Processing Conference EUSIPCO 2004, pp. 1333–1336 (2004)
8. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
9. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
10. Beis, J., Lowe, D.: Shape indexing using approximate nearest-neighbour search in highdimensional spaces. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 1000–1006 (1997)
11. Aguilar, W., Frauel, Y., Escolano, F., Pérez, M.M., Espinosa-Romero, A., Lozano, M.: A robust graph transformation matching for non-rigid registration. Image and Vision Computing (2008), doi:10.1016/j.imavis.2008.05.004

From Vision Sensor to Actuators, Spike Based Robot Control through Address-Event-Representation

A. Jimenez-Fernandez, C. Lujan-Martinez, R. Paz-Vicente, A. Linares-Barranco,
G. Jimenez, and A. Civit[1]

Departamento de Arquitectura y Tecnología de Computadores,
Universidad de Sevilla,
Av. Reina Mercedes s/n, 41012-Sevilla, Spain
angel@us.es

Abstract. One field of the neuroscience is the neuroinformatic whose aim is to develop auto-reconfigurable systems that mimic the human body and brain. In this paper we present a neuro-inspired spike based mobile robot. From commercial cheap vision sensors converted into spike information, through spike filtering for object recognition, to spike based motor control models. A two wheel mobile robot powered by DC motors can be autonomously controlled to follow a line drown in the floor. This spike system has been developed around the well-known Address-Event-Representation mechanism to communicate the different neuro-inspired layers of the system. RTC lab has developed all the components presented in this work, from the vision sensor, to the robot platform and the FPGA based platforms for AER processing.

1 Introduction

Humanoids are robots that mimic the human body, emulate the behaviour of human movements and try to reproduce the human mental algorithms. Neuromorphic engineers work in the study, design and development of neuro-inspired artefacts developed with artificial mechanism, like VLSI chips for sensors[5][6], neuro-inspired processing, filtering or learning [9][10][11][12], neuro-inspired control-pattern-generators (CPG) [13][14][15], neuro-inspired robotics [16] and so on.

One of the problems that neuromorphic VLSI engineers had to face was the viability to implement several thousand of neuro-inspired cells (spiking neurons) in a chip, and be able to communicate them with a few available pins. This matter was solved multiplexing in time the neurons behaviour using a very high speed digital bus. The solution was called Address-Event-Representation (AER). AER was proposed by the Mead lab in 1991 [1] for communicating between neuromorphic chips with spikes (Figure 1). Each time a cell on a sender device generates a spike, it communicates with the array periphery and a digital word representing a code or address for that pixel is placed on the external inter-chip digital bus (the AER bus). In the receiver chip, spikes are directed to the pixels whose code or address was on the bus. In this way, cells with

---

[1] This work was supported by Spanish grant SAMANTA 2 (TEC2006-11730-C03-02), and by the Andalusia Council with the BrainSystem project (P06-TIC-01417).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 797–804, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the same address in the emitter and receiver chips are virtually connected by streams of spikes. Cells that are more active access the bus more frequently than those less active. There is a growing community of AER protocol users for bio-inspired applications in vision, audition systems, and robot control, as demonstrated by the success in the last years of the AER group at the Neuromorphic Engineering Workshop series [3]. The goal of this community is to build large multichip and multi-layer hierarchically structured systems capable of performing massively-parallel data-driven processing in real time [2][4][8].

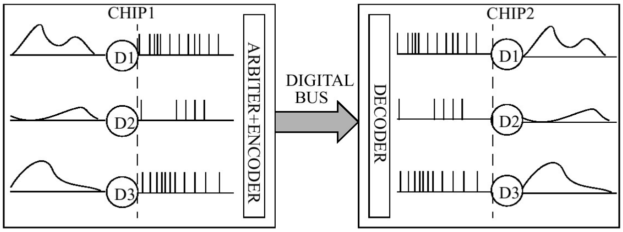


Fig. 1. Rate-coded AER inter-chip communication scheme

2 Neuro-inspired Sensing, Filtering and Actuating Model

In this paper we describe a neuro-inspired mobile robot with a double spike-based control mechanism for two DC motors. This control model works with pulse frequency modulation (PFM), just like the spikes, to power directly the actuators with spikes. Visual information comes from a spike-based sensor to the control model through a spike filter, extracting the direction and speed references for the robot control layer. The spike filter has the aim to obtain the geometric centre of the object detected by the vision sensor as fast as possible. This filter only requires several events for accurate results. The vision sensor converts the analogue luminosity of the CCD camera pixels into spikes after applying a pre-processing of the visual information, extracting the temporal difference information of the sensor. The mobile robot has been constructed around reconfigurable components (FPGA and microcontrollers). Two AER-tools [25] have been used: the USB-AER, for the visual information pre-processing and spike conversion, and the AER-robot, for the spike filtering and the spike based motor control operation for the two wheels of the robot. Figure 2 shows system blocks. A commercial low cost CCD is sending the information in RAW format to a FPGA through an ADC converter. These frames are pre-processed to extract temporal differences and converted into AER information by the Exhaustive AER generator [24] using the USB-AER tool [25]. The complete neuro-inspired implementation allows processing the information with very low time delays compared with the delay of the motor dynamic changes. Furthermore, by changing our low-cost AER-visual sensor by an AER retina, all the processing will be no-frame dependant, compared with the completely digital visual processing that has to process the complete frame before extracting an order for the mobile platform.

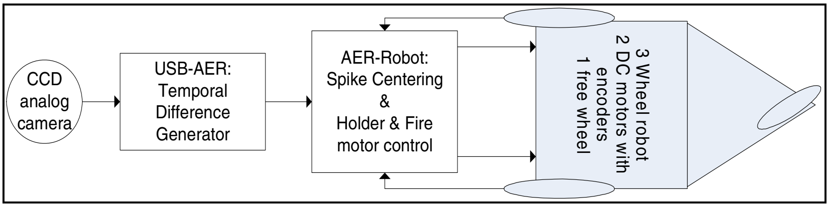

CCD analog camera
USB-AER: Temporal Difference Generator
AER-Robot: Spike Centering & Holder & Fire motor control
3 Wheel robot
2 DC motors with encoders
1 free wheel

Fig. 2. Neuro-inspired system for mobile robot control

3 Visual Stage

AER image sensors are devices that capture moving images from real world and convert image information to spiking events, according to the AER neuromorphic approach. Different AER retinas are developed using CMOS based ASIC chips as image sensor that directly generates AER streams [5]. In this paper we propose a cheaper alternative image sensor based in a traditional image video camera (CCD sensor with composite video) and a FPGA board that captures frame information from a video composite signal generated by a standard web cam or camcorder, and convert this frame information in an AER stream. Main advantage of Synthetic Retina is for AER chains testing and debugging. As AER events production is controlled by an FPGA based design, written in VHDL, the behavioral of the retina can be easily modified by reprogramming the FPGA circuits.

As Synthetic Retina uses an inexpensive Xilinx Spartan II FPGA and an inexpensive CMOS web cam, it is presented also as a cheap alternative to traditional ASIC AER retinas for AER testing when these retinas are not available .Synthetic Retina has some limitations derived from video composite signal limitation and sensor used. Traditional video sensor has an automatic gain control built-in, so image brightness is modified by video sensor for accommodating to lighting conditions. Therefore, AER generated does not describe exactly captured image, but a bright-compensated version of it. Also maximum change speed is limited by frame-time period. Using high speed video camera can improve this limitation.

3.1 Brightness Retina and Contrast Retina

Two different kind of AER retinas are commonly used in AER chains: brightness, and contrast retinas. Both of them operate in a very similar way, so image is captured and converted to AER events. Brightness retina [24] generates an AER stream describing captured information. Frequency of event for each pixel is a function of the amount of light received by this pixel sensor. For that, last captured frame is maintained on memory and used to generate events. In Derivative retinas [5], AER information does not describe image brightness, but image brightness changes for each pixel. An invariant luminosity pixel does not generate any events. High events activity is present when pixel evolves from dark to light or opposite. For that, Synthetic Retina maintains two copies of captured frames instead of just one. One copy stores last frame, and the other the subtraction of this last frame with previous one. Second frame buffer can otherwise store previous frame, and subtraction is computed at AER generation process. Result of this subtraction produces two sign results, positives ones when a pixel evolves from dark to bright, and negatives ones when opposite, a brilliant pixel darkens. Derivative retinas can be useful because the AER traffic generated is lower than Brightness retina. Only those parts of the images that changes generates AER events.

3.2 Synthetic Retina Implementation

Figure 3 diagram describes hardware implementation blocks in VHDL. An analogue to digital converter (ADC) is used for digitalizing composite video signal. Actually all known AER retina works with gray-levels so we don't need to decode colour subcarrier. A VHDL module waits for vertical and horizontal syncs to capture video information. An arbiter is used to share external memory access between video captured and AER events generation. As shown in Figure 3, external RAM is divided in two different frame buffers, one storing last frame information, and the other that stores image difference between last frame and previous one. Choosing which frame is used to feed the AER generation, we obtain a Brightness or Derivative Retina.

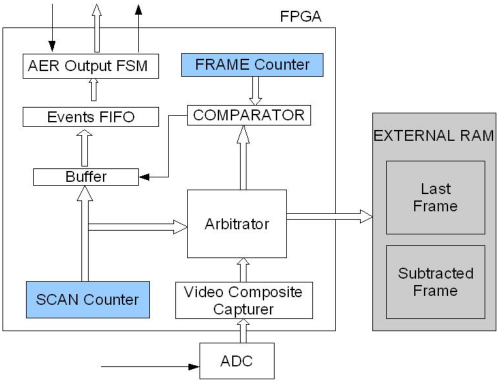


Fig. 3. Synthetic retina block diagram

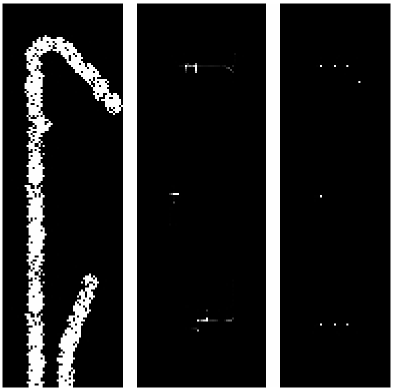


Fig. 4. Centre from events algorithm applied to a path

To generate AER events, exhaustive method has been implemented [23]. A binary counter is used to scan captured frames and generate events for each pixel if comparison between pixel value and slice counter gives that slice counter is below pixel value. Slice counter divides a frame time in slices. To transmit pixel information of 255 (maximum allowed value), 255 events needs to be transferred, and so all slots are occupied. A pixel with a value of 100 only needs to transfer 100 events of the 255 given slots. Bit order in slice counter is reversed to avoid event concentration in the first slices of each frame. AER generation method can be modified in order to test other method of Frame-to-AER conversion [22][23].

4 Centre Detection and Movement Information Translation

In order to command the robot, for line tracking, there is need to determine line centre from an AER synthetic retina generated events. The idea is to calculate locations, nearest as possible to centre of an object in motion or a path, which let to infer the object's location or path's direction. Centring from events is based on these ideas: only active information is present in the AER bus, transmitting the cell addresses allows performing extra operations on the events while they travel from one chip to another and operations must consume a short period of time.

Highest and lowest values for each coordinate are calculated at each event arrival from its address. Then, centre is inferred dividing by 2 (shifting right) the addition of the corresponding limits from each coordinate. Periodically, these limits values are forgetting, making them to achieve their initial values progressively if no event is received for a while. Forgetfulness is defined by the input event rate. A higher one means more information, possibly redundant, so forgetfulness can be applied more frequently. This guarantees that object in motion does not saturate the limits, which would produce bad centres at the output. The inferred address is always contained in the object in motion area; as only contrast change information is received. In figure 4 experimental results is shown after applying cantering algorithm. The first column represents the frames. The second one represents each inferred centre (on each event arrival). Finally, he third one shows sent centres as outputs. The inferred centre is the desirable position to achieve from the robot reference position. Y-difference from the robot reference one to the y-centre coordinate is used to increase or decrease the speed. If this difference is so far, speed can be increased. The situation is under control when tracking a line, or object motion is going faster than the robot and it may move faster. On the other hand, if it is short, speed may be decreased. It could be a closed, or hairpin, bend or the robot is approaching to the object. Turn sense and its intensity are defined by the difference of x-centre coordinate and the reference position. Turn direction is defined by the difference's sign and turn intensity depends on the absolute value of it. The AER DC Motor Control stage receives exciting and inhibitory spikes for both motor turn and speed. Spike frequency controls intensity for each one, from 1Kevent/s to 260Kevent/s, and exciting and inhibitory spikes mean sense, coded as positives and negatives addresses. Spike output is generated by a synthetic random generator [17][23]. Y-difference and X-difference are mapped to AER DC motor control stage frequency (from 1Kevent/s to 256Kevent/s) and corresponding address event, turn sense and speed, are sent.

5 Motor Control Stage

In biology, one of the last steps of the information processing is to drive a limb. Trying to mimic the biology, we use spikes to drive actuators in robotic platforms, for example, driving DC-motors with spikes. We have developed closed-loop speed control for DC-motors, and we have used them to control a mobile robot. All these controls has been implemented in a FPGA and tested in the AER-Robot board [9]. Digital systems usually drive DC-motors using a Pulse Width Modulation (PWM). PWM signals have a fixed period, and variable high time or duty-cycle [18]. In the neuronal model presented, spikes are codified following a Pulse Frequency Modulation (PFM) where information resides in the pulse frequency. In a PFM scheme the pulse's high time is fixed, and the frequency variable, just opposite to a PWM scheme. We use the PFM as other way to drive DC-motors, applying spikes almost directly to the DC-motor. Spikes are so short in time (e.g. 20ns) that they will be filtered by the DC-motor. The solution is to increase them in time (e.g. 1us) for a fixed time. As we can found in [18], the spike width affects to the DC-motor static gain. The longer spike width is the higher static gain will be.

Table 1. *Robot Features*

Parameter	Value	Units
Motor nominal voltage	12	V
Motor start current	4.8	A
Max motor ang. Speed	133.5	r.p.s.
Motor rise time	17.6	ms
Gear box relation	1:19	n.a.
Max. gear box speed	7.02	r.p.s.
Encoded channels	3	Ch
Encoder counts per turn	500	p.p.r.
Max. encoder freq.	100	kHz
Max. ref. spikes rate	263	kSp/s

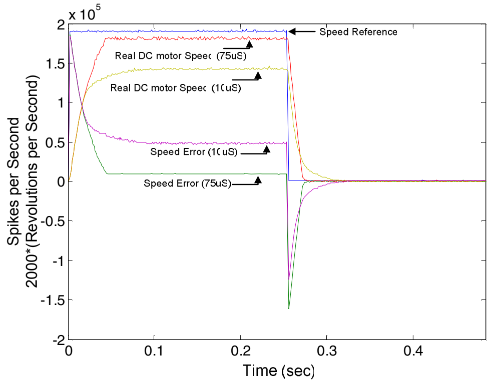


Fig. 5. Closed-loop spiking-speed control response for diverses spikes width

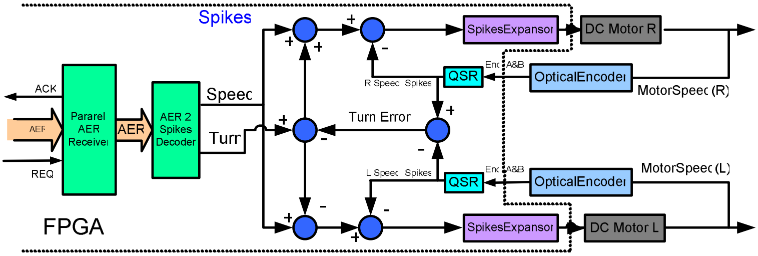


Fig. 6. Control blocks for Eddie

5.1 Closed-Loop Spiking Speed Control

Another step to drive DC-motors with spikes is to implement a closed-loop control, like a speed control. For this purpose there is needed an element that subtracts two spiking signals, the speed reference and the feedback information, the real DC-motor speed, being this element output the error between the desired speed and the real speed. This is the function of the Spike Hold and Fire (SH&F) component. The SH&F subtracts in real-time two incoming spike streams (U and Y) being the output a spike stream whose spike rate is the difference between the U and Y spike rate. Another question is how to measure the feedback information (DC-motor speed), and translate it to a spike stream. In our case, we have used an optical encoder, which outputs are two signals (A&B). These signals frequency represents the DC-motor speed, following just a PFM scheme, and the phase between A&B, the speed direction. To translate them to spikes we use a finite state machine (FSM) that determines the speed direction (sign). Then it fires a signed spike for every edge in the A&B signals.

Using the AER-Robot board and an USB2AER monitor/sequencer [20], we have measured the motor responses. Inside the AER-Robot lies the VHDL components that implements the closed-loop control: an AER decoder (for the references from the AER input port), a SH&F (who gives the error), a spikes expansor (to apply the spikes to the DC-motor), and a massive spikes monitor (who gives an address to every spikes, and send them through an AER parallel port). Connected to these ports there is an USB2AER monitor/sequencer, which generates the AER events for speed reference, and sends the spikes information to a PC, through a full speed USB2.0 port. In Figure 5 we can see the DC-motor speed response when we apply a constant spike stream, just like a step, for two different spikes width. We show the information recovered from the AER bus, after integrating the spikes for a fixed time. In the plot there are 5 signals, the two first one is the speed reference, with a constant spike rate. The next two ones are the DC-motors speed for each spike width, read form the optical encoder's spikes. And finally there are the speed errors, which start with a high spike rate, and start to decrease when the motor speeds are increasing. As a typical closed-loop control system, we can see both static gains below one, which introduces the static errors. We have also decreased the rise time, from 17.6ms to 0.73ms, and achieve a low ripple (including the oscillation introduced by the spikes integration).

5.2 Eddie, the Robotic Platform

Finally, we have tested these controls in a robotic platform, Eddie. This platform is a differential robot, with two independent DC-motors in the rear, and a free wheel in the front [19]. The control have been implemented in VHDL and loaded in the AER-Robot board. In Figure 6 we can find control blocks for Eddie. Where inputs are: the speed and turn references, and both motors speed feedback. Outputs are the spikes applied to the DC-motors. For this application, the input spikes will come from a parallel AER port, and it will be send to a reference input (speed or turn) after decode them (from an address event to a simple spike). Figure 6 also shows the control loops for each motor. Spikes references flows from left to right, the turn reference are added, or subtracted, to the speed reference (using two SH&F), obtaining the individual speed reference for each DC-motor. Both individual references are subtracted to DC-motor real speed; this is the individual speed error. Finally, the speed error spikes, are increased in time, and applied to the DC-motor. We also measure the turn speed (the difference between both motors speed) and implement a third control loop for the turn. This one helps to keep the same speed for both motors.

6 Conclusions

Joining the three AER layers (sensing, centre filtering and actuating) it is possible to follow a path drawn on the floor using a mobile robot. The speed response of the whole system depends on the motors dynamic response, because the AER system is able to change the motor commands with a few events (some microseconds). The principal advantage of this control model is that no frame processing is required to detect the line and then act to the motors, while the actuation to the motors is done in real time while the visual information is received and processing during the transmission.

References

[1] Sivilotti, M.: Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks, Ph.D. Thesis, California Institute of Technology, Pasadena CA (1991)

[2] Serrano-Gotarredona, T., et al.: AER Image Filtering Architecture for Vision-Processing Systems. IEEE Transactions on Circuits and Systems. Fundamental Theory and Applications 46(9) (September 1999)

[3] Cohen, A., et al.: Report to the National Science Foundation: Workshop on Neuromorphic Engineering, Telluride, Colorado, USA, June-July (2004), `http://www.ini.unizh.ch/telluride`

[4] Mahowald, M.: VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function. PhD. Thesis, California Institute of Technology Pasadena, California (1992)

[5] Lichtsteiner, P., Posh, C., Delbruck, T.: A 128X128 120dB 15 us Asynchronous Temporal Contrast Vision Sensor. IEEE Journal on Solid-State Circuits 43(2), 566–576 (2008)

[6] Chan, V., Liu, S.C., van Schaik, A.: AER EAR: A Matched Silicon Cochlea Pair with Address-Event-Representation Interface. IEEE Transactions on Circuits and Systems-I 54(1), 48–59 (2007)

[7] Linares-Barranco, A., et al.: AER Neuro-Inspired interface to Anthropomorphic Robotic Hand. In: IEEE World Conference on Computational Intelligence, IJCNN, Vancouver (July 2006)

[8] Serrano-Gotarredona, R., et al.: AER Building Blocks for Multi-Layer Multi-Chip Neuromorphic Vision Systems. In: NIPS 2005 (2005)

[9] Serrano-Gotarredona, R., et al.: On Real-Time AER 2-D Convolutions Hardware for Neuromorphic Spike-Based Cortical Processing. IEEE Transactions on Neural Networks 19(7), 1196–1219 (2008)

[10] Serrano-Gotarredona, R., et al.: A Neuromorphic Cortical-Layer Microchip for Spike-Based Event Processing Vision Systems. IEEE Transactions on Circuits and Systems-I 53(12), 2548–2566 (2006)

[11] Hafliger, P.: Adaptive WTA with an Analog VLSI Neuromorphic Learning Chip. IEEE Transactions on Neural Networks 18(2), 551–572 (2007)

[12] Indiveri, G., et al.: A VLSI Array of Low-Power Spiking Neurons and Bistables Synapses with Spike-Timig Dependant Plasticity. IEEE Transactions on Neural Networks 17(1), 211–221 (2006)

[13] Gomez-Rodríguez, F., et al.: AER Auditory Filtering and CPG for Robot Control. In: ISCAS 2007 (2007)

[14] Vogelstein, R.J., et al.: Phase-Dependant Effects of Spinal Cord Stimulation on Locomotor Activity. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14(3), 257–265 (2006)

[15] Vogelstein, R.J., et al.: Dinamic Control of Spinal Locomotion Circuits. In: ISCAS 2007 (2007)

[16] Linares-Barranco, A., Gomez-Rodriguez, F., Jimenez-Fernandez, A., Delbruck, T., Lichtensteiner, P.: Using FPGA for visuo-motor control with a silicon retian and a humanoid robot. In: ISCAS 2007 (2007)

[17] Linares-Barranco, A., et al.: Inter-spike-intervals analysis of AER Poisson-like generator hardware. Neurocomputing (2007)

[18] Jiménez-Fernández, A., et al.: Civit. AER and dynamic systems co-simulation over Simulink with Xilinx System Generator. In: ICECS 2008 (2008)

[19] Jiménez-Fernández, A., et al.: AER-based robotic closed-loop control system. In: ISCAS 2008 (2008)

[20] Berner, R., et al.: A 5 Meps $100 USB2.0 Address-Event Monitor-Sequencer Interface. In: ISCAS 2007 (2007)

[21] Gomez-Rodriguez, F., et al.: Two Hardware Implementation of the Exhaustive Synthetic Aer Generation Method. LNCS. Springer, Heidelberg (2005)

[22] Gomez-Rodriguez, F., et al.: AER tools for communications and debugging. In: ISCAS 2006 (2006)

[23] Linares-Barranco, A., et al.: On Algorithmic Rate-Coded AER Generaetion. IEEE Transaction on Neural Network 17(3), 771–788 (2006)

[24] Culurciello, E., et al.: A biomorphic digital image sensor. IEEE Journal of Solid-State Circuits 38, 204–281 (2003)

Automatic Generation of Biped Walk Behavior Using Genetic Algorithms

Hugo Picado[1,2], Marcos Gestal[3], Nuno Lau[1,2], Luis P. Reis[4,5],
and Ana M. Tomé[1,2]

[1] Institute of Electronics and Telematics Enginnering of Aveiro, Portugal
[2] Dep. of Electronics, Telecommunications and Informatics, Univ. Aveiro, Portugal
[3] Artificial Neural Network and Adaptive System Lab., Univ. Coruña, Spain
[4] Artificial Intelligence and Computer Science Lab., Univ. Porto, Portugal
[5] Faculty of Engineering of the University of Porto, Portugal
`hugopicado@ua.pt`, `mgestal@udc.es`, `nunolau@ua.pt`, `lpreis@fe.up.pt`,
`ana@ua.pt`

Abstract. Controlling a biped robot with several degrees of freedom is a challenging task that takes the attention of several researchers in the fields of biology, physics, electronics, computer science and mechanics. For a humanoid robot to perform in complex environments, fast, stable and adaptive behaviors are required. This paper proposes a solution for automatic generation of a walking gait using genetic algorithms (GA). A method based on partial Fourier series was developed for joint trajectory planning. GAs were then used for offline generation of the parameters that define the gait. GAs proved to be a powerful method for automatic generation of humanoid behaviors resulting on a walk forward velocity of 0.51m/s which is a good result considering the results of the three best teams of RoboCup 3D simulation league for the same movement.

Keywords: biped, locomotion, genetic algorithms, humanoid, robotics.

1 Introduction

For a long time, wheeled robots were used for research and development in the field of Artificial Intelligence and Robotics [1]. However, wheeled locomotion is not adapted to many human environments [2]. This increased the interest in biped locomotion and especially in humanoid robotics. Biped locomotion control is a complex problem to solve because the goal is to achieve stability counting on a small support area considering the several degrees of freedom of a biped. The most common approach consists of finding a set kinematics trajectories and using stabilization criteria to ensure that the generated gait is stable. In this paper, the Center of Mass (CoM) was used to monitor the quality of the gait. If the CoM remains inside the support polygon (the convex hull of the contact points of the feet with the ground), the walk is considered statically stable. Other methods may be used as stabilization criteria. The most popular is the Zero Moment Point (ZMP), (Vukobratovic, 1972) [3], defined as the point on the ground about

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 805–812, 2009.
© Springer-Verlag Berlin Heidelberg 2009

which the sum of the moments of all the active forces equals zero. If ZMP remains inside the support polygon, the gait is considered dynamically stable. ZMP considers the kinematics and the kinetics of the system, which requires the computation of complex dynamic equations that are computationally expensive. This paper presents a method based on GA for developing of efficient and robust humanoid behaviors tested in a simulated version of the humanoid robot NAO [4] using the Simspark [5] simulation environment. The remainder of this paper is organized in five more sections. Section 2 defines how the gait was defined and the method developed for joint trajectory planning, section 3 briefly describes the GAs, section 4 defines the configuration of the GA used for the tests, which is followed by the experimental results, presented in section 5. Section 6 presents some conclusions and interesting proposals for future work.

2 The Walking Gait

Some human-like movements are inherently periodic and repeat the same set of steps several times (e.g. walk, turn, etc). The principle of PFS consists of the decomposition of a periodic function into a sum of simple oscillators as represented by the following expression:

$$f(t) = C + \sum_{n=1}^{N} A_n \sin\left(n\frac{2\pi}{T}t + \phi_n\right), \forall t \in \Re \tag{1}$$

where N is the number of frequencies, C is the offset, $A_{n=1..N}$ are amplitudes, T is the period and $\phi_{n=1..N}$ are phases.

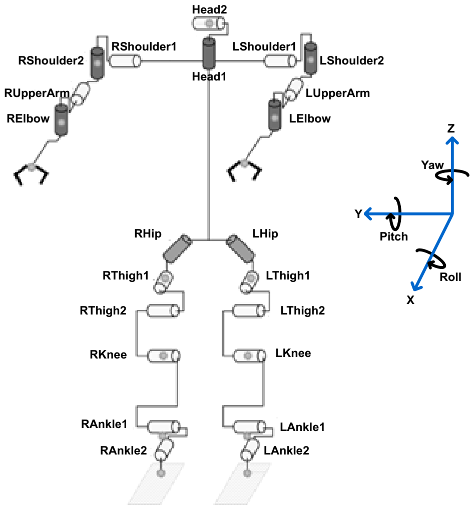


Fig. 1. Developed behaviors: Humanoid structure. Adapted from [4].

Applying these osillators to each joint, a walking gait was developed and the tests were performed with the simulated humanoid NAO in the scope of the RoboCup 3D soccer simulation league using the Simspark Simulation Environment [5]. Figure 1 shows the humanoid structure and the referential axis considered. The figure also shows the referential considered in the experiments. The main idea behind the definition of this gait is to place an oscillator on each joint we pretend to move in order to define its trajectory. The oscillators are placed on the following joints: LShoulder1, RShoulder1, LThigh1, RThigh1, LThigh2, RThigh2, LKnee, RKnee, LAnkle1, RAnkle1, LAnkle2 and RAnkle2. Hence, 12 single-frequency oscillators are used. Since each single-frequency oscillator will have 4 parameters to define, 48 parameters are needed to completely define the gait. It is common to assume a walk sagittal symmetry, which determines the same movements for corresponding left and right sided joints with a half-period phase shift. Hence, it is possible to reduce the number of parameters by half of the original size, resulting on 24 parameters. Additionally, the period of all oscillators should be the same to keep all the joints synchronized by a single frequency clock. This consideration reduces the number of parameters to 19. A set of equations can be obtained for the left-sided joints:

$$f_{LShoulder1}(t) = C_1 + A_1 \sin(2\pi t/T + \phi_1) \tag{2}$$

$$f_{LThigh1}(t) = C_2 + A_2 \sin(2\pi t/T + \phi_2) \tag{3}$$

$$f_{LThigh2}(t) = C_3 + A_3 \sin(2\pi t/T + \phi_3) \tag{4}$$

$$f_{LKnee}(t) = C_4 + A_4 \sin(2\pi t/T + \phi_4) \tag{5}$$

$$f_{LAnkle1}(t) = C_5 + A_5 \sin(2\pi t/T + \phi_5) \tag{6}$$

$$f_{LAnkle2}(t) = C_6 + A_6 \sin(2\pi t/T + \phi_6) \tag{7}$$

where $f_X(t)$ is the trajectory equation for the joint X, $A_{i=1..6}$ are amplitudes, T is the period, $\phi_{i=1..6}$ are phases and $C_{i=1..6}$ are offsets. The right-sided joints can be obtained with no additional parameters: For roll joints the left and the right side perform the same trajectories over the time. For pitch joints, the right side can be obtained by adding a phase, π, on the corresponding oscillator. The unknown parameters together form the genome that will be used by the genetic algorithm to generate the gait.

3 Genetic Algorithms

This paper presents an approach based on Genetic Algorithms (GA) [6] for modeling the previously described behavior. GAs were developed following ideas and techniques from genetics and natural selection theories [7]. After generation of an initial population of individuals, and according to the principle of survival of the fittest, they generate the next population by transmitting their genes through several operations. Each individual in this population represents a possible solution to the problem, and each one is represented by a chromosome, which is a strand of numerical values or genes. There are three major operations in a GA (summarized in Figure 2):

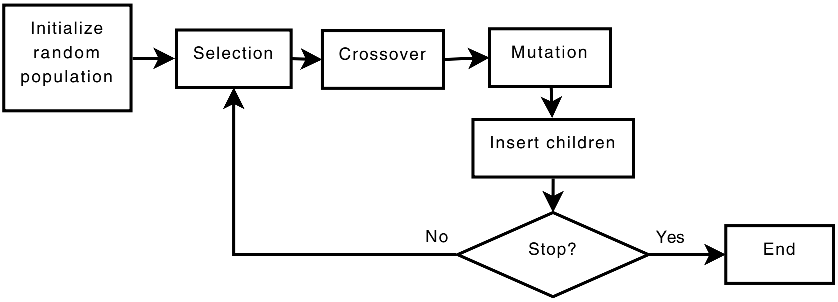


Fig. 2. General schema for a genetic algorithm

i Selection: chooses some parents for crossover according to predefined rules (cost function or fitness).

ii Crossover: generates offspring from parents, by exchanging some genes according to different schemas: one-point, two-point, uniform, etc. The offspring thus inherits some characteristics from each parent. Although there are several ways to get a new population, the offspring will usually replace an individual of the actual population providing it has a similar fitness. Another option consists on the use of an auxiliary population which will replace the actual one after it is full of individuals.

iii Mutation generates offspring by randomly changing one or several genes in an individual. It allows searching for new regions of solutions which, otherwise, would not be explored. Mutation, therefore, avoids the GA to focus only on a local search which, in turn, increases the probability of finding global optima.

A standard GA proceeds as follows: an initial population of individuals is generated at random. The individuals in the current population are evaluated according to some predefined quality criterion (fitness function). To form a new population (the next generation), individuals are selected according to their fitness. Then some or all of the existing members of the current population are replaced with the newly created members. Creation of new members is done by different operations (crossover, mutation), which (hopefully) should make the new individuals better than the old ones. If the GA has been well designed, the population will converge to an optimal solution to the problem. Finally, two pragmatic criteria are generally used to stop the GA either when a given number of generations passed away or when a given performance error is reached (both set by the user).

4 GA Configuration

The parameters described in Section 2 were defined by a GA using the GADS toolbox for Matlab [8]. The algorithm creates an initial population of 100 chromosomes initialized randomly. The roulette method used for selection consists of simulating a roulette-wheel where the parents are selected with a probability that is proportional to their fitness. The mutation follows an uniform distribution with a probability defined by $p_m = 0.5$. Crossover uses the scattered method, which creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. The fraction of the population that is created by crossover is defined by the parameter $p_c = 0.8$. For the elitism, 10 chromosomes are selected to survive for the next generation.

The fitness function has to be chosen carefully in order to achieve good results. For forward walking, a simple but effective fitness function to minimize can be the distance to the ball, assuming that the robot is placed in a fixed position away from the ball for each individual test. Additionally, the torso average oscillation is also used in order to obtain more stable gaits. The final version of the fitness function is stated as follows:

$$fitness = d_{Ball} + \overline{\theta} \tag{8}$$

where d_{Ball} is the distance to the ball (in meters) and $\overline{\theta}$ is the average oscillation of the torso (in radians per second). The torso average oscillation, $\overline{\theta}$ is a measure calculated with base on the values received from the gyroscope installed on the torso. It is calculated using the following equation [9]:

$$\overline{\theta} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2 + \sum_{i=1}^{N}(y_i - \overline{y})^2 + \sum_{i=1}^{N}(z_i - \overline{z})^2}{N}} \tag{9}$$

where N represents the number of simulation cycles, x_i, y_i and z_i are the values received from the gyroscope in the i^{th} cycle and $\overline{x}$, $\overline{y}$ and $\overline{z}$ are the arithmetic mean of gyroscope readings over the time. The velocity is implicitly considered in the evaluation since each test is also time-bounded with a fixed-length time.

5 Experimental Results

For the optimization process, GADS generates a file containing the parameters, and then the simulator tests those parameters. When the test finishes, the simulator generates a file with the resultant fitness, and GADS associates the fitness with the corresponding parameters. Then the process restarts until being stoped

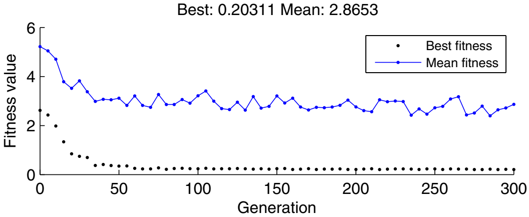


Fig. 3. Evolution of the fitness

Table 1. Best generated individual

A_1	57.1842	ϕ_1	2.9594	C_1	-88.4624
A_2	5.6445	ϕ_2	-2.2855	C_2	3.6390
A_3	57.1211	ϕ_3	0.0887	C_3	35.9536
A_4	39.6205	ϕ_4	-1.8292	C_4	-39.9481
A_5	46.6315	ϕ_5	1.7640	C_5	28.5095
A_6	3.7947	ϕ_6	-1.2067	C_6	-2.9360

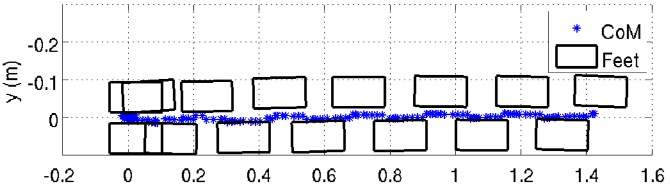
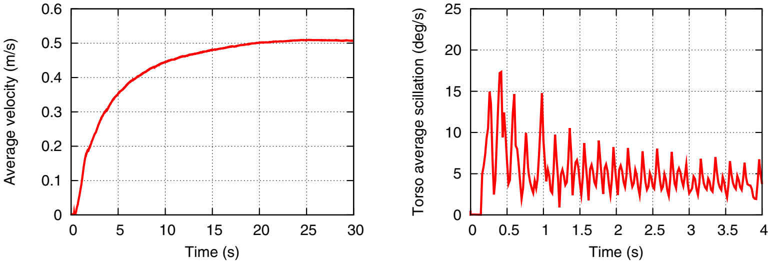


Fig. 4. The CoM and the feet in the XY plane



(a) (b)

Fig. 5. (a) Average velocity over time (b) Torso average oscillation over time

manually or using some criteria. The generation process took 5 entire days to complete using a Core 2 Duo 2.4Gz CPU with 1GB of physical memory. Figure 3 shows the evolution of the fitness. The minimum fitness is 0.20311, which is a very good result. Table 1 shows the values of the best individual for $A_{1..6}$, $\phi_{1..6}$ and $C_{1..6}$. For the period, T, the optimal generated value was 0.3711.

Figure 4 shows the evolution of the CoM and the placement of feet in the XY plane. It is possible to note that the robot tends to shift the CoM to the support foot while walking. Another characteristic shown by the same graphic is the large size of the steps.

The average velocity (Figure 5a) shows very good results. More than 50 centimeters per second were achieved. This is a good velocity taking into account the torso average oscillation, that is represented in the Figure 5b. The obtained results can be considered good results, comparing with the three best teams of the RoboCup 3D simulation league competition of 2008 (China, Suzhou). They

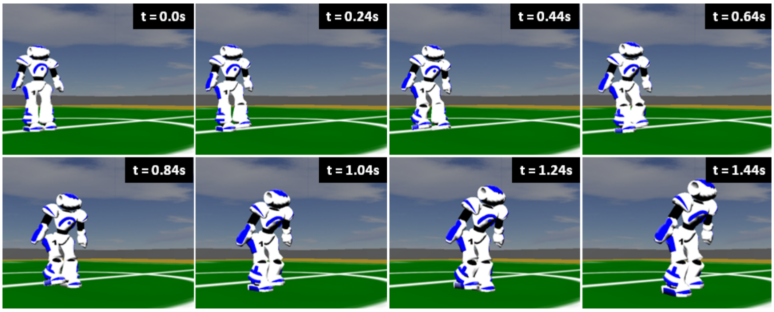


Fig. 6. Walking gait screenshots

were able to reach forward velocities of 1.20m/s (SEU-RedSun [10]), 0.67m/s (WrightEagle [11]) and 0.43m/s (LittleGreenBats [12])[1].

Figure 6 shows NAO walking forward using the proposed solution. At t=1.44s the biped already covered a great distance. It is also possible to note the large steps and the height of the steps.

6 Conclusion and Future Work

This paper proposed an efficient method for automatic generation of biped locomotion behaviors which consists of combining partial Fourier series and genetic algorithms (GA). GAs are based on the biological evolution of species by applying selection, mutation and crossover operators to a set of chromosomes. A forward walking gait was developed using 12 single-frequency oscillators whose parameters were defined using a GA for offline generation of parameters. GA proved to be very good on achieving the expected results. The generated walking gait is faster and it is also stable and has the great advantage of being less sensitive to the disturbances inherent to the simulation environment. The generated walk was mainly based in the CoM to monitor the quality of the gait. However, as future work, the calculation and monitoring of the ZMP trajectory is essential for achieving dynamic stability. Moreover, motion capturing, which consists of monitoring the human behaviors, provides a great way to define the humanoid behaviors, due to the anthropomorphic characteristics between both. As future work, it would be good to invest not only in genetic algorithms, but in machine learning methods such as reinforcement learning. These methods provide great advantages for the automatic generation of behaviors which reduce, and possibly eliminate, the human intervention during the optimization or learning process.

Acknowledgements

This research was partially supported by FCT-PTDC/EIA/70695/2006 Project – "ACORD - Adaptive Coordination of Robotic Teams".

---

[1] These results were retrieved from the logfiles of the RoboCup 2008 competition, which may be found in http://www.robocup-cn.org/.

References

1. Murphy, R.: Introduction to AI Robotics. MIT Press, Cambridge (2000)
2. Prado, M., Simón, A., Pérez, A., Ezquerro, F.: Effects of terrain irregularities on wheeled mobile robot. Robotica 21, 143–152 (2003)
3. Vukobratovic, M., Stepanenko, Y.: On the stability of anthropomorphic systems. Mathematical Biosciences 15 i1, 1–37 (1972)
4. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B.: The NAO humanoid: a combination of performance and affordability. CoRR (2008)
5. Obst, O., Rollmann, M.: Spark - a generic simulator for physical multi-agent simulations. In: Lindemann, G., Denzinger, J., Timm, I.J., Unland, R. (eds.) MATES 2004. LNCS, vol. 3187, pp. 243–257. Springer, Heidelberg (2004)
6. Holland, J.: Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control, and artificial intelligence. The University of Michigan Press (1975)
7. Darwin, C.: On the Origin of Species by Means of Natural Selection. Murray (1859)
8. The MathWorks. Genetic Algorithm and Direct Search Toolbox, User's Guide, Version 2. The MathWorks (2005)
9. Heinen, M., Osório, F.: Applying genetic algorithms to control gait of physically based simulated robots. In: Proceedings of 2006 IEEE Congress on Evolutionary Computation, pp. 500–505 (2006)
10. Yuan, X., Hui, S., Cheng, Q., Si, C., Yingzi, T.: SEU-RedSun 2008 soccer simulation team description. In: Proceedings CD of RoboCup 2008, China (2008)
11. Feng, X., Yunfang, T., Jiongkun, X., Weimin, Z., Dinghuang, J., Zhiqiang, X.C.Z.: Wright Eagle 2008 3D team description paper. In: Proceedings CD of RoboCup 2008, China (2008)
12. van Dijk, S., Klomp, M., Kloosterman, H., Neijt, B., Platje, M., van de Sanden, M., Scholtens, E.: Little Green Bats humanoid 3D simulation team description paper. In: Proceedings CD of RoboCup 2008, China (2008)

Motion Planning of a Non-holonomic Vehicle in a Real Environment by Reinforcement Learning[*]

M. Gómez[1], L. Gayarre[1], T. Martínez-Marín[2], S. Sánchez[1], and D. Meziat[1]

[1] Departamento de Automática, Escuela Politécnica Superior,
Universidad de Alcalá, Spain
[2] Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal,
Universidad de Alicante, Spain
`mgomez@aut.uah.es`

Abstract. In this work, we present a new algorithm that obtains a minimum-time solution in real-time to the optimal motion planning of a non-holonomic vehicle. The new algorithm is based on the combination of Cell-Mapping and reinforcement learning techniques. While the algorithm is performed on the vehicle, it learns its kinematics and dynamics from received experience with no need to have a mathematical model available. The algorithm uses a transformation of the cell-to-cell transitions in order to reduce the time spent in the knowledge of the vehicle's parameters. The presented results have been obtained executing the algorithm with the real vehicle and generating different trajectories to specific goals.

Keywords: Cell-Mapping, Dynamic Programming, Optimal Control, Reinforcement Learning, Q-Learning.

1 Introduction

Path planning and optimal control algorithms for solving specific practical problems, such as trajectory generation for Wheeled Mobile Vehicles (WMV) and, specifically, for Car-Like Vehicles (CLV) had been addressed in several research projects. The CLV are dynamic systems whose kinematics laws have been widely studied [1][2]. The motion planning and control are difficult tasks because they are non-holonomic systems. Typical wheeled mobile vehicles have fixed rear/front driving wheels and front steering wheels. In these kinds of vehicles, different optimal control algorithms are used to perform an intelligent trajectory generation [3]. Other approaches are based on algorithms that search the shortest path [2]. In real environments, where there are disturbances, closed-loop solutions using Dynamic Programming [1][4] provide good results but they require high computational effort and the dynamic model of the vehicle.

---

[*] This work has been supported in part by the Comisión Interministerial de Ciencia y Tecnología under the grant ESP2005-07290-C02-02, by the Generalidad Valenciana under Project GV07/214, and by the University of Alicante under Project GRE07/4P.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 813–819, 2009.
© Springer-Verlag Berlin Heidelberg 2009

There are other techniques that do not need a dynamic model for achieving a closed-loop solution. These techniques are based on reinforcement learning, where the optimal motion is estimated through an interaction between the vehicle and its environment. Due to the fact that optimal control problems are generally very difficult to solve, both analytically and numerically, the Cell-Mapping method [5][6][7][8] offers an effective numerical approach to obtain global and local solutions of the non-linear optimal control problem. However, it requires a mathematical model of the system and also a high computational effort because it needs to know the whole model of the system.

Taking into account the advantages associated to the Dynamic Programming, reinforcement learning techniques and Cell-Mapping techniques, a new algorithm is proposed in [8][9]. Usually, most of the systems in which we need to apply an optimal control are continuous. Thus, it is necessary to consider a discretization of the state space when finding optimal controllers [8][9][10].

2 Working Environment

The optimal control algorithm proposed in this work has been designed to be executed in a non-holonomic vehicle on a 1/8 scale, like the one shown in Figure 1. The control software developed has been designed in compliance with all constraints and memory limits of the hardware, which is based on the ARM7 LPC2138 processor. It is assumed that the concerned vehicle is moving on a horizontal plane. The two rear wheels are driven via a single motor through a differential device to get the same driving velocity. The two front steering wheels are controlled by a single axis to obtain the same steering. The body of the vehicle is perfectly rigid and the wheels are rigid too with the same spoke.

The platform shown in Figure 1 can move in a limited area shown in Figure 2, whose real dimensions are 1.80 × 2.60 m. This area is in the field of vision of a CMOS camera [11] in charge of obtaining the odometry information of the vehicle in real-time. The algorithm runs on a PC due to the memory constraints of the control board embedded in the vehicle, which is connected with the camera and vehicle by means of two different serial floating links. Also, there are two more floating cables to feed the camera and the vehicle.

Before executing the proposed algorithm in the platform depicted in Figure 1, it has been tested under a simulation environment using a mathematical model considering the kinematics and dynamics of the vehicle [8][9]. The state variables used in the algorithm are: linear velocity (v), Cartesian coordinates (X, Y) and orientation (θ).

The state of the vehicle is acquired in real time by a CMUcam3 camera mounted on the roof of the room, from which the embedded control software is able to acquire X and Y coordinates and orientation of the vehicle. Thanks to the camera capacity of tracking different colors, two different color cards have been located on the top of the vehicle in order to capture its position (X-Y) and from the center of each color card, the camera's embedded software is able to calculate the orientation (θ). Due to the resolution of the camera, the working area where the vehicle can move is quite reduced (1.90 x 2.60m). The origin of coordinates is in the centre of the area, and the orientation

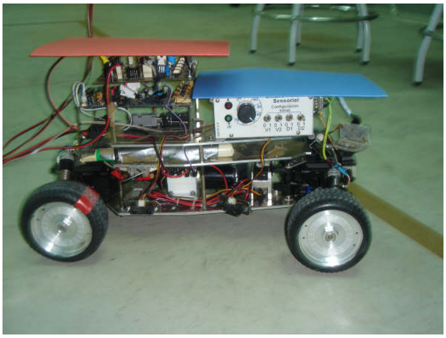


Fig. 1. Wheeled mobile vehicle

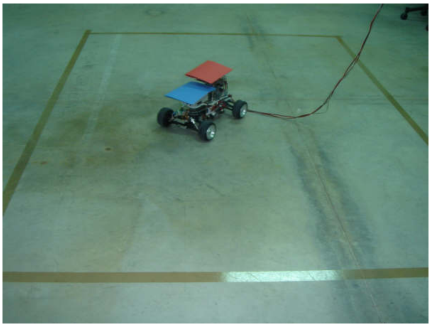


Fig. 2. Motion area of the vehicle with floating cables

is considered 0 degree when the vehicle is faced positive X axis. Finally, the velocity (v) is obtained from the position given by the encoder coupled to the traction motor of the vehicle.

3 Optimal Control Algorithm

The optimal control algorithm implemented in our experiments is based on the Cell-Mapping [5][6][7][8] and Reinforcement Learning [8][12][13][14] techniques. The algorithm proposed in [8] is modified applying a transformation of the *Cell-to-Cell Mapping*, in order to greatly reduce the time spent in the knowledge of the vehicle dynamic and the environment.

The Cell-Mapping technique is applied taking into account that a transition is valid when the distance between any state variable of the initial and final states is equals one. This low distance has two drawbacks: the learning time is longer and the period used for generation of transitions must be small enough for achieving the distance. The implemented algorithm creates a *Model-table* that is used for updating the Optimal Control Table (*Q-table*). The transitions from the origin of coordinates for each velocity are stored in the *Model-table* [8].

The error due to the positioning system may provoke a different action from the right one associated to the current state. This algorithm is impervious to this error, performing a motion planning that travels to the goal. Although this planning could be different depending on the error, the goal is always reached.

In this work, we introduce new transformation equations, from which the vehicle only needs to obtain the transitions from the origin for all actions (*model*). The rest of transitions among states are obtained by transformation. In this way, the learning time is noticeably reduced. It is important to highlight that the equations do not depend on the parameters of the vehicle. The transformation equations are specified next:

$$\begin{aligned} x_f &= x_0 + x_m \cos(\theta_0) - y_m \sin(\theta_0) \\ y_f &= y_0 + x_m \sin(\theta_0) + y_m \cos(\theta_0) \qquad \textit{Eqs. 2} \\ \theta_f &= \theta_0 + \theta_m \end{aligned}$$

where, x_f, y_f and θ_f are the state after the transition, x_0, y_0 and θ_0 are the state before the transition and x_m, y_m and θ_m are the state after the transition starting from the state (0, 0, 0). In Table 1, the new version of the algorithm proposed in [8][9] applying this transformation is specified. According to this algorithm, each EPISODE is equivalent to one training session. In each training session, the agent explores the environment, obtaining the accumulated time (reward) until it reaches the goal state or goes to the drain state. The purpose of the training is to enhance the "brain" of the controller that is represented by the *Q-table*. More training will give better *Q-table* that can be used by the controller for the motion planning of the vehicle. The first learning stage consists of acting on the system during a transition time that is controlled by the LPC2138 processor embedded in the platform (vehicle). When this time is over, an interruption is provoked to the processor in order to read the new state (v, X, Y and θ) and update the *Model-table* and *Q-table*. After this updating, a new action is applied in the vehicle until the transition distance equals 1 or it reaches a drain. The training session is performed during the necessary number of iterations to fill adequately the *Model-table*. If the vehicle goes to the drain, a reactive function is executed in order to put the vehicle into the state space again.

Table 1. *Pseudocode* of the new proposed algorithm

```
For each training EPISODE
  begin
    Read the current state;
    Select one action for the current state;
      Act on the system until distance is 1;
      Update Q-table;
      Update Model-table from received experience
         by the inverse transformation;
end.

Apply the direct transformation to all states;

Repeat N times
   Update Q-table using Model-table;
end.
```

Once the training session is finished, the direct transformation (Eqs. 2) is applied to all states using information stored in the *Model-table* created from received experience. After, the *Q-table* is updated N times (see Table 1). If we seize the main advantage of the transformation equations (reduction of the learning time), we could add more state variables to the algorithm in order to take into account other dynamic characteristics of the vehicle (e.g. steering velocity). The controller that we generate using the proposed algorithm in Table 1 is non-linear. We are able to perform an optimal control of a non-linear system (vehicle) without linearizing the model of the vehicle. One of the strong points of cell-mapping techniques is to be able to perform optimal control both in linear and non-linear systems using the same concepts and strategies.

4 Results

In this section, the results obtained after applying the proposed algorithm are presented. The results are based on the following scenario: the vehicle has learned how to move around the working area in the training session and it is able to reach the goal from any origin cell inside the state space. The parameters used by the algorithm are specified in Table 2. The grid dimension has been chosen taking into account the next criteria: an odd number of cells per variable, the same size of cell in X-Y and more accuracy in X, Y and θ.

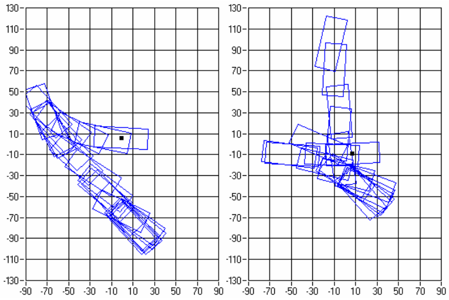
The transition time, also called “cell processing time” [6], is the average time that the robot needs to move to a different cell from the current one with a specific transition distance (in our case it is equals 1). The sample time that we used during the learning stage was 0.8 s. This time must be enough to receive the new state and act on the platform. The time needed to acquire the new state is restricted due to the tracking time of the camera (approximately 300 ms). But if we compare this time to the sample time, it is low enough. The error associated to each trajectory is not accumulative because we have the optimal control for reaching the goal in each cell.

The goal state that the vehicle has to reach from each initial state is (0, 0, 0, 0), and the initial states from where trajectories have been generated are: (0, 11, -74, -44.7) and (0, -13, 95, -100.3). Table 2 shows two trajectories from the previous initial positions. In both trajectories, the initial state of the vehicle is smoothly rotated due to the discretization error. The picture on the left of the Table 2 shows how the vehicle starts backwards from the origin state. When it reaches the coordinates (-70, 30), it goes forwards and reaches the goal state. In the picture on the right, the vehicle is orientated into the negative Y axis. It goes forwards turning to the left when it approximately reaches the point (0, 0). In the point (20, -40), it goes backwards reaching the final orientation (0º) in (-50, -10). Finally it goes forwards in order to reach the goal state. In both trajectories, we can see that the goal is reached.

One of the main characteristics of the proposed algorithm is that it is able to obtain the model of the system from received experience (saving the transitions), and applying the control law in order to reach a specific goal inside the state space from any origin in minimum time. The controller is generated by means of a direct method that determines the control rule without needing a mathematical model. In order to understand how the trajectories are related to the optimal strategy, it is necessary to take into account the bang-bang control concept and the called S-shaped switching curve. In optimal control problems, the control actions are usually restricted to be between a lower and an upper bound. In general, if the optimal control switches from one extreme to the other at certain times, then that control is referred to as a bang-bang solution. In our case, where we have a minimum time problem, it is a model of a bang-bang problem [15].

The S-shaped switching curve is typical of "bang-bang control". The robot starts moving as it turns until it reaches the switching curve, then it turns in the opposite direction in order to reach the goal (0, 0, 0, 0). The “S-shaped switching curve” separates the XY-plane into several subsets (red, yellow, blue and cyan), where the steering action assumes either a right turning or a left one. The Figure 3 shows two trajectories that the vehicle generates in order to reach the goal in minimum time. Independently of where the robot is located, the trajectories shall always reach the switching curve. Once it is reached, the robot will follow it. In practice, we can see that our solution approximates to the optimal controller.

Table 2. Parameters of the algorithm and trajectories without obstacles. Two trajectories are represented starting from different origin states. The goal state of each trajectory is marked with a black solid square

Sample period: 0.8 s. Grid dimension: 3x9x13x31 (cells) Traction motor PWM signal 0, 50, 100 $0 \le \tau_\delta \le 100$ (% high level) Steering servomotor PWM signal -23, 0, 23 $-23 \le \tau_\psi \le 23$ (degrees) Velocity range: $-0.33 \le v \le 0.33$ (m/s) X range: $-90 \le x \le 90$ (cm) Y range: $-130 \le y \le 130$ (cm) Orientation range: $-180 \le \theta \le 180$ (deg.)	

5 Conclusions

In this paper we have presented a new method for calculating optimal controllers applied to non-linear systems like non-holonomic autonomous vehicles. The combination of Cell-Mapping techniques along with reinforcement learning strategies allow us to reduce the time spent in the calculations of the controllers. Our proposal reduces the computation time because applying the transformation of the cell-to-cell mapping, the time needed to reach an optimal controller is about a few minutes. On the other hand, the off-line characteristic involved in the Cell-Mapping techniques is converted to an on-line method applying the *Q action-value function*. This is very important because we do not need the model of the system, it learns from received experience. In short, with this new algorithm, the vehicle can quickly learn the optimal actions involved in all states of the state space.

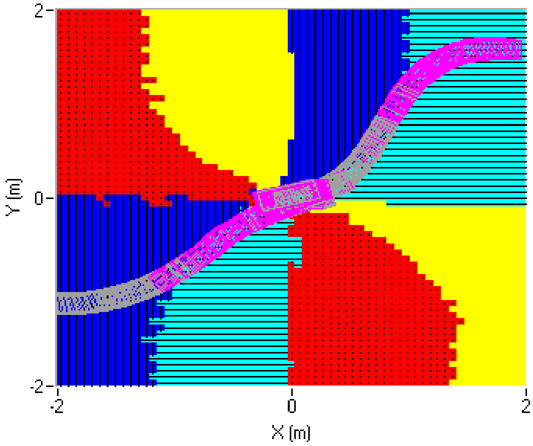


Fig. 3. S-shaped switching curve. The S-shaped switching curve has been generated with a constant velocity and orientation (0 degrees) in the XY-plane.

With the approach described, the optimal motion planning of a non-holonomic vehicle is robust in presence of external perturbations, due to the algorithm implements a closed-loop solution. Furthermore, the results of the proposed algorithm do not depend on possible changes in the physical or mechanical structure of the vehicle.

References

1. Gómez, M., Martínez, T., Sánchez, S.: Optimal trajectory generation using the Simple Cell-Mapping method for wheeled mobile vehicles. In: Proc. of SAAEI, pp. 1–5. Universidad de Alcalá, Spain (2002)
2. Reeds, J.A., Shepp, R.A.: Optimal paths for a Car that Goes both Forwards and Backwards. Pacific J. Math. 145(2), 367–393 (1990)
3. Qin, B., Soh, Y.C., Xie, M., Wang, D.: Optimal trajectory generation for wheeled mobile robot. In: 5th International Conference on Computer Integrated Manufacturing, Singapore, March 28-30, 2000, vol. 1, pp. 434–444 (2000)
4. Barraquand, J., Latombe, J.C.: On nonholonomic mobile robots and optimal maneuvering. Revue d'Inteligence Artificielle 3(2), 77–103 (1989)
5. Hsu, S.: A discrete method of optimal control based upon the cell state space concept. J. Optim. Th. Appl. 46(4) (1985)
6. Papa, M., Tai, H.M., Shenoi, S.: Cell Mapping for Controller Design and Evaluation. IEEE Control Systems Magazine, 52–65 (April 1997)
7. Zufiria, P.J., Martínez-Marín, T.: Improved Optimal Control Methods based upon the Adjoining Cell Mapping Technique. J. Optimization Theory and Applications 118(3), 657–680 (2003)
8. Gómez, M., Martínez, T., Sánchez, S., Meziat, D.: Optimal control applied to Wheeled Mobile Vehicles. In: Proceedings of 2007 IEEE International Symposium on Intelligent Signal Processing, October 2007, pp. 83–88. Alcalá de Henares, Madrid (2007)
9. Gómez, M., Martínez, T., Sánchez, S., Meziat, D.: Optimal control applied for Wheeled Mobile Vehicles based on Cell Mapping techniques. In: IEEE Intelligent Vehicles Symposium (IV 2008), June 4-6, 2008, Conference Proceedings Book (2008)
10. Munos, R., Moore, A.: Variable Resolution Discretization in Optimal Control. Machine Learning Journal 49, 291–323 (2002)
11. CMUcam3: Open Source Programmable Embedded Color Vision Platform, `http://www.cmucam.org`
12. Sutton, R.S., Barto, A.: Reinforcement Learning: An introduction. MIT Press, Cambridge (1998)
13. Watkins Christopher, J.C.H.: Learning from Delayed Rewards. PhD thesis. Cambridge University, Cambridge (1989)
14. Kaebling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Artificial Intelligence Research 4, 237–285 (1996)
15. Bryson, Ho: Applied optimal control: Optimization, estimation and control, book, p. 481. Hemisphere Pub. Corp. (1975) ISBN 0470114819

Applications in Bio-informatics and Biomedical Engineering

I. Rojas[1], H. Pomares[1], O. Valenzuela[2], and J.L. Bernier[1]

[1] Department of Computer Architecture and Computer Technology, CITIC-UGR
[2] Department of Applied Mathematics
University of Granada. Spain
`{irojas,hector,ovalenzuela,jbernier}@ugr.es`

Abstract. In this paper, an overview of the main topics presented in the special session of bioinformatics and biomedical engineering is presented. Bioinformatics consists of two subfields: the development of computational tools and databases, and the application of these tools and databases in generating biological knowledge to better understand living systems, being the main subject genomics and proteomics. Another knowledge scope close to the previous one are the problems related to medicine and biomedical engineering in which it is needed the participation of computer technologies and intelligent systems. The evolution of both disciplines, analyzing the number of publications presented in the bibliography during the last twenty years is presented.

Keywords: Bioinformatics, proteomics, genomics, biomedical engineering.

1 Introduction: Genesis of an Important Bioinformatics Project

On June 26th of the year 2000, the sciences of biology, medicine and bioinformatics were altered by an important event: The Prime Minister of the United Kingdom and President of the United States held a joint press conference, linked via satellite, to announce the completion of the draft of the Human Genome Project (HPG). The HPG was completed in 2003, and it was a 13-year project coordinated by the U.S. Department of Energy and the National Institutes of Health. During the early years of the HGP, the Wellcome Trust (U.K.) became a major partner; additional contributions came from Japan, France, Germany, China and others.

The sequence of three billion bases was the culmination of over a decade of work, during which the goal was always clearly in sight and the only questions were how fast the technology could progress and how generously the funding would flow (Table 1 shows the Landmarks in the Human Genome Project).

Bioinformatics, the intersection of molecular biology and computer science, is a fascinating and challenging area on which to work, hybridizing different paradigms and technologies [1]-[4]. Bioinformatics could be defined as the science and

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 820–828, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Table 1. Main landmarks in the Human Genome Project

1953	**Watson-Crick structure of DNA published.**
1975	F. Sanger, and independently A. Maxam and W. Gilbert, develop methods for sequencing DNA.
1977	Bacteriophage Q)X-174 sequenced: first 'complete genome'.
1980	US Supreme Court holds that genetically-modified bacteria are patentable. This decision was the original basis for patenting of genes.
1981	Human mitochondrial DNA sequenced: 16569 base pairs.
1984	Epstein-Barr virus genome sequenced: 172 281 base pairs.
1990	International Human Genome Project launched-target horizon 15 years.
1991	J. Craig Venter and colleagues identify active genes via Expressed Sequence Tags-sequences of initial portions of DNA complementary to messenger RNA.
1992	Complete low resolution linkage map of the human genome.
1992	Wellcome Trust and United Kingdom Medical Research Council establish The Sanger Centre for large-scale genomic sequencing, directed by J. Sulsron.
1992	J. Craig Venter forms The Institute for Genome Research (T1GR), associated with plans to exploit sequencing commercially through gene identification and drug discovery.
1995	First complete sequence of a bacterial genome, Haemophilus influenzae, by TIGR.
1996	High resolution map of human genome-markers spaced by near 600 000 base pairs,
1996	Completion of yeast genome, first eukaryotic genome sequence.
1998	Celera claims to be able to finish human genome by 2001. Wellcome responds by increasing funding to Sanger Centre
1998	Caenorhabditis elegans genome sequence published.
1999	Human Genome Project states goal: working draft of human genome by 2001 (90% of genes sequenced to >95% accuracy).
12/1999	Sequence of first complete human chromosome published.
06/2000	Joint announcement of complete draft sequence of human genome,
2003	Fiftieth anniversary of discovery of the structure of DNA. Completion of high-quality human genome sequence by public consortium

technique for organizing and analyzing biological data. The advent of various ‘omics’ tools motivated the rapid generation of enormous amounts of data, such as DNA and protein sequences, gene expression profiles and protein-protein interactions.

Related with this topic, biomedical engineering applications is also a very interested field, in which theoretical advances and applications of information systems, artificial intelligence, signal processing, electronics and other engineering tools in knowledge areas related to biology and medicine have a big impact in the multi-disciplinary research community. A large part of the information to support biological and biomedical research is available in an increasingly wide variety of rapidly growing decentralized databases, and the use of advanced computing plays an important role in this field. For example, the new so-called high throughput measurement and sequence techniques have made possible genome wide studies of gene function. The use of computer technology for storing DNA sequence information and constructing the correct DNA sequences from fragments identified by restriction enzymes was one of the first applications, arising from the Human Genome Project, where intelligent systems and computer applications have an important impact.

2 Application of Intelligent Systems in Bioinformatics and Biomedicine

Bioinformatics, understood as the development and use of computational techniques and tools for the analysis of biological information, is a discipline born to cover the needs of managing the enormous quantities of information coming from the projects destined to obtain the composition of life molecules (DNA and proteins). The development of Computer Science together with the establishment of Internet and molecule databases have made possible, in these last years, the study of genetic information by means of computer programs that accelerate the analysis of a high number of variables. In any case, the studies that Bioinformatics tries to carry out require the collaboration of a number of heterogeneous disciplines not so close to Biological problems. Computer science, as an indispensable complement, provides and develops systems, tools and algorithms for the study of the hereditary information. Statistics is essential for the management of large numbers and for the development of models with a high number of parameters. Physics and Chemistry are used in the development of models and in the analysis of molecular-level processes. Furthermore there are many disciplines that, in some way, tackle with the study of complex systems; and being biological systems the most complex in the known universe, there are many of those that can be involved in this new Science field, highlighting for example the, in principle seemingly remote, Information Theory.

The increasing computational demand is a real problem in the bioinformatics area, and is present at least in two fronts: the uncontrolled growth of the biological data that results in higher execution times (seldom in a linear relation) and new and more complex problems caused by the availability of these data. Sequential computing is not always capable enough to solve this demand. Parallel computing, the use of multiprocessor systems and the new approximations from Grid computing are coming out as the best options to successfully tackle with those problems.

Another knowledge scope close to the previous one are the problems related to medicine in which it is needed the participation of computer technologies. An area in which very successful results are being presented is the usage of intelligent systems formed using soft-computing paradigms (use of neural networks [7], fuzzy logic [8], support vector machines or evolutionary algorithms).

The research performed in the area of bio-medical applications using soft-computing techniques is an emerging field, in which every time there are more contributions and publications aimed to solve different problems related to medicine. The database MEDLINE (http://www.nlm.nih.gov/) is considered as one of the most important in the field, and the number of papers that make use of soft-computing tools has impressively grown during the last years. For example, during 1999-2003 more than two thousand and five hundred papers can be found related to fuzzy logic, neural networks and evolutionary algorithms, and during the last years the number of papers keeps on growing up. Although it is difficult to predict what will happen in the future, according to [5] and considering the classic evolution of a scientific discipline, the medical applications field tacked using intelligent systems capable of handling and facing complex problems with a high volume of data and high computational cost is under a phase of huge growth on an international level.

3 Main Topics of Bioinformatics and Biomedicine

Bioinformatics is an interdisciplinary research area at the interface between computer science and biological science. A variety of definitions exist in the literature and in the World Wide Web. Luscombe et al. define bioinformatics as a union of biology and informatics. Bioinformatics involves the technology that uses computers for storage, retrieval, manipulation and distribution of information related to biological macromolecules such as DNA, RNA, and proteins. Within bioinformatics two main topics are defined: genomics and proteomics.

- Genomics: The focus of genomics is the genome, being a genome all the DNA in an organism, including its genes. Genes carry information for making all the proteins required by all organisms. These proteins determine, among other things, how the organism looks, how well its body metabolizes food or fights infections, and sometimes even how it behaves. DNA is made up of four similar chemicals (called bases and abbreviated A, T, C, and G) that are repeated millions or billions of times throughout a genome. The human genome, for example, has 3 billion pairs of bases. The particular order of As, Ts, Cs, and Gs is extremely important. The order underlies all of life's diversity, even dictating whether an organism is human or another species such as yeast, rice, or fruit fly, all of which have their own genomes and are themselves the focus of genome projects. Because all organisms are related through similarities in DNA sequences, insights gained from non-human genomes often lead to new knowledge about human biology [3].

- Proteomics: Proteomics is focused on the identification of the tri-dimensional structure of the proteins, its interaction and its functionality. Papers about prediction of the structure and protein-protein interaction are presented in this session. The functional analyses include gene expression profiling, protein-protein interaction prediction, protein sub-cellular localization prediction, metabolic pathway reconstruction, and simulation [9].

The main aspects of bioinformatics analysis are not isolated but often interact in order to obtain a better understanding of the biological mechanism. For example, protein structure prediction depends on sequence alignment data; clustering of gene

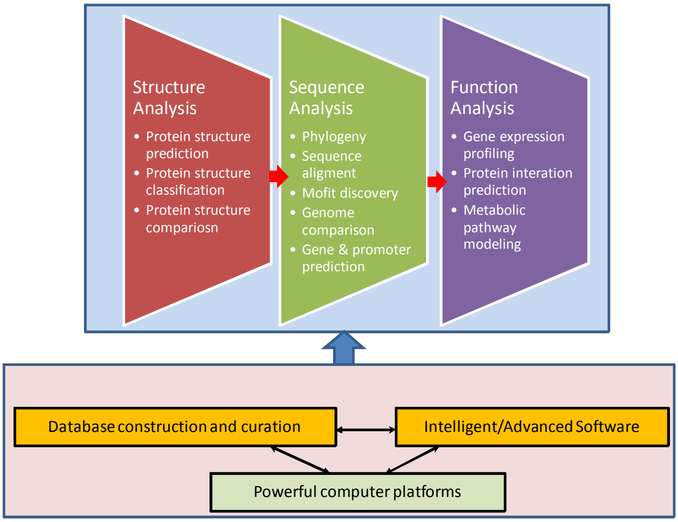
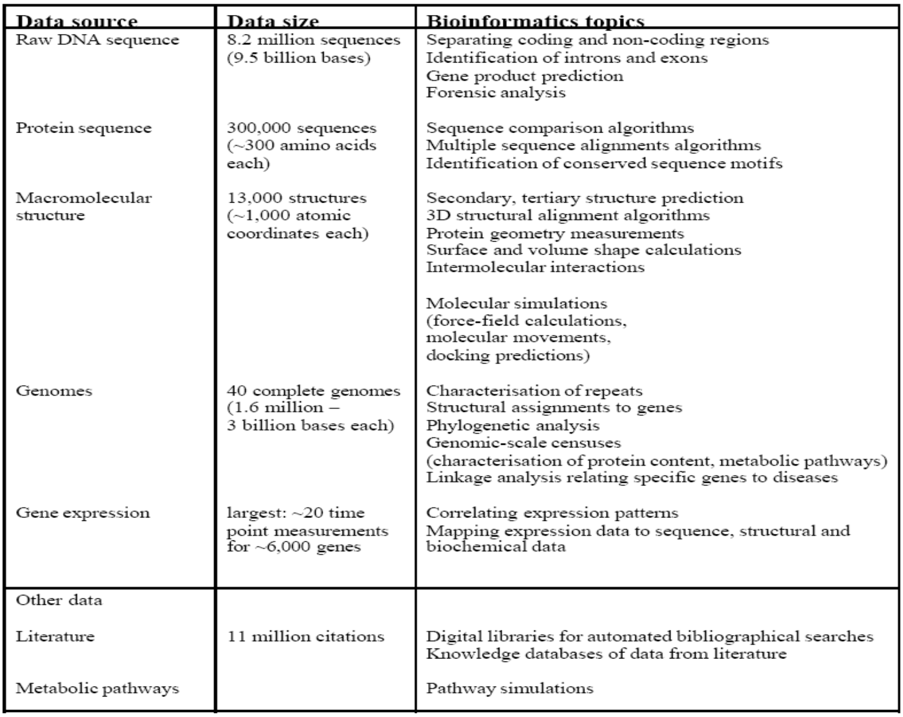

Structure Analysis
Protein structure prediction
Protein structure classification
Protein structure compariosn
Sequence Analysis
Phylogeny
Sequence aligment
Mofit discovery
Genome comparison
Gene & promoter prediction
Function Analysis
Gene expression profiling
Protein interation prediction
Metabolic pathway modeling
Database construction and curation
Intelligent/Advanced Software
Powerful computer platforms

Fig. 1. Main topic in bioinformatics and the intrinsic connections with software development

Table 2. Sources of data used in bioinformatic and subject areas that utilize this data when the draft of Human Genome was announced (August 2000, table from [6])



Data source	Data size	Bioinformatics topics
Raw DNA sequence	8.2 million sequences (9.5 billion bases)	Separating coding and non-coding regions Identification of introns and exons Gene product prediction Forensic analysis
Protein sequence	300,000 sequences (~300 amino acids each)	Sequence comparison algorithms Multiple sequence alignments algorithms Identification of conserved sequence motifs
Macromolecular structure	13,000 structures (~1,000 atomic coordinates each)	Secondary, tertiary structure prediction 3D structural alignment algorithms Protein geometry measurements Surface and volume shape calculations Intermolecular interactions Molecular simulations (force-field calculations, molecular movements, docking predictions)
Genomes	40 complete genomes (1.6 million – 3 billion bases each)	Characterisation of repeats Structural assignments to genes Phylogenetic analysis Genomic-scale censuses (characterisation of protein content, metabolic pathways) Linkage analysis relating specific genes to diseases
Gene expression	largest: ~20 time point measurements for ~6,000 genes	Correlating expression patterns Mapping expression data to sequence, structural and biochemical data
Other data		
Literature	11 million citations	Digital libraries for automated bibliographical searches Knowledge databases of data from literature
Metabolic pathways		Pathway simulations

expression profiles requires the use of phylogenetic tree construction methods derived in sequence analysis (Figure 1). The advent of various ‘omics tools allows the rapid generation of enormous amounts of data, such as DNA and protein sequences, gene expression profiles and protein-protein interactions [6].

Analyzing this new paradigm, some questions can be formulated. For example: How do we deal with these massive amounts of data and make sense out of them? How computer science and intelligent systems (such as neural networks) could deal

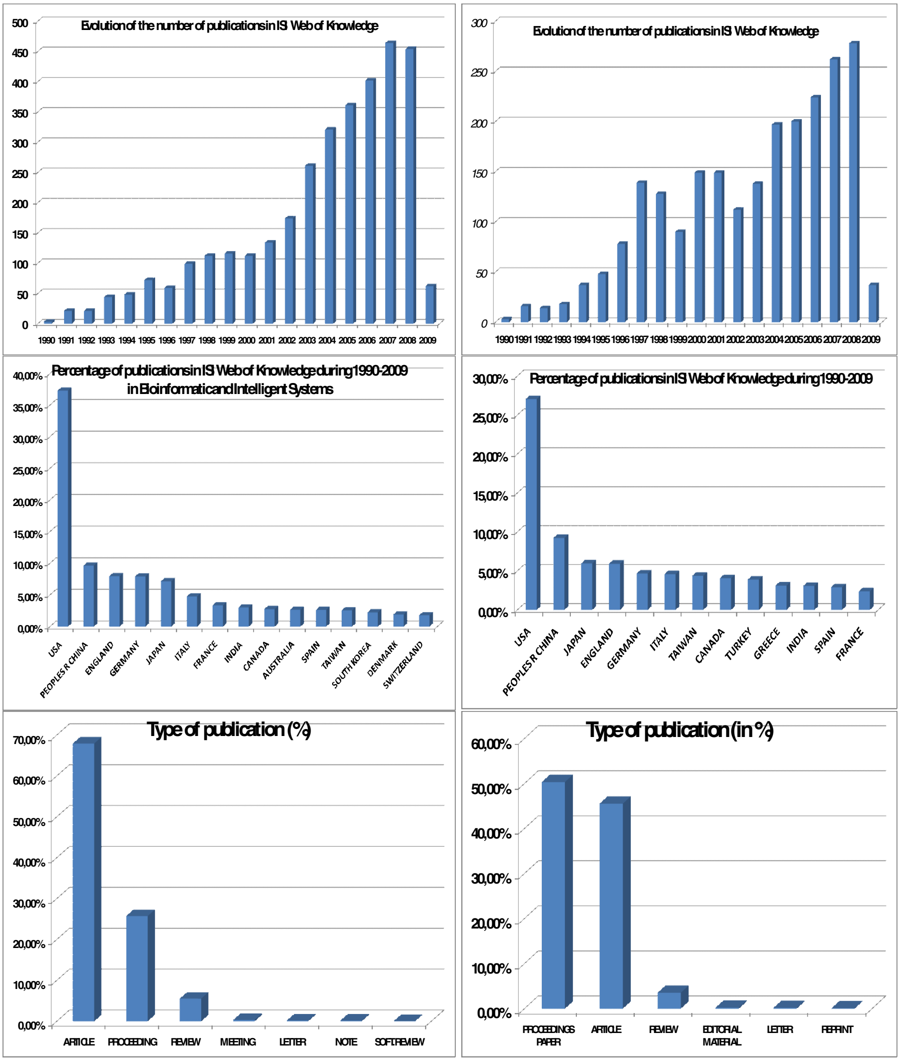


Fig. 2. A) Left: Analysis of the publication in bioinformatics and intelligent systems; B) Right: Analysis of the publication in biomedical engineering and intelligent systems

with and manage all the information contained in such large data base? In order to have an idea of the volume of data used in bioinformatics, Table 2 shows the order of magnitude used in genomics and proteomics.

The use of intelligent systems and powerful platforms (such as clusters of computers) is where bioinformatics becomes extremely useful and essential for the next generation biologists. The emphasis is on the use of computers because most of the tasks in genomics data analysis are highly repetitive or mathematically complex. The use of computers, and the use of intelligent/advanced systems, is absolutely indispensable in mining genomes for information gathering and knowledge building.

Another knowledge scope close to the previous one are the problems related to medical or biomedical applications using intelligent system [10]. This is a very active and energetic field, where multidisciplinary research teams work together in different disciplines, such as: Biomedical imaging, image processing & visualization, rehabilitation engineering and clinical engineering, health monitoring and wearable systems, bio-signal processing and analysis, biometrics and bio-measurement, medical robotics, micro nano biomedical devices & systems, neural and advanced systems engineering, telemedicine & healthcare; etc.

4 Future Trend and Conclusion

The evolution of the number of papers per year of publication, obtained by the ISI Web of Knowledge in Journal and Conference proceedings is shown in Figure 2.a for bioinformatics applications using intelligent systems (defined with keywords such as neural networks, fuzzy systems and genetic algorithms), and Figure 2.b shows biomedical applications using also intelligent systems (using medical or medicine or biomedicine and some of the intelligent systems previously defined). It is important to note, that, of course, this is just an example of the keywords used for the search. Nevertheless, we will try to explore the evolution of publications related with bioinformatics and biomedicine and soft computing techniques. It is difficult to predict the future, but considering the classic shape representing the life of the scientific disciplines (Figure 3), the field of bioinformatics and biomedicine applications using soft-computing techniques has left behind the initial period of slow and moderate growth and is now in the phase of an explosive/strong growth. One can expect that this phase will last until 2010, and then will reach the plateau for about 10-15 years (depending of the new discoveries).

The aim of the special session is to bring together researchers, professionals and industrial practitioners from all over the world for interaction and exchange of knowledge and ideas in all areas of bioinformatics, computational biology and biomedical engineering. Research, development or applications of advanced computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, represent, describe, store, analyze or visualize such data are welcome in this session (NIH).

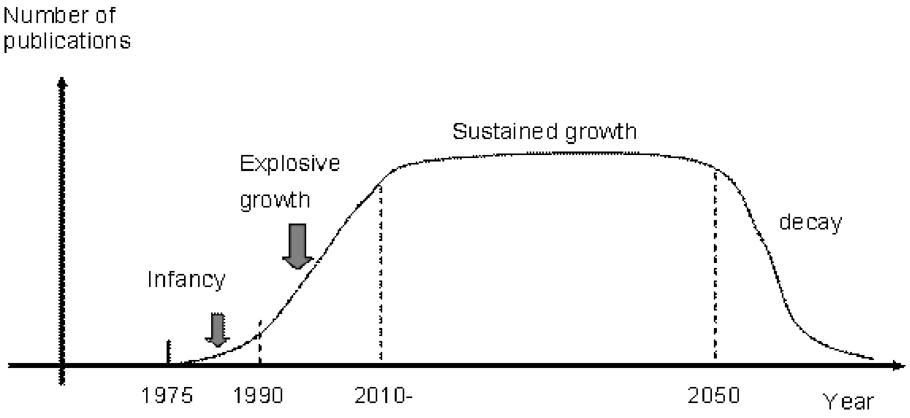


Fig. 3. Predictable development of the applications of intelligent systems in bioinformatics and biomedical engineering

Acknowledgement

This work has been partially supported by the CICYT Spanish Project TIN2007-60587.

References

1. Ramamurthi, K.S., Lecuyer, S., Stone, H.A., et al.: Geometric Cue for Protein Localization in a Bacterium. Science 323(5919), 1354–1357 (2009)
2. Venteicher, A.S., Abreu, E.B., Meng, Z.J., et al.: A Human Telomerase Holoenzyme Protein Required for Cajal Body Localization and Telomere Synthesis. Science 323(5914), 644–648 (2009)
3. Marass, F., Upton, C.: Sequence Searcher: A Java tool to perform regular expression and fuzzy searches of multiple DNA and protein sequences. BMC Res. Notes 2, 14 (2009)
4. Palidwor, G.A., Shcherbinin, S., Huska, M.R., et al.: Detection of alpha-rod protein repeats using a neural network and application to huntingtin. PLoS Comput. Biol. 5(3) (2009)
5. Teodorescu, H., Kandel, A., Jain, L.C.: Fuzzy and neuro-fuzzy systems in medicine. CRC Press, Boca Raton (1999)
6. Luscombe, N.M., Greenbaum, D., Gerstein, M.: What is bioinformatics? An introduction and overview. Yearbook of Medical Informatics 2001 (2001)
7. Rojas, I., Pomares, H., Gonzales, J., et al.: Analysis of the functional block involved in the design of radial basis function networks. Neural Processing Letters 12(1), 1–17 (2000)

8. Rojas, I., Ortega, J., Pelayo, F.J., et al.: Statistical analysis of the main parameters in the fuzzy inference process. Fuzzy Sets And Systems 102(2), 157–173 (1999)
9. Malekpour, S.A., Naghizaideh, S., Pezeshk, H., et al.: Protein secondary structure prediction using three neural networks and a segmental semi Markov model. Mathematical Biosciences 217(2), 145–150 (2009)
10. Ubeyli, E.D.: Decision support systems for time-varying biomedical signals: EEG signals classification. Expert Systems With Applications 36(2), 2275–2284 (2009)

A Large-Scale Genomics Studies Conducted with Batch-Learning SOM Utilizing High-Performance Supercomputers

Takashi Abe[1], Yuta Hamano[1,2], Shigehiko Kanaya[2], Kennosuke Wada[1],
and Toshimichi Ikemura[1]

[1] Nagahama Institute of Bio-Science and Technology, Tamura-cho 1266, Nagahama-shi,
Shiga-ken 526-0829, Japan
{takaabe,k_wada,t_ikemura}@nagahama-i-bio.ac.jp
[2] Nara Institute of Science and Technology, Ikoma, Japan
y-hamano@bs.naist.jp, skanaya@gtc.aist-nara.ac.jp

Abstract. Self-Organizing Map (SOM) developed by Kohonen's group is an effective tool for clustering and visualizing high-dimensional complex data on a two-dimensional map. We previously modified the conventional SOM to genome informatics, making the learning process and resulting map independent of the order of data input. This BLSOM developed on the basis of batch-learning SOM became suitable for actualizing high-performance parallel-computing using high-performance supercomputers. This BLSOM revealed phylotype-specific characteristics of oligonucleotide frequencies occurred in their genome sequences and thus permitted clustering (self-organization) of genome fragments (e.g., 10 kb) according to phylotype without phylogenetic information during the BLSOM learning. Using a high-performance supercomputer "the Earth Simulator", almost all prokaryotic, eukaryotic and viral sequences currently available could be classified according to phylotypes on a single map. Using this large-scale BLSOM, phylotypes of a large number of genomic fragments obtained by metagenome analyses of environmental samples could be predicted.

Keywords: batch learning SOM, oligonucleotide frequency, genome informatics, bioinformatics, high-performance supercomputer.

1 Introduction

Unculturable environmental microorganisms should contain a wide range of novel genes of scientific and industrial usefulness. Recently, sequencing analyses of mixed genome samples (metagenome analyses) of uncultured environmental micro- organisms have been established and vast amounts of short genomic fragments derived from species-unknown microorganisms have been sequenced. A large portion of the environmental sequences has been registered in the International DNA Sequence Databanks (DDBJ/EMBL/GenBank) with almost no phylogenetic and functional annotation, and therefore, in the least useful manner.

The homology search for nucleotide and amino-acid sequences such as BLAST has become widely accepted as a basic bioinformatics tool for both phylogenetic

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 829–836, 2009.
© Springer-Verlag Berlin Heidelberg 2009

characterization of gene/protein sequences and a prediction of their biological functions when genomes and genomic segments are decoded. Whereas usefulness of this sequence homology search is apparent, it is also clear that the sequence homology search can not properly predict phylogenetic origins of individual gene or genomic fragments when there does not exist an orthologous sequence set suitable for constructing a reliable phylogenetic tree. Therefore, phylogenetic origins of most of the genomic sequences obtained in metagenome analyses have not been predicted.

We previously modified the conventional SOM developed by Kohonen's group [1-3] to genome informatics on the basis of batch-learning SOM, making the learning process and resulting map independent of the order of data input [4-6]. The BLSOM thus developed could recognize phylotype-specific characteristics of oligonucleotide frequencies in their genomes and permitted clustering of genome fragments according to phylotypes with neither the orthologous sequence set nor the troublesome and mistakable processes of sequence alignment [6,7]. Furthermore, the BLSOM was suitable for actualizing high-performance parallel-computing with high-performance supercomputers such as "the Earth Simulator", and permitted clustering (self-organization) of almost all genomic sequences available in the International DNA Databanks on a single map [8-11]. In the present report, we describe the use of the BLSOM method for the phylogenetic classification of genomic fragments obtained by recent large-scale metagenome analyses.

2 Methods

We modified the conventional SOM developed by Kohonen and his colleagues [1-3] for genome informatics, based on batch-learning SOM, to make the learning process and resulting map independent of the order of data input [4-6]. The initial weight vectors were defined by PCA instead of random values on the basis of the findings that PCA can classify genomic sequences at least partly into phylogenetic groups. Oligonucleotide frequencies in vast amounts of genomic sequences were analyzed with BLSOM as described previously [5,6]. The BLSOM program suitable for PC cluster systems could be obtained from UNTROD, Inc. (k_wada@nagahama-i-bio.ac.jp).

Nucleotide sequences were obtained from http://www.ncbi.nlm.nih.gov/Genbank/. When the number of undetermined nucleotides (Ns) in a sequence exceeded 10% of the window size, the sequence was omitted from the analysis. When the number of Ns was less than 10%, the oligonucleotide frequencies were normalized to the length without Ns and included in the analysis.

In the International DNA Databanks, only one strand sequence of a pair of double strand sequences is registered, and the choice between the two complementary sequences of genomic fragments is primarily arbitrary in the databank registration. When global characteristics of oligonucleotide frequencies in the genome are considered, distinction of frequencies between the complementary oligonucleotides (e.g., AAAC versus GTTT) in the genome is not important in most cases. BLSOMs for phylogenetic classification, therefore, were previously constructed with frequencies for the degenerate sets of oligonucleotides, in which the frequencies of a pair of complimentary tetranucleotides (e.g. AAAC versus GTTT) were added. This roughly halved the computation time, but the accuracy level of the species-specific classification was almost equivalent [5-7]. On the basis of this previous observation, the present BLSOMs were constructed for the degenerate sets of oligonucleotides.

3 Results and Discussion

3.1 BLSOMs for 101 Eukaryotic Genomes

To investigate clustering power of BLSOM for sequences derived from a wide range of phylotypes, we initially analyzed tetra- and pentanucleotide frequencies in ca. 900,000 non-overlapping 10-kb sequences and the overlapping 100-kb sequences with a 10-kb sliding step from 101 eukaryotic genomes, most of which have been recently sequenced. In order to analyze sequences even derived from the lower eukaryotes with small genome sizes in accurate detail, excess representation of higher eukaryote's sequences derived from their large genomes had to be avoided. Therefore, in the cases of higher eukaryotes, 100-kb sequences were selected randomly from each large genome up to 200 Mb and used for the analyses. The BLSOM was constructed for tetra- and pentanucleotide frequencies in all 10- or 100-kb segments derived from the genome sequences thus chosen (Fig. 1). Here, the oligonucleotide frequencies were initially analyzed by PCA, and the resulting first and second principal components were used to set the initial weight vectors for the successive BLSOM [4-6]. After 145 cycles of the BLSOM learning, oligonucleotide frequencies in the fragment sequences were represented by the final weight vectors in the two-dimensional array, and the resulting BLSOM revealed clear separations (self-organization) according to phylotypes (Fig. 1). In this figure, a node that includes sequences from a single phylogenetic family is indicated in color, and one that includes sequences from more than one family is indicated in black. Sequences from

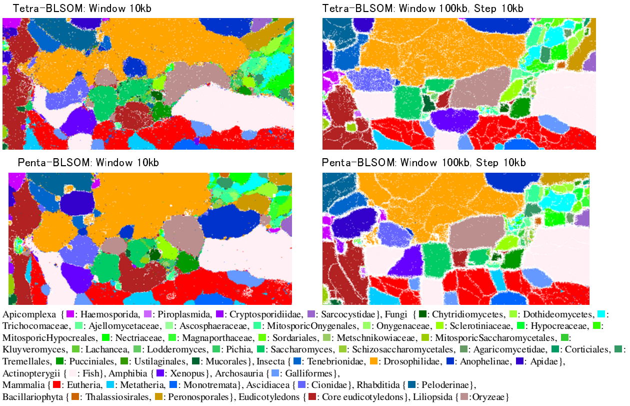


Fig. 1. BLSOMs for the non-overlapping 10-kb and the overlapping 100-kb sequences of 101 eukaryotic genomes. Nodes that include sequences from more than one family are indicated in black, those that contain no genomic sequences are indicated in white, and those containing sequences from a single family are indicated in colors shown under the figure.

each phylotype were clustered tightly on these tetra- and pentanucleotide BLSOMs (abbreviated as Tetra- and Penta-BLSOMs, respectively). Accuracy level of separation according to the families on the 10- or 100-kb BLSOM was approximately 90 or 99%, respectively.

In the 100-kb BLSOMs, the phylogenetic family territories were surrounded by contiguous white nodes, which contained no genomic sequences in the final map. In other words, family borders could be drawn automatically on the basis of the contiguous white nodes. This is because the representative vectors of the family-specific nodes were very distinctive between different families even near territory borders. Much narrower white borders were observed within a certain family territory, and these primarily represented genus/species separations.

3.2 Diagnostic Oligonucleotides for Phylotype Clustering

G+C% has long been used as a fundamental parameter for phylogenetic characterization of organisms and especially of microorganisms. However, G+C% is apparently too simple for a parameter to differentiate a wide variety of organisms. We previously found the G+C%, which was obtained from the weight vector of each node on BLSOMs for genomic sequence analyses, to be reflected in the horizontal axis [5]. Supporting this previous finding, G+C% increased from left to right on the Tetra- and Penta-BLSOMs (Fig. 1), and therefore, sequences with high G+C% were located on the right side. Importantly, sequences even with the same G+C% were clearly separated on BLSOMs by a complex combination of oligonucleotide frequencies, resulting in accurate phylotype separation; BLSOM recognized the phylotype-specific combination of oligonucleotide frequencies, which is the representative signature of each phylotype. This enabled us to identify the diagnostic oligonucleotides having the frequency pattern that is characteristic of each phylogenetic family (Fig. 2). In the analysis to identify the diagnostic

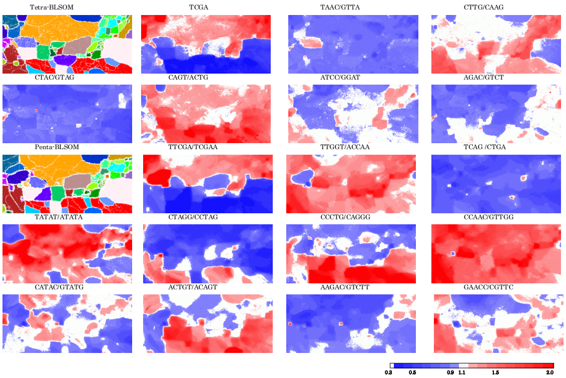


Fig. 2. Level of each tetra- or pentanucleotide in 100-kb BLSOM. Diagnostic examples of phylotype separations are presented. Level of each tetra- or pentanucleotide in each node in the 100-kb Tetra- and Penta-BLSOMs in Fig. 1 was calculated and normalized with the level expected from the mononucleotide composition of the node. The observed/expected ratio is indicated in colors shown at the bottom of the figure.

oligonucleotides, the frequency of each oligonucleotide in each weight vector in the 100-kb BLSOMs was calculated and normalized with the frequency level expected from the mononucleotide composition obtained from the weight vector at each node, and the observed/expected ratios are illustrated in red (overrepresented), blue (underrepresented) or white (moderately represented). This normalization allowed us to study oligonucleotide frequencies in each node independently of mononucleotide composition. Several examples of the diagnostic tetra- and pentanucleotides for the phylotype separations are presented in Fig. 2. Transitions between red (overrepresentation) and blue (underrepresentation) observed for these tetra- and pentanucleotide examples coincided exactly with family borders.

3.3 A Large-Scale BLSOM Constructed with Almost All Sequences Available from Species-Known Genomes

In Fig. 1, a good separation (self-organization) of eukaryotic sequences according to phylotypes was shown. We next examined phylogenetic separation of prokaryotic sequences, by applying the BLSOM to the phylogenetic prediction of sequences obtained from metagenomics studies. Large-scale metagenomics studies of uncultivable microorganisms in environmental and clinical samples have recently been conducted to survey genes useful in industrial and medical applications and to assist in developing accurate views of the ecology of uncultivable microorganisms in each environment. Traditional and conventional methods of phylogenetic classification of gene/genomic sequences obtained by metagenome analyses have been based solely on sequence homology searches and therefore inevitably focused on well-characterized gene sequences, for which orthologous sequences from a wide range of phylotypes are available for constructing a reliable phylogenetic tree. The well-characterized genes often are not industrially attractive. It would be best if microbial diversity and ecology could be assessed during the process of screening for novel genes with industrial and scientific significance. The present unsupervised and alignment-free clustering method, BLSOM, is thought to be the most suitable one for this purpose [10-15].

When we consider phylogenetic classification of genomic sequences derived from species-unknown environmental microorganisms obtained by metagenome analyses, it is necessary to construct BLSOM in advance with all sequences not only from species-known prokaryotes but also species-known eukaryotes, viruses and organelles available in the International DNA Databanks. This is because various eukaryotic and viral DNAs are inescapably present in environmental and clinical samples. On the basis of the findings in our previous exploratory studies of metagenome sequences [10-15], BLSOM was constructed with the tetranucleotide frequencies in 5-kb sequence fragments. Using a high-performance supercomputer “the Earth Simulator”, we could analyze almost all genomic sequences available from 2813 prokaryotes, 111 eukaryotes, 31486 viruses, 1728 mitochondria, and 110 chloroplasts. The 2813 prokaryotes were selected because at least 10-kb genomic sequences were registered in the International DNA Databanks.

The most important target of the phylogenetic classification of metagenome sequences is the sequences derived from species-unknown novel microorganisms. It is therefore necessary to keep good resolution for microorganism sequences on the BLSOM by avoiding excess representation of sequences derived from higher

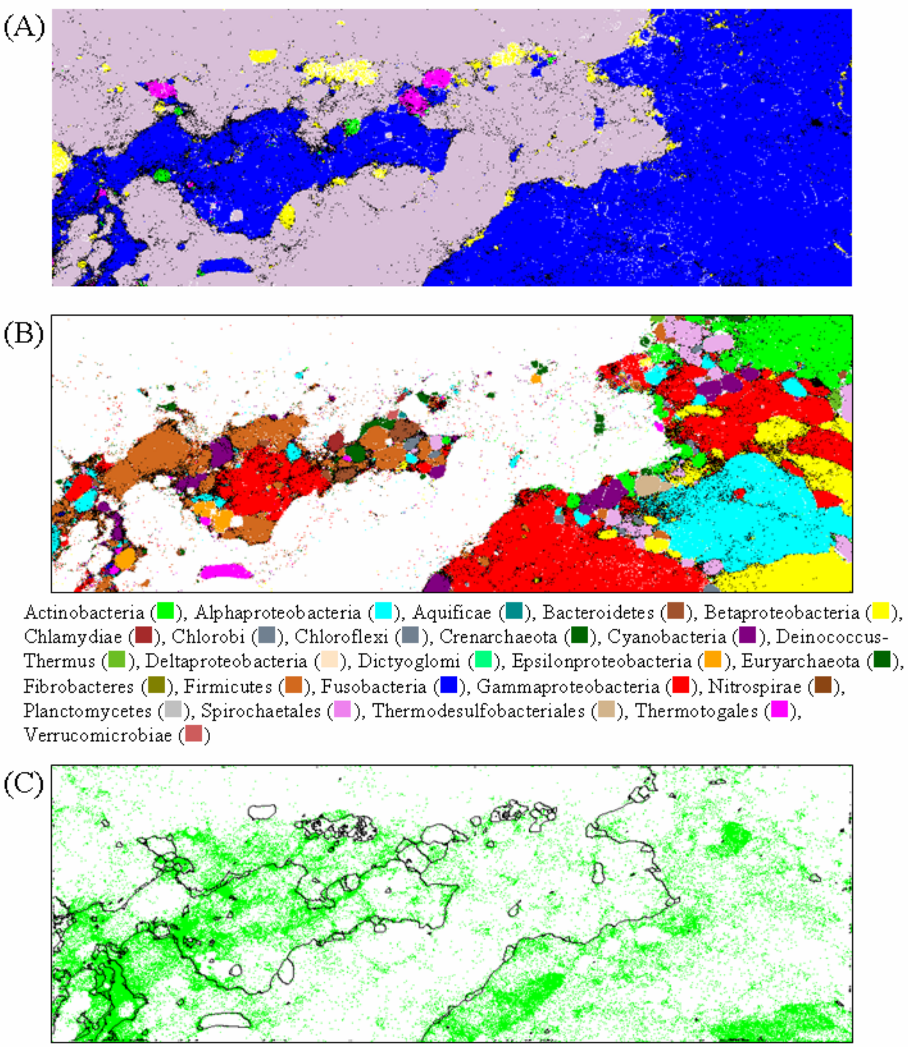


Fig. 3. 5-kb Tetra-BLSOM for almost all genomic sequences available in the International DNA Databanks. **(A)** Nodes that contain only the sequences from prokaryotes (🟦), eukaryotes (⬜), viruses (🟨), mitochondria (🟪), or chloroplasts (🟩) were separately colored, and those containing the sequences of more than one category are marked in black. **(B)** Nodes that contain only the sequences from one prokaryotic family were in colors shown at the bottom of the figure, and those containing the sequences of more than one family are marked in black. **(C)** Sargasso sequences longer than 1 kb were mapped onto above Tetra-BLSOM after normalization of the sequence length.

eukaryotes with large genomes. In the cases of higher eukaryotes, therefore, 5-kb sequences were selected randomly from each large genome up to 200 Mb. This allowed us to analyze an equivalent number of prokaryotic and eukaryotic 5-kb sequences and to provide good resolution for microbe sequences. On the Tetra-BLSOM constructed with these 5-kb sequences (Fig. 3A), the power of BLSOM to separate prokaryotic and eukaryotic sequences from each other was very high (ca. 97% accuracy). Clear separation of the species-known prokaryote sequences into 28 major families was also observed (Fig. 3B). As supporting the results in Fig. 1, the separation of eukaryotic sequences according to families was also observed on this 5-kb BLSOM (data not shown).

3.4 Phylogenetic Classification of Environmental Sequences

Venter et al. [16] applied the large-scale metagenome sequencing to mixed genomes collected from the Sargasso Sea near Bermuda and deposited vast amounts of fragmental sequences to the international DNA Databanks. Most of the Sargasso sequences have not been characterized phylogenetically in the Databanks because there are no orthologs sets for a large portion of these sequences, which are required to construct reliable phylogenetic trees necessary for proper prediction of their phylogenetic origins.

In Fig. 3C, the Sargasso sequences longer than 1 kb were mapped onto the Tetra-BLSOM presented in Fig. 3A; for each Sargasso sequence, the node with the shortest distance in the multidimensional vectorial space was assigned and indicated in green. Approximately 80% of the Sargasso sequences were classified into the prokaryotic territories and skewed distribution even in the prokaryotic territory was clear. This shows the ecological microbe populations representative in the Sargasso Sea. In the phylogenetic classification of genomic sequences derived from uncultivable and poorly characterized species, classification into phylotypes such as family and genus rather than into individual species is important. Using the BLSOM, we could assign 92 genera for the Sargasso sequences (http://lavender.genes.nig.ac.jp/takaabe/IWANN09/SPT1.htm).

In the case of environmental sequences from totally novel organisms, sequences even from related species may not be represented in the current DNA databanks and thus on the BLSOM in Fig. 3A, which was constructed with species-known sequences in the databanks. Such novel environmental sequences can be specified by calculating the distance between the vectorial data of the respective sequence and that of the sequence-mapped node in the multidimensional space. Another and a secure strategy for finding sequences derived from highly novel species is to construct BLSOM with all species-known sequences plus environmental sequences, and the result of such study was published previously [11].

Acknowledgements. This work was supported by Grant-in-Aid for Scientific Research (C) and for Young Scientists (B) from the Ministry of Education, Culture, Sports, Science and Technology of Japan. The present computation was done with the Earth Simulator of Japan Agency for Marine-Earth Science and Technology.

References

1. Kohonen, T.: Self-organized formation of topologically correct feature maps: Biol. Cybern. 43, 59–69 (1982)
2. Kohonen, T.: The self-organizing map. Proceedings of the IEEE 78, 1464–1480 (1990)
3. Kohonen, T., Oja, E., Simula, O., Visa, A., Kangas, J.: Engineering applications of the self-organizing map. Proc. IEEE 84, 1358–1384 (1996)
4. Kanaya, S., et al.: Gene classification by self-organization mapping of codon usage in bacteria with completely sequenced genome. Genome Inform 9, 369–371 (1998)
5. Abe, T., Kanaya, S., Kinouchi, M., Ichiba, Y., Kozuki, T., Ikemura, T.: Informatics for unveiling hidden genome signatures. Genome Res. 13, 693–702 (2003)
6. Kanaya, S., et al.: Analysis of codon usage diversity of bacterial genes with a self-organizing map (SOM) - characterization of horizontally transferred genes with emphasis on the E. coli O157 genome. Gene 276, 89–99 (2001)
7. Abe, T., et al.: Self-organizing map reveals sequence characteristics of 90 prokaryotic and eukaryotic genomes on a single map. In: WSOM 2003, pp. 95–100 (2003)
8. Abe, T., et al.: A large-scale Self-Organizing Map (SOM) constructed with the Earth Simulator unveils sequence characteristics of a wide range of eukaryotic genomes. In: WSOM 2005, pp. 187–194 (2005)
9. Abe, T., et al.: A large-scale Self-Organizing Map (SOM) unveils sequence characteristics of a wide range of eukaryote genomes. Gene 365, 27–34 (2006)
10. Abe, T., et al.: Sequences from almost all prokaryotic, eukaryotic, and viral genomes available could be classified according to genomes on a large-scale Self-Organizing Map constructed with the Earth Simulator. J. Earth Simulator 6, 17–23 (2006)
11. Abe, T., Sugawara, H., Kinouchi, M., Kanaya, S., Ikemura, T.: Novel phylogenetic studies of genomic sequence fragments derived from uncultured microbe mixtures in environmental and clinical samples. DNA Res. 12, 281–290 (2005)
12. Hayashi, H., et al.: Direct cloning of genes encoding novel xylanases from human gut. Can. J. Microbiol. 51, 251–259 (2005)
13. Uchiyama, T., Abe, T., Ikemura, T., Watanabe, K.: Substrate-induced gene-expression screening of environmental metagenome libraries for isolation of catabolic genes. Nature Biotech. 23, 88–93 (2005)
14. Abe, T., Sugawara, H., Kanaya, S., Ikemura, T.: A novel bioinformatics tool for phylogenetic classification of genomic sequence fragments derived from mixed genomes of environmental uncultured microbes. Polar Bioscience 20, 103–112 (2006)
15. Kosaka, T., et al.: The genome of Pelotomaculum thermopropionicum reveals niche-associated evolution in anaerobic microbiota. Genome Res. 18, 442–448 (2008)
16. Venter, J.C., et al.: Environmental genome shotgun sequencing of the Sargasso Sea. Science 304, 66–74 (2004)

Clustering Method to Identify Gene Sets with Similar Expression Profiles in Adjacent Chromosomal Regions

Min A. Jhun[1] and Taesung Park[1,2,*]

[1] Bioinformatics Program, Seoul National University, Seoul, 151-742, South Korea
[2] Department of Statistics, Seoul National University, Seoul, 151-742, South Korea
mhj617@snu.ac.kr,
tspark@snu.ac.kr

Abstract. The analysis of transcriptional data accounting for the chromosomal locations of genes can be applied to detecting gene sets sharing similar expression profiles in an adjacent chromosomal region. In this paper, we propose a new distance measure to integrate expression profiles with chromosomal locations. The performance of the proposed distance measure is evaluated via the bootstrap resampling procedure. We applied the proposed method to the microarray data in *Drosophila* genome and identified the set of genes of Toll and Imd pathway in adjacent chromosomal regions. Not only the proposed method gives stronger biological meaning to the clustering result, but also it provides biologically meaningful gene sets.

Keywords: clustering, chromosomal location, pathway, microarray.

1 Introduction

The microarray is a high throughput technique to investigate expression of thousands of genes, simultaneously [1]. The clustering techniques have been developed and applied to identify groups of genes with similar expression patterns [2]. The accumulation of vast amounts of genetic information makes it possible to integrate expression data into other types of genetic information, such as chromosomal location [3] and pathway information [4]. In particular, the analysis of transcriptional data with the chromosomal location of genes can be applied to detect chromosomal regions of genes sharing similar expression profiles. In our research, we assume that there exist some correlations between the expression profile and chromosomal location. Several evidences have been reported to support our assumption about a possible correlation between expression profile and chromosomal location [7-11].

In prokaryotic genome, genes are not randomly distributed and adjacent genes are expressed in groups, known as operons [5]. The operon structure contains one or more adjacent genes which are transcribed into one polycistronic mRNA and has function as a unit by sharing the same operators and regulators [6]. In *Saccharomyces cerevisiae*, the expression patterns from various conditions along the linear gene order

---

[*] Corresponding author.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 837–844, 2009.
© Springer-Verlag Berlin Heidelberg 2009

of the chromosomes reveal that the genes from transcriptomes dedicated to cell cycle phases, sporulation, and the pheromone responses are found in pairs throughout the genome [7]. Similarly, in *Caenorhabditis elegans*, mRNA-tagging and microarray approaches have discovered clusters of two to five genes that are co-regulated in specific cell lineages, including muscle, sperm, oocytes, and germ line [8]. An analysis of gene expression in *Drosophila*, using microarray data obtained under 80 different experimental conditions, has revealed an organization of co-expressed genes in groups of 10 to 30, covering 20 to 200 kb [9]. The clustering of genes within the *Drosophila* genome has also been demonstrated by expressed sequence tag (EST) database analyses of tissue-specific expression from the testis, head region, and embryo. In each cell type, the co-regulated genes are organized into clusters of three or more genes, with a trend toward grouping with a large number of genes [10]. In humans, several metabolic pathways are also shown to have their protein components encoded in genes that are proximal in the genome [11].

In this research, we propose new distance measures for the clustering analysis to integrate expression profiles with chromosomal location information of genes. New distance is defined as a weighted sum of two distances: one is based on the expression data and the other is on the chromosomal location. The contribution of each distance is controlled by a weight parameter ω. New distance measure can be applied to any distance-based clustering methods such as hierarchical clustering, K-means clustering and Self-Organizing Maps [12, 13]. In order to evaluate the proposed distances we apply bootstrapping method to the hierarchical clustering method using our distance measures. The original bootstrapping method and multi-scale bootstrapping method are applied to find significant clusters. We applied the hierarchical clustering method using our distances to the analysis of *Drosophila* genome and identified the gene set of Toll and Imd pathway in adjacent chromosomal regions.

2 Methods

2.1 Datasets

We applied our method to Toll and Imd pathway data in *Drosophila* [14]. Microrray analysis was performed with Affymetrix *Drosophila* GeneChips as described in Gregorio *et al.*[15] They observed that some of the genes responding to microbial infection are located in the same cytological region or are associated in transcriptionally co-regulated genomic clusters. Fig.1 is a modified version of Figure 4 in De Gregorio *et al.*'s paper.

The chromosomal location of each gene was extracted from the annotation file which is provided by Affymetrix (http://www.affymetrix.com/). The start location of gene in chromosome is used for representing the chromosomal location. The unit is a bp (base pair).

The expression values are normalized via the RMA normalization method [16]. As the expression data and chromosomal data have different scales and distributions, the expression values of all genes of each sample and the chromosomal location values are standardized to mean 0 and variance 1.

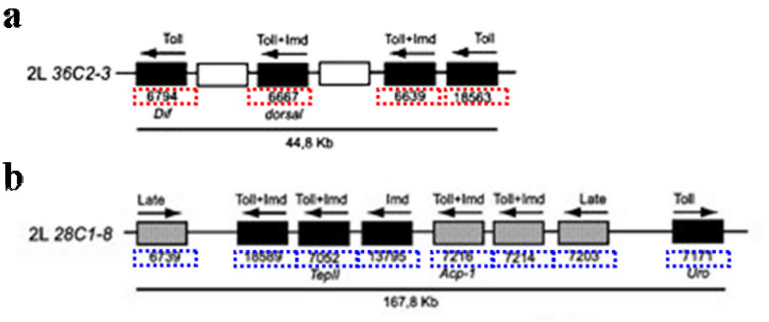


Fig. 1. Examples of local genomic clusters of genes of Toll and Imd pathway in *Drosophila* chromosome 2L (a) is the cluster of four genes in chromosome 2L from 36C2 to 36C3 and (b) is the clustering of eight genes in chromosome 2L from 28C1 to8. Each black rectangle corresponds to one infection-induced gene, grey rectangles to infection-repressed genes and white rectangles represent genes not regulated by infection. The CG number and the name of the gene (when known) are given below. The arrow on the top of each rectangle indicates the direction of transcription. The sizes of local gene clusters are given in kb.

2.2 Weighted Distance Measure

In the gene expression data, suppose there are p genes and n samples. We denote x_{ik} as an expression value of the i th gene from the k th sample. Then $gene_i = [x_{i1}, \cdots, x_{in}]$ denotes an expression vector of the i th gene. Let $x_{i,chr}$ denote the chromosomal location of the i th gene.

We propose a new weighted distance measure to integrate gene expression data and chromosomal location. Using the Euclidean distance, the expression distance between $gene_i$ and $gene_j$ is computed as follows:

$$d_{\exp}(gene_i, gene_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \ . \tag{1}$$

The chromosomal distance is defined as the absolute value of difference between chromosomal locations of two genes.

$$d_{chr}(gene_i, gene_j) = \left| x_{i,chr} - x_{j,chr} \right| \tag{2}$$

To scale the distance measure to range between 0 and 1, we find the maximum values of the expression distances and the location distances.

$$d_{\max,\exp} = \max_{1 \le i, j \le n} \left[d_{\exp}(gene_i, gene_j) \right] \tag{3}$$

$$d_{\max,chr} = \max_{1 \le i, j \le n} \left\lfloor d_{chr}(gene_i, gene_j) \right\rfloor \tag{4}$$

The expression distances and the location distances are then divided by these maximum values, respectively. Integration of the expression distance with the chromosomal distance is made by using a weight parameter ω . The integrated distance between the i th gene and the j th gene, D_{ij} , is defined as follows:

$$D_{ij} = \omega \frac{d_{\exp}(gene_i, gene_j)}{d_{\max,\exp}} + (1-\omega) \frac{d_{chr}(gene_i, gene_j)}{d_{\max,chr}} . \tag{5}$$

2.3 Clustering Analysis and Calculation of Bootstrap Probability

We apply hierarchical clustering method with the average linkage. Note that our distances can be applied to any kinds of distance-based clustering methods.

To evaluate uncertainty of the hierarchical clustering result in terms of the cluster sizes and reliability of each individual cluster, we calculate the probability values for each cluster using bootstrap resampling technique. The bootstrap probability (BP) value is calculated by the ordinary bootstrap resampling with replacement [17]. Additionally, the multiscale bootstrap resampling is used for calculating approximately unbiased (AU) probability values [18-19].

3 Results

3.1 An Application Result in *Drosophila* Chromosome 2L

The results of hierarchical clustering in *Drosophila* chromosome 2L are shown in Fig. 2. Fig. 2(a) is the result of the ordinary distance using the expression values, and Fig. 2(b) is the result of the weighted distance using both the expression values and the chromosomal locations.

As the ordinary distance is used, the genes belong to the two clusters are not closely clustered except CG7203, CG7214 and CG7216 (Fig. 2(a)). Although CG6667, CG6794, CG7052 and CG13795 were included in the red box with significantly high AU values, but two of them belonged to 2L 36C2-3 (RED) and others belonged to 2L 28C1-8 (BLUE). Most genes belonging to the clusters did not show any consistent pattern.

As the weighted distance is used the genes belong to the two clusters are well separated to two different branches (Fig. 2(b)). In the left branch, CG6667, CG6794 and CG18563 were clustered closely with high AU values. On the other hand, CG6639 was isolated as shown in the original hierarchical clustering. In the right branch, seven genes of 2L 28C1-8 (BLUE) were clustered together but they were separated into two groups: one for the induced genes and the other for the repressed genes.

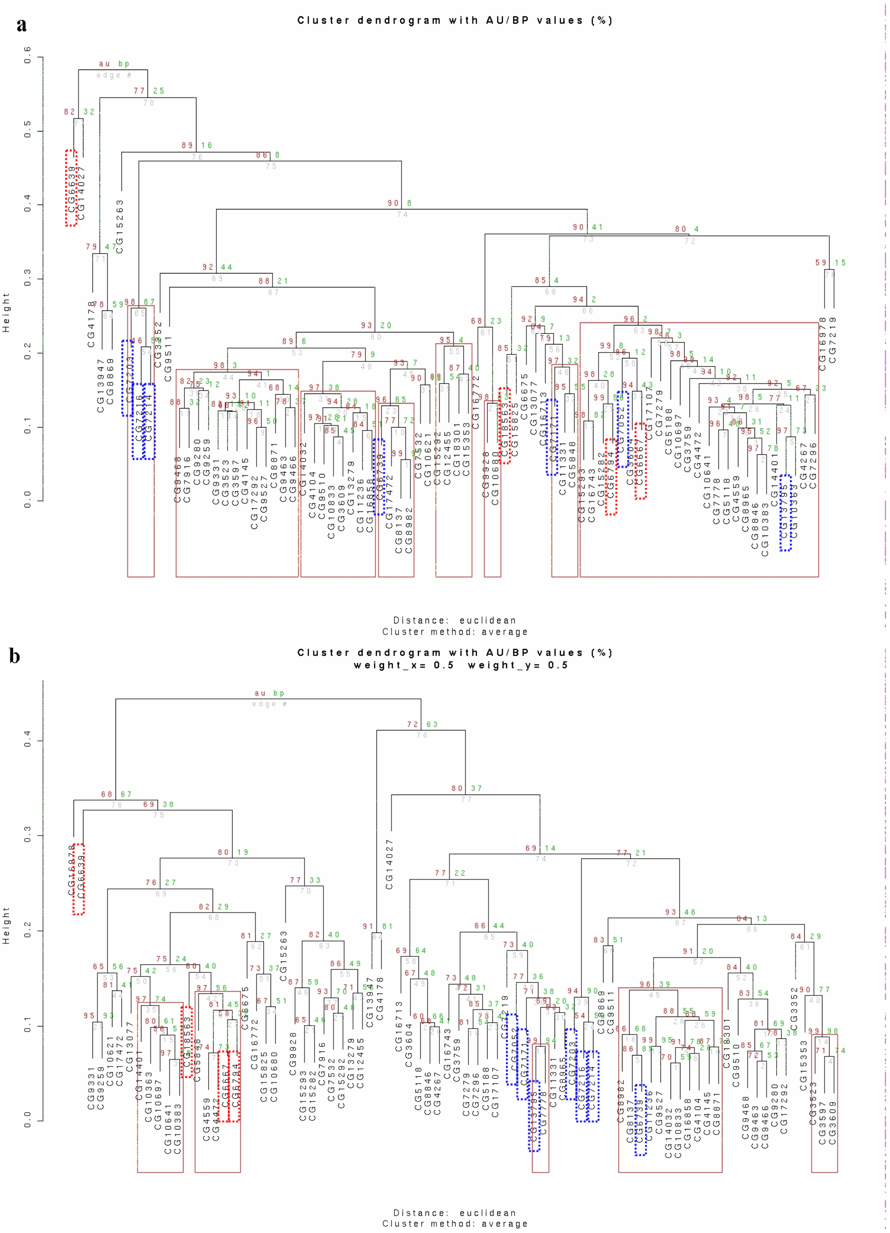


Fig. 2. Dendrogram of genes in Drosophila chromosome 2L (a) is the result of the hierarchical clustering using the expression data and (b) is the result of the weighted distance. The genes in the dashed red rectangle are members of 2L 36C2-3 cluster (Fig. 1(a)) and the genes in the dashed blue rectangle are members of 2L 28C1-8 cluster (Fig. 1(b)). The values at branches are approximately unbiased (AU) probability (left, red) and bootstrap probability (BP) (right, blue). The straight lined red box represents the significant cluster with AU $\geq$ 95.

Although we identified several groups of significantly clustered genes (red boxes), only three genes of 2L 36C2-3 (RED) and 2L 28C1-8 (BLUE) were located in the red boxes. In the second red box (ordering direction is left to right), among five genes only two genes are the member of 2L 36C2-3 (RED). In the third and fourth red boxes, only one gene of 2L 28C1-8 (BLUE) is included for each box. In this case, other genes included in red boxes can be unknown genes that have similar expression profiles with the known genes (CG13795 in third box and CG6739 in fourth box) in adjacent chromosomal region. The first and last red boxes do not include any member of 2L 36C2-3(RED) or 2L 28C1-8 (BLUE). The genes of these boxes can be candidate for another cluster of genes that have correlated expressions in the adjacent chromosomal regions.

3.2 The Biological Validation of the Results

The red boxes of genes are statistically significant clusters integrating expression data with chromosomal location. To examine the biological significance of these clusters, we examined the characteristics of genes such as gene names and Gene Ontology terms using FlyBase database (http://flybase.bio.indiana.edu/).

In the first cluster, there are five genes in the range of 2.2Mbp in 2L chromosome. The function of two genes in this cluster is unknown and the other three genes are shown to be related to enzyme activities, immune and development. In the second cluster, there are also five genes in the range of 1.1Mbp. Three of them are shown to be related to transcription activity, immune response, organ development, regulation of metabolism and cell communication. Other two genes are imaginal disc growth factors 1 and 3 that are related to cell-cell signalling and carbohydrate metabolism. The third cluster includes two genes but the function of one gene is unknown and other gene has ion and amino acid transporter activity. In the fourth cluster, the biggest cluster, there are 11 genes in the range of 3.8Mbp. One is unknown and others are related to hormone or enzyme activity in molecular function. In biological process, six genes are related to electron or phosphate transport, two genes are related to proteolysis and one is related to ovum, sperm and mating. In the last cluster, there are three genes in the range of 0.7Mbp with enzyme fatty-acid synthase and oxidoreductase activities.

Most genes in the same cluster shared the similar functions. The second cluster was characterized by transcription activity, immune response, organ development, regulation of metabolism and cell communication. Most genes from the fourth cluster shared the function of hormone, enzyme activity, electron or phosphate transport. In the last cluster, oxidoreductase activity and electron transport are shared by most genes.

4 Discussion

Our research was motivated by the question about the relationship between gene expression profile similarity and chromosomal adjacency. There is no clear explanation about the genomic structure inducing the clustering of co-expressed genes. There have been many researches demonstrating that the genes having similar expression tend to locate in adjacent chromosomal region.

To take into consideration the chromosomal location in clustering analysis, we proposed a new distance measure which is defined as the weighted sum of the expression distance and the chromosomal distance. Since the contribution of each distance is determined by a weight parameter ω, it is quite flexible to control the contribution of each distance.

Our clustering method has several advantages over the original hierarchical clustering. First, our method can cluster the ambiguous genes in aspect of expression profiles. The original hierarchical clustering is a one-dimensional clustering method using only the expression information, while our method is a two-dimensional clustering method using both expression information and chromosomal information. The additional chromosomal information reduces the ambiguity of clustering result and improves the performance of clustering. Secondly, our method helps to reduce the size of chromosomal region of genes sharing the same profiles and clusters them into the more compact regions. Since the biological experiment for validation requires considerable cost and time, reduction of candidate gene set and candidate region in genome is quite useful. In many cases, clustering result is exhibited in two groups that consist of up-regulated genes and down-regulated genes, which does not provide much information to biologists. On the contrary, our methods separate the large group into the several small groups based on chromosomal information. The reduced gene set and compact chromosomal region would provide more useful information.

Further, the weighted distance can be applied to other kind of clustering methods such as Self-Organizing Maps (SOM). We remain the application of our idea to other clustering methods as a future work [13].

Acknowledgments. This work was supported by the National Research Laboratory Program of Korea Science and Engineering Foundation (M10500000126).

References

1. Chee, M., et al.: Accessing genetic information with high-density DNA arrays. Science 270, 467–470 (1996)
2. Michael, B.E., et al.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences of the United States of America 95(25), 14863–14868 (1998)
3. Caron, H., et al.: The human transcriptome map: Clustering of highly expressed genes in chromosomal domains. Science 291(5507), 1289 (2001)
4. Lee, J.M., Sonnhammer, E.L.L.: Genomic Gene Clustering Analysis of Pathways in Eukaryotes. Genome Res. 13(5), 875–882 (2003)
5. Miller, J.H., Reznikoff, W.S.: The operon. Cold Spring Harbor Laboratory Press, Cold Spring Harbor (1978)
6. Jeong, K.S., Ahn, J., Khodursky, A.B.: Spatial patterns of transcriptional activity in the chromosome of Escherichia coli. Genome Biology 5, R 86(2004)
7. Cohen, B.A., et al.: A computational analysis of whole-genome expression data reveals chromosomal domains of gene expression. Nat. Genet. 26(2), 183–186 (2000)
8. Roy, P.J., et al.: Chromosomal clustering of muscle-expressed genes in Caenorhabditis elegans. Nature 418(6901), 975–979 (2002)
9. Spellman, P., Rubin, G.: Evidence for large domains of similarly expressed genes in the Drosophila genome. Journal of Biology 1(1), 5 (2002)

10. Boutanaev, A.M., et al.: Large clusters of co-expressed genes in the Drosophila genome. Nature 420(6916), 666–669 (2002)
11. Hurst, L.D., Pal, C., Lercher, M.J.: The evolutionary dynamics of eukaryotic gene order. Nat. Rev. Genet. 5(4), 299–310 (2004)
12. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
13. Kohonen, T.: The self-organizing map. Neurocomputing 21(1-3), 1–6 (1998)
14. De Gregorio, E., et al.: The Toll and Imd pathways are the major regulators of the immune response in Drosophila. Embo. J. 21(11), 2568–2579 (2002)
15. De Gregorio, E., et al.: Genome-wide analysis of the Drosophila immune response by using oligonucleotide microarrays. Proc. Natl. Acad. Sci. U S A 98(22), 12590–12595 (2001)
16. Irizarry, R.A., et al.: Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics 4(2), 249–264 (2003)
17. Suzuki, R., Shimodaira, H.: Pvclust: an R package for assessing the uncertainty in hierarchical clustering. Bioinformatics 22(12), 1540–1542 (2006)
18. Bradley, E., Elizabeth, H., Susan, H.: Bootstrap confidence levels for phylogenetic trees. Proceedings of the National Academy of Sciences of the United States of America 93(23), 13429 (1996)
19. Shimodaira, H.: Approximately unbiased tests of regions using multistep-multiscale bootstrap resampling. Annals of Statistics 32(6), 2616–2641 (2004)

On Selecting the Best Pre-processing Method for Affymetrix Genechips

J.P. Florido[1], H. Pomares[1], I. Rojas[1], J.C. Calvo[1], J.M. Urquiza[1],
and M. Gonzalo Claros[2]

[1] Department of Computer Architecture and Computer Technology, University of Granada,
Granada, Spain
`{jpflorido,hector}@ugr.es, {irojas,jccalvo,jurquiza}@atc.ugr.es`
[2] Department of Molecular Biology and Biochemistry, University of Málaga, Málaga, Spain
`claros@uma.es`

Abstract. Affymetrix High Oligonucleotide expression arrays, also known as Affymetrix GeneChips, are widely used for the high-throughput assessment of gene expression of thousands of genes simultaneously. Although disputed by several authors, there are non-biological variations and systematic biases that must be removed as much as possible before an absolute expression level for every gene is assessed. Several pre-processing methods are available in the literature and five common ones (RMA, GCRMA, MAS5, dChip and VSN) and two customized Loess methods are benchmarked in terms of data variability, similarity of data distributions and correlation coefficient among replicated slides in a variety of real examples. Besides, it will be checked how the variant and invariant genes can influence on preprocessing performance.

1 Introduction

Microarray technology is a powerful tool used for the high-throughput assessment of gene expression of thousands of genes simultaneously which can be used to infer metabolic pathways, to characterize protein-protein interactions or to extract target genes for developing therapies for various diseases [1]. Several platforms are currently available, including the commonly used high oligonucleotide-based Affymetrix GeneChip® arrays.

As described in [1], an Affymetrix GeneChip contains probe sets of 10-20 probe pairs representing unique genes. Each probe pair consists of two oligonucleotides of 25 bp in length, namely perfect match (PM) probes (the exact complement of an mRNA) and the mismatch (MM) probes (which are identical to the perfect match except that one base is changed at the center position). The MM probe is supposed to distinguish noise caused by non-specific hybridization from the specific hybridization signal, although some researchers recommend avoiding its use [17].

A typical microarray experiment has biological and technical sources of variation [2]. Biological variation results from tissue heterogeneity, genetic polymorphism, and changes in mRNA levels within cells and among individuals due to sex, age, race, genotype-environment interactions and other “living” factors. Biological variation is of interest to researchers as it reflects true variation among experiments. On the other

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 845–852, 2009.
© Springer-Verlag Berlin Heidelberg 2009

hand, sample preparation, labeling, hybridization and other steps of microarray experiment can contribute to technical variation, which can significantly impact the quality of array data. Therefore, to that systematic non-biological sources of variation mask real biological variation, significant pre-processing is required and involves four steps for Affymetrix GeneChips: background correction, normalization, PM correction and summarization [15].

In this paper, a comparison of some of the most well known pre-defined pre-processing methods (RMA, GCRMA, MAS5, dChip and VSN) and two customized Loess methods is performed. Accuracy of preprocessing is assessed in terms of data variability, similarity in data distributions and correlation among replicates in a variety of real examples. Besides, one of the pre-processing method will be selected to evaluate how the variant and invariant genes have an influence on the quality metrics compared to the whole set of genes.

Section 2 describes the main pre-processing methods existing in the literature for Affymetrix GeneChips and section 3 describes implementations, data sets and results. Conclusions are drawn in section 4.

2 Pre-processing Affymetrix GeneChips

Instead of describing how does work every pre-processing method, they will be compared in each of the four pre-processing steps [3]:

- *Background correction.* It removes unspecific background intensities of scanner images. Three possible algorithms can achieve this correction: the Robust Multi-chip Average (*RMA*) convolution [5] and the *MAS* [6] and *Gcrma* [7] algorithms.
- *Normalization.* It is intended to reduce most of the non-biological differences between chips providing normalized (comparable) signal intensities for every chip [1]. The following methods were applied: *Scaling (constant)* [6], *Quantile* [4], *Loess* [4], *Invariant Set* [8] and Variance Stabilization Method (*VSN*) [9].
- *PM correction.* PM signal intensity should be adjusted to account for nonspecific signal. There are two possible algorithms: *MAS* [6] and *PMonly* [3].
- *Summarization.* It is the final stage in pre-processing Affymetrix GeneChip data and computes expression values from all within-chip replicates by combining the intensities of the 11-20 probe replicates to produce a single expression value for a gene. There are some well-known methods in the literature: *Median Polish* [5], *Tukey Biweight* [6], *Li-Wong MBEI* expression index [10] and *Avgdiff* [11].

The algorithms described above can be found in the Bioconductor *affy*, *affyPLM* and *vsn* packages [3][9][12], which are R libraries of functions and classes.

3 Experiments and Results

In this section, pre-defined and custom pre-processing methods used in the comparison will be explained as well as the data sets and the quality metrics used in the experiment for evaluating such comparison.

3.1 Implementation of Pre-processing Methods

There are six different functions (Table 1) to calculate all pre-processing methods:

- *expresso()* (*affy* package [3][12]), which provides quite general facilities for computing expression summary values.
- *rma()* (*affy* package [3][12]) to calculate only the RMA method.
- *threestep()* (*affyPLM* package [12]), which provides the user the ability to compute very general expression measures.
- *gcrma()* (*affyPLM* package [12]), which computes the GCRMA method.
- *mas5()* (*affy* package [3][12]) to calculate the MAS5 method.
- *vsnrma()* (*vsn* package [9][12]) to calculate only the VSN method.

Table 1. Summary of microarray data pre-processing methods used in this paper and the R function utilized for its calculation

Methods	*Background Correction*	*Normalization*	*PM correction*	*Summarization*	*Functions* [a]
RMA	rma	quantiles	pmonly	medianpolish	expresso(), rma(), threestep()
GCRMA	gcrma	quantiles	pmonly	medianpolish	gcrma(), threestep()
MAS5	mas	mas	mas	mas	expresso(), mas5()
VSN		vsn	pmonly	medianpolish	expresso(), vsnrma()
dChip		invariantset	pmonly	liwong	expresso()
Loess 1	mas	loess	mas	mas	expresso()
Loess 2		loess	pmonly	avgdiff	expresso()

[a] Note that a pre-processing method can be used with more than one function, usually from different R libraries.

3.2 Data Sets

Five different data sets covering different experimental designs (from control/ treatment to a time series, with different replications) were used:

- *Dilution experiment*. Two sources of cRNA A (human liver tissue) and B (Central Nervous System cell line) have been hybridized to human array (*HGU95A*) in a range of proportions and dilutions. The data is available at the *affydata* package [12] and detailed description can be accessed there.
- *Estrogen*. The package *estrogen* [12] contains 8 Affymetrix *HG-U95Av2* CEL files from an experiment involving estrogen receptor positive (ER+) breast cancer cells [13]. Estrogen effect on time is evaluated.
- *Pig infection.* This experiment contains 10 samples run on *Porcine* Affymetrix Genechip arrays in which 6 pigs were treated with an infectious agent and the remaining ones were not treated. These unpublished data were kindly given by Prof. Juan José Garrido (Department of Genetics, University of Córdoba, Spain).
- *Cell infection*. This experiment was also kindly given by Prof. Juan J. Garrido and consists of 18 Affymetrix *Porcine* CEL files. It contains the infection of 2 cell lines (IPI and IPEC) with a specific pathogen for each one. Each infection is sampled at 0, 2 and 4 h of infection by triplicate. The IPI and IPEC strains have been treated separately since they are different cell lines treated with different pathogens.

- *CLL.* The CLL package [12] contains the chronic lymphocytic leukemia (CLL) gene expression data. The CLL data had 24 samples run on *HG-U95Av2* Affymetrix GeneChip arrays that were either classified as progressive or stable in regards to disease progression.

3.3 Quality Metrics

In order to evaluate the performance of the pre-processing methods, three different metrics have been used:

- *Replicate variability* [14]. It is based on the assumption that expression level of a gene should ideally remain the same across multiple replicated slides. Variability is measured by the mean of the standard deviation over all genes. Smaller mean is indicative of better pre-processing.
- *Kolmogorov-Smirnov (K-S) test* [14]. It is a goodness-of-fit test of two continuous distributions and it is based on the hypothesis that an effective normalization procedure should result in two similar, ideally identical, distributions with a small, ideally zero-valued K-S statistic. On the other hand, two different distributions will generate a large K-S statistic.
- *Spearman Rank Correlation Coefficient* [15]. It is based on the comparison of the correlation coefficient between replicated slides assuming that, given an experiment, the correlation coefficient among replicated slides will be increased after the pre-processing stage.

3.4 Results

Raw expression data. The different pre-processing methods should be compared with the raw expression data in terms of the quality metrics described above. Thus, all the experiments were run with no background correction, no normalization and no PM correction. Just the summarization step is needed to obtain gene expression values. Three different summarization algorithms using the *expresso* function were tested: *median-polish*, *Tukey-Biweight* and *AvgDiff.*

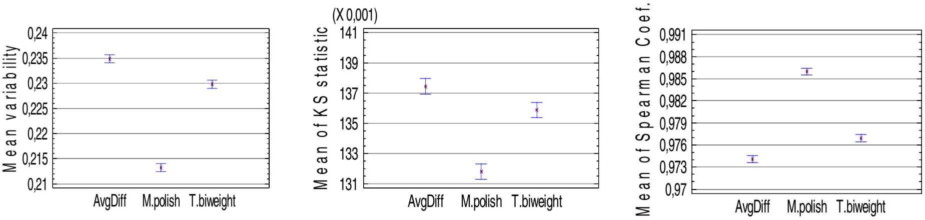


Fig. 1. Means and 95% Least Significant Differences (LSD) Intervals of the different summarization algorithms through the quality metrics

For each quality metric, a two-way Analysis Of Variance (ANOVA) test checked the statistical significance of the results. Fig.1 shows that *median-polish* summarization method performs significantly better ($P < 0.05$). Hence, it has been selected to obtain raw expression values for comparing with all pre-processing methods.

Comparison of functions for pre-defined pre-processing methods. Since there are several pre-defined pre-processing methods (RMA, GCRMA, MAS5 and VSN) that can be used with more than one function (Table 1), selection of one function for each method is a must. dChip pre-processing method could not be included: it does not converge for most of the data sets since a minimum of 10 replicates arrays is recommended [12]. Thus, for each quality metric, a one-way ANOVA test was run for the different functions within a pre-processing method and no statistical difference ($P > 0.05$) among functions for the same method is found. Hence, the decision was taken from the average running time of a total of 10 executions (Fig. 2). As a result, *rma()* performs faster for RMA, *gcrma()* and *threestep()* for GCRMA are practically the same (although the latter will be used due to it provides the user with a great deal of control), *mas5()* is the fastest for MAS5 and *vsnrma()* is the best for VSN.

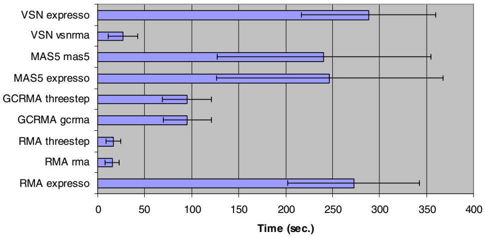


Fig. 2. Average and standard dev. running time of pre-processing functions for all data sets

Comparison of different pre-processing methods. For each quality metric, a two-way ANOVA test was used to check how each pre-processing method performs in all data sets. According to the mean variability, the *VSN* method gets the best results (Fig. 3a). *VSN*, *RMA*, *GCRMA* and *Loess 2* are statistically better ($P < 0.05$) than the raw data. The *VSN* performance was not unexpected because it specifically aims to stabilize the variance across the replicated arrays. On the other hand, *MAS5* and *Loess 1* obtain worse results than the raw data. Since both methods are identical but for normalization step, it can be concluded that the normalization method alone cannot account for the pre-processing performance.

With regard to the mean Kolmogorov-Smirnov statistic (Fig. 3b), *RMA* and *GCRMA* get the best results, followed by *Loess 2*, *VSN*, *MAS* and *Loess 1*, and all of them perform significantly better ($P < 0.05$) than raw data. *RMA* and *GCRMA* perform the best since the use of *Quantile* normalization algorithm (see section 2) that forces the empirical distributions in different slides to be identical.

Finally, according to the mean Spearman rank coefficient (Fig. 3c), *RMA*, *VSN* and Raw data performs better with no statistical difference among them. This means that no significant improvement has achieved when pre-processing the data in terms of correlation.

In conclusion, a pre-processing method is considered appropriate when (i) its mean variability is lower than for the raw data; (ii) the mean K-S statistic is lower than for the raw data; and (iii) the Spearman rank coefficient is higher than for the raw data. Only RMA and VSN method seems to fulfill these rules, although in (iii) the difference is not statistically significant.

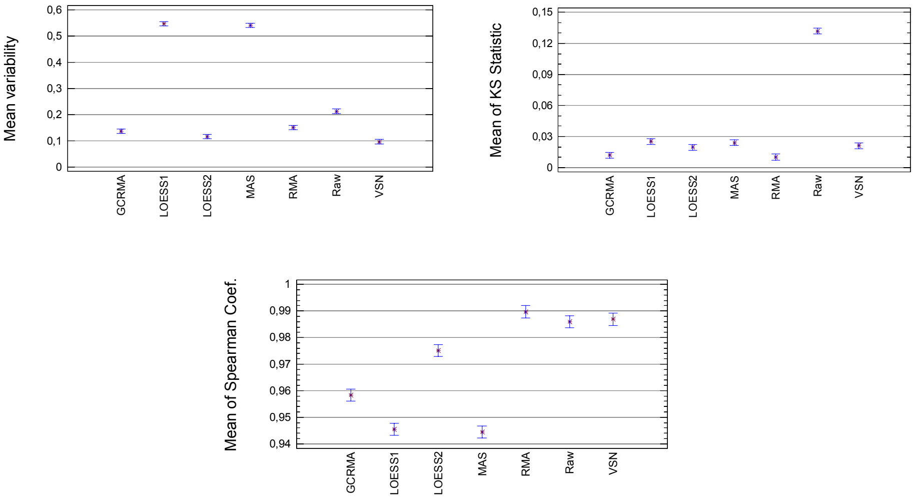


Fig. 3. Means and 95% LSD intervals of the different pre-processing methods through the quality metrics: a) mean variability, b) mean of K-S statistic, c) mean of Spearman Coefficient

The influence of variant and invariant genes with respect the whole set of genes. It would be interesting how the variant and invariant genes can influence on pre-processing performance. Invariant and highly-variant genes were inferred from raw summarized data. Two sets (A and B) of replicated slides must be selected. For each set, the mean value for each gene in all replicates is obtained. Then, a list of interesting genes is obtained according to M = abs(log2(A/B)), so that the n genes (1000 in our case) with the highest value of M will be the variant genes and the n genes with the lowest value of M, closest to zero, will be the invariant ones. These groups were pre-processed with *RMA*, due to its great performance shown in the previous subsection. Raw summarized data were utilized as controls using these groups as well as the pre-processing with *RMA* and raw summarized data (*median-polish*) using the whole set of genes. Statistical significance was assessed by a three-way ANOVA test.

Mean variability and K-S statistic (Fig. 4a, 4b) reveal that any set of genes pre-processed with RMA is statistically more homogeneous ($P < 0.05$) than raw data (which only have been summarized by *median-polish*). This improvement, as expected due to *quantile* normalization, is more aggresive for the K-S statistic. Moreover, pre-processing whole data is statistically equivalent ($P > 0.05$) to pre-processing only the invariant set of genes, while the difference is clearly significant ($P < 0.05$) when only the variant set is pre-processed. This result is not surprising, since it is assumed that most of the genes in a microarray experiment are non-differentially ones, i.e. they are invariants. The mean Spearman rank coefficient (Fig. 4c) also reveals that pre-processing an invariant set of genes with RMA is equivalent to do the same with the whole data. However, it is noticeable that invariant and all genes datasets do not show a great improvement after the pre-processing while variant genes do it clearly, showing that pre-processing does not break (even it can improve it slightly) the correlation among data replication. Therefore, the variant set improves the correlation with pre-processing since they are supposed to be more spread before the pre-processing. Hence, it has been confirmed that pre-processing with an invariant dataset is similar to pre-processing the whole chip provided that most of genes in chip are not differentially expressed and it is also confirmed that pre-processed data are preferable to raw data.

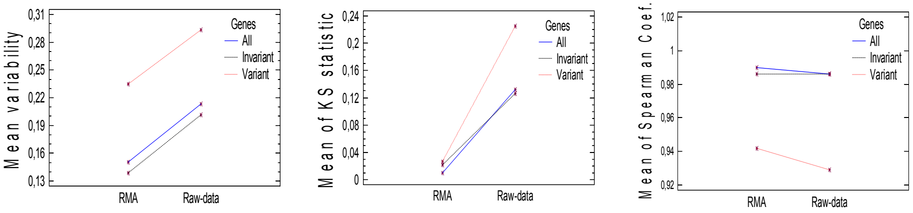


Fig. 4. Interaction plots of methods and genes in terms of the quality metrics: a) mean variability, b) mean of K-S Statistic, and c) mean of Spearman Rank Coefficient

4 Conclusions and Future Work

In this paper, we performed a comparison of some of the most well known pre-processing methods (RMA, GCRMA, MAS5 and VSN) and two custom implementations of LOESS in terms of quality metrics such as data variability, similarity of data distributions, and correlation coefficient among replicated slides. Five real datasets obtained from different laboratories, with different design and with different number of replica were employed, which provide a strong support for the conclusions since they are not linked to particular data. Implemented functions for preprocessing are time-efficient but for *expresso()* and *median-polish* is the summarization method that seems to perform better. According to our three golden rules for an adequate pre-processing method, only VSN and RMA seems to normalize data in such a way that the results are better devoid of experimental and technical errors. Indirectly, it has been shown that *Loess* normalization performance is highly dependent on the rest of pre-processing steps. When an invariant gene set is pre-processed with RMA, its behaviour is similar than when the whole chip is pre-processed with the same method since most of the genes in a microarray are not expected to be differentially expressed. Moreover, most of the genes in a microarray experiment are expected to be correlated even if no pre-processing method is applied.

As a future work, a more complex system is desired: apply supervised learning models such as Artificial Neural Networks with the aim of recognize data patterns to decide the best pre-processing method for a given data set [16]. We are also searching for a function that combines the three quality metrics to obtain an objective way to evaluate which is the best way of pre-processing for a given set of data.

Acknowledgment

This paper has been supported by the Spanish Ministry of Education and Science under project TIN2007-60587 and the FPU research grant AP2007-03009.

References

1. Hochreiter, S., Clevert, D.A., Obermayer, K.: A new summarization method for affymetrix probe level data. Bioinformatics 22(8), 943–949 (2006)
2. Zakharkin, S.O., Kim, K., Mehta, T., et al.: Sources of variation in Affymetrix microarray experiments. BMC Bioinformatics 6, 214 (2005)
3. Gautier, L., Cope, L., Bolstad, B.M., Irizarry, R.A.: affy-analysis of Affymetrix GEneChip data at the probe level. Bioinformatics 20(3), 307–315 (2003)
4. Bolstad, B.M., et al.: A Comparison of normalization methods for high density oligonucleotide array data based on bias and variance. Bioinf. 19, 185–193 (2002)
5. Irizarry, R.A., Hobbs, B., Collin, F., et al.: Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics 4, 249–264 (2003)
6. Affymetrix Microarray Suite Users Guide, Affymetrix, Santa Clara, v.5.0 edn. (2001)
7. Wu, Z., Irizarry, R., Gentleman, R., et al.: A model based background adjustement for oligonucleotide expression arrays. Journal of the American Statistical Association (2005)
8. Schadt, E.E., et al.: Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. J. Cell Biochem. Suppl., 120–125 (2001)
9. Huber, W., et al.: Variance stabilization applied to microarray data calibration and to the quantification of differential expression. Bioinformatics 18(suppl. 1), S96–S104 (2002)
10. Li, C., Wong, W.H.: Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. Proc. Natl. Acad. Sci. USA 98(1), 31–36 (2001)
11. Affymetrix Microarray Suite Users Guide. Affymetrix, Santa Clara, v.4.0 edn. (1999)
12. Gentleman, R.C., et al.: Bioconductor: open software development for computational biology and bioinformatics. Genome Biology 5(10), Article R80 (2004)
13. Scholtens, D., Miron, A., Merchant, F., et al.: Analyzing factorial designed microarray experiments. Journal of Multivariate Analysis 90, 19–43 (2004)
14. Xiong, H., Zhang, D., Martyniuk, C.J., et al.: Using Generalized Procrustes Analysis (GPA) for normalization of cDNA microarray data. BMC Bioinformatics 9(25) (2008)
15. Lim, W.K., Wang, K., et al.: Comparative analysis of microarray normalization procedures: effects on reverse engineering gene networks. Bioinformatics 23, 282–288 (2007)
16. Rojas, I., Pomares, H., et al.: Analysis of the functional block envolved in the design of radial basis function networks. Neural Processing Letters 12(1), 1–17 (2000)
17. Irizarry, R.A., Bolstad, B.M., Collin, F., et al.: Summaries of Affymetrix GeneChip proble level data. Nucleic Acids 31(4) (2003)

Method for Prediction of Protein-Protein Interactions in Yeast Using Genomics/Proteomics Information and Feature Selection

J.M. Urquiza, I. Rojas, H. Pomares, J.P. Florido, G. Rubio, L.J. Herrera,
J.C. Calvo, and J. Ortega

Department of Computer Architecture and Computer Technology,
University of Granada, 18017 Granada, Spain
`jurquiza@atc.ugr.es`

Abstract. Protein-protein interaction (PPI) prediction is one of the main goals in the current Proteomics. This work presents a method for prediction of protein-protein interactions through a classification technique known as Support Vector Machines. The dataset considered is a set of positive and negative examples taken from a high reliability source, from which we extracted a set of genomic features, proposing a similarity measure. Feature selection was performed to obtain the most relevant variables through a modified method derived from other feature selection methods for classification. Using the selected subset of features, we constructed a support vector classifier that obtains values of specificity and sensitivity higher than 90% in prediction of PPIs, and also providing a confidence score in interaction prediction of each pair of proteins.

1 Introduction

In the current research in Proteomics, one of the most important areas is the elucidation of the structure, interactions and functions of proteins that constitute cells and organisms. Most of fundamental biological processes involve protein-protein interaction [10].

The information of interactions can be obtained both through experimental methods or computational approaches [19]. Among the studied organisms, S. cerevisiae is the most widely analysed organism even though its interactome is still far to complete [2,19]. However, it is thought that protein-protein interaction data obtained from high-throughput experiments have a high number of false positives, i.e., interactions that are spurious or biologically irrelevant and do not occur in the cell [2]. In the literature there are different methods for proteins interaction prediction. As in other previous works, we have adopted an approach based in Support Vector Machines [1,5].

Supervised classification methods reckon on a set of positive and negative examples for training. Building a intelligent classifier for PPI relies on the several existing approaches that contribute in several grades of precision and bias to get

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 853–860, 2009.
© Springer-Verlag Berlin Heidelberg 2009

positive and negative pairs of proteins [10,15,11]. Specifically negative examples are quite difficult to report [19]. In this work, we have used a reliable set of positive and negative pairs taken from Saeed el al. [19] for yeast.

In this work, we propose a predictor of protein-protein interaction in yeast (S. cerevisiae) using Support Vector Machines (SVM). A genomic/proteomic features extraction process [10,15] was performed to obtain a total of 26 features introducing a similarity measure. Subsequently, we obtained the most relevant features by means of our proposed feature selection method derived from the methods studied in Gilad-Bachrach et al. et [17]. Finally, we constructed a SVM predictors that returns a confidence score in classification of each pair of proteins and obtains results higher than 90% of specificity and sensibility in the prediction of PPIs.

2 Features Extraction

The data set in this work have been obtained from Saeed et al. [19]. They provide a set of 4 million negative pairs of proteins and a set of 4809 interacting pairs of proteins for yeast. We have used all positive examples (4809) and a random subset of 4895 negative examples, giving a total of 9704 pairs.

From this data set, we extracted 26 genomic/proteomic features of yeast from diverse databases. For each pair, we have mined the information of following databases:

- SwissPfam (version 22.0) [7] contributes information of protein domains.
- Gene Ontology Annotation (GOA) Database [8] provides high quality annotation of Gene Ontology (GO) [4] (version may 2008). GOA classifies protein annotation in three ontologies: biological processes (P), cellular components (C) and molecular functions (F)
- MIPS Comprehensive Yeast Genome Database CYGD (version June 2008) gathers information of molecular estructure and functional network in yeast [9]. We consider all catalogues: functional, complexes, phenotype, proteins and subcellular compartments
- HINTdb (version 13 June 2007) that provides the homology pairs of proteins[16].
- 3did database (version 25 may 2008). [20] that gives a collection of domain-domain interactions in proteins which 3D-structures are known in Protein Data Bank (PDB) [6].

The fundamental idea for our features extraction is to find the number of common terms in a particular database or catalogue between two proteins, as we consider that each protein has a number of associated terms in each database. Furthermore, we introduce a similarity measure that shows the percentage of common terms in a pair with respect to the sum of all terms which have two given proteins, for the databases specified next. This is a measure called "local". Its equation would be 1:

$$local\ similarity = \frac{number\ of\ sharedelements}{number\ of total\ elements\ in\ a\ pair\ of\ proteins} \qquad (1)$$

A different measure is also proposed that shows the percentage of common terms in a pair with respect to the sum of all terms in a database, it is called "global similarity".

We explain now the 26 used features for yeast, indicating between parenthesis their order:

- Common GO annotation terms between two proteins of three ontologies (P,C,F) together (1^{st}) and their "local" and "global" similarity measures (13^{st}, 14^{st}). Moreover we considered each separately ontology (5^{st} P, 6^{st} C, 7^{st} F) and their respective "local similarity" measure (16^{st}, 17^{st}, 18^{st}) and "global ones" (19^{st}, 20^{st}, 21^{st}).
- Number of homologous between two proteins obtained from HintDB (2^{nd}).
- Commons Pfam domains between two proteins, extracted from SwissPfam, which are found in 3did database (3^{rd}). The similarity would be the number of common domains Pfam-3did between the total of Pfam domains that have two proteins (15^{st}).
- mRAN expression value from Rosetta compendium extracted from Jansen et al. work [10] (4^{st}).
- Considering each MIPS catalogue separately, we counted the number of common terms (catalogue identifier) between two proteins (functional 8^{st}, complexes $9^{s}t$, proteins 10^{st}, phenotypes 11^{st}, subcellular compartments 12^{st}). Furthermore we considered their "local similarity" measures (22^{st}, 23^{st}, 24^{st}, 25^{st}, 26^{st}).

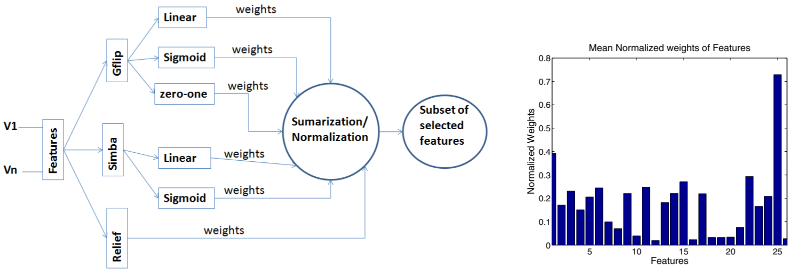
3 Feature Selection

In this work, we propose a novel method based on 3 feature selection method: G-flip, Simba [17] and Relief [14]. They use a margin based criterion to measure quality of sets of features. G-flip and Simba assign a score to sets of features according to the induced margin from an evaluation function. This GFLIP and SIMBA evaluation function requires a utility function; we used three functions proposed by Navot et al. [17]: linear, sigmoid and zero-one. The linear utility function is a simple sum of margins. The Zero-one utility (not for SIMBA) equals 1 when the margin is positive and 0 otherwise. And finally, the sigmoid utility function is proportional to the leave-one-out (LOO) error.

Our proposed feature selection method consists of normalizing the weights between [0,1] of all previous methods results (Gflip, Simba and Relief). Subsequently, the final weight for each feature is calculated as a mean of all weights of all methods. The feature with a final weight above the mean are the features selected to perform the classification (see figure 1(a)).

4 Support Vector Machines

Support Vector Machines (SVM) is a classification and regression paradigm developed by Vapnik and his group at AT&T Bell Laboratories [3,12]. The SVM



(a) Our proposed selection of features

(b) Our proposal: Mean Weights of features

Fig. 1. Our proposal: selection of features 1(a), Feature Weights 1(b)

approach learns a separating hyperplane to maximize the margin and it gives a good generalization performance. Therefore SVM is a prominent large margin algorithm [17]. SVM is originally a model for the binary-class classification, thus it is straightforward to use it in the present problem for classification among protein-protein interaction and non protein-protein interaction. SVM is being widely used in Bioinformatics due to their effectiveness [1,5].

In this work, we have adopted a SVM approach to construct the classifiers. In order to perform the predictors, we divided randomly data set in two groups: training and test. We propose a score in prediction which is the difference of probabilities in absolute value returned by SVM for each pair of proteins, that could be used as measure of confidence. SVM classifies the pairs returning two probability values that represent the chance to be an interacting or non-interacting pair. A pair of proteins will be classified in positive or negative interaction according to best probability.

5 Results

Following the steps of our experiment, we extracted all features from the given problem and its set of examples, and applied our feature selection method to obtain the subset of relevant features.

Due to computational needs we use randomly 70% of data for feature selection methods. Simba and Gflip were executed over 10 iterations applying all their utility functions for evaluation function. Finally, features that were selected 10 times in Gflip were chosen. On the other hand, in Simba the features were selected according to mean weights in all executions. Relief is deterministic. Our proposed method calculated a mean weight per feature derived from the weights of these methods. Table 1 shows the selected features by method, and figure 1(b) shows the final features weights for our proposed method.

In the next step, the data again was randomly divided in two groups: 70% pairs to train the SVMs and so getting the classifiers, and 30% pairs to test the models and to evaluate them. Two types of kernels were used for SVM classification:

Table 1. Table of selected features by different selection methods

Method	Selected Features
Gflip Linear	1 2 3 5 6 7 8 10 11 13 15 16 17 18 19 22 23 24 25
Gflip Sigmoid	1 2 3 4 5 6 7 8 9 10 11 13 15 16 17 18 20 21 22 23 24 25
Gflip zero-one	1 2 25
Relief	25 22 11 1 13 5 23 2 24
Simba Linear	3 15 22 25
Simba Sigmoid	1 3 4 6 11 15 22 23 25
Our Proposal	1 3 5 6 9 11 13 14 15 17 22 24 25

linear and RBF (Radial-Basis Function) [18]. A single combination of calculated parameters γ and C for RBF Kernel and C for linear kernel was employed. The performance of the test set was evaluated by sensitivity and specificity. We executed three times our predictors with three different randomly groups for training and test. We obtain for linear-SVMs: classification error 6,63 % ± 1,37, sensitivity 97,4 % ± 0,37 and specificity 89,17 % ± 2,5. For RBF-SVMs the results are: classification error 5,03 % ± 0,25, sensitivity 98,3 % ± 0,36, specificity 91,5 % ± 0,88.

In figure 2, we can observe the classification score of a test group for correct and wrong predicted pairs, determining clearly two groups. Correct predicted pairs obtain score 0.905 ± 0.149, and as we expected the wrong predicted pairs obtain 0.051 ± 0.220. So we can say that our SVM predictor is reliable, returning high score for correct classified ones.

To check that the selected features calculated by our similarity measure help to improve the predictive power of SVM, we performed a comparison between the three proposed features by Patil et al. [15] (d,g,h) and three highest weighted features (25^{st}, 1^{st}, 22^{st}) with our approach (see figure 1(b)). Table 2 shows the results of this comparison.

As it can be observed in table 2, in general, our selection of three features does not improve the obtained results with the proposed features by Patil et al. [15]. However, we notice that more than 3 features are necessary to provide a correct classification, giving higher values of specificity and sensibility.

Table 2. Comparison table with results of Selected Features

LINEAR SVM	Our proposal with best features appeared in the bibliography [15]			Results by Patil [15]			Our Proposal with the 3 best features		
Features	Error (%)	Sv. (%)	Sp. (%)	Sv. (%)	Sp. (%)	Features	Error (%)	Sv (%)	Sp (%)
d+g+h	7,03±0,47	92,53±0,52	93,4±0,42	12,3	99,4	1+2+3	11,63±0,05	95,13±0,09	81,23±0,19
d+g	15,73±0,61	92,6±0,57	75,5±0,57	14,5	99,3	1+3	14±0,14	99,83±0,05	71,47±0,52
d+h	20,5±0,85	99,9±0	58±1,27	14,7	99,2	1+2	12,7±0	92,6±0,57	81,73±0,75
d	34,3±0,42	67,57±0,66	63,77±0,24	14,8	99,2	1	14,37±0,05	99,83±0,05	70,67±0,38
g+h	7,03±0,47	92,53±0,52	93,4±0,42	44,1	94	2+3	14,7±0,42	95,13±0,09	74,9±0,42
g	15,73±0,61	92,6±0,57	75,5±0,57	86,7	74,3	3	34,43±0,38	38,93±0,05	93,7±0
h	20,5±0,85	99,9±0	58±1,27	89,7	62,9	2	15,73± 0,61	92,6±0,57	75,5±0,57
RBFs SVM	Our proposal with best features appeared in the bibliography [15]			Results by Patil [15]			Our Proposal with the 3 best features		
Features	Error (%)	Sv. (%)	Sp. (%)	Sv. (%)	Sp. (%)	Features	Error (%)	Sv (%)	Sp (%)
d+g+h	7,03±0,47	92,53±0,52	93,4±0,42	12,3	99,4	1+2+3	7,53±0,24	98,9±0	85,67±0,66
d+g	15,37 ± 0,8	91,1±0,85	77,8±2,26	14,5	99,3	1+3	8,77±0,19	99,73±0,05	82,27±0,52
d+h	20,5±0,85	99,9±0	58±1,27	14,7	99,2	1+2	8,47±0,33	98,47±0,09	84,27±0,8
d	34,37±0,47	67,57±0,66	63,67±0,24	14,8	99,2	1	10,83±0,38	99,83±0,05	77,97±0,94
g+h	7,03±0,47	92,53±0,52	93,4±0,42	44,1	94	2+3	14±0,42	95,57±0,19	75,97±0,47
g	15,73±0,61	92,6±0,57	75,5±0,57	86,7	74,3	3	24,73±0,09	99,9±0	49,23±0,75
h	20,53±0,83	99,9±0	58±1,27	89,7	62,9	2	15,73±0,61	92,6±0,57	75,5±0,57

Sv: sensitivity, Sp: Specificity. 1: (25^{th} feature) Phenotype MIPS calculated by our "local similarity" measure, 2:($1^{st} feature$) similar GO annotations, 3: (22^{th} feature) Functional MIPS calculated by our "local similarity" measure .d: interacting Pfam domains (3^{rd} feature); g: similar GO annotations (1^{st} feature) ; h: homologous interactions (2^{st} feature). More than one features are indicated by listing the features separated by a '+' sign.

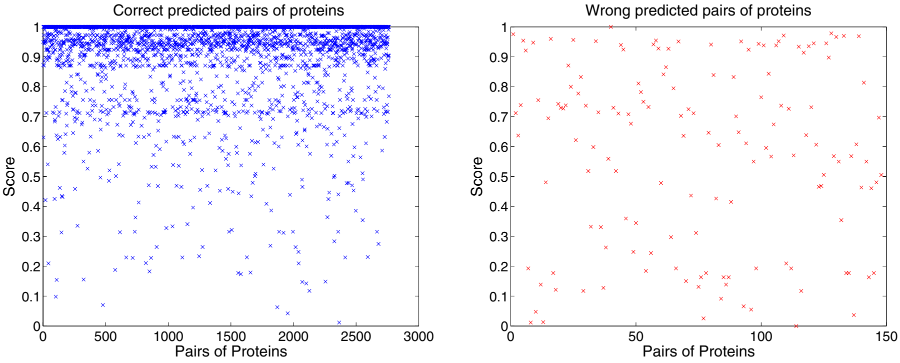


Fig. 2. Scores for correct predicted pairs (left) and not correct predicted pairs (right)

Though a ROC (Receiver Operating Characteristic) curve (see figure [3]), using our score as measure of prediction, we represented the results of the RBF-SVM classifier using the 14 selected features by our feature selection approach (red) versus the best results of our RBF-SVM approach with the "d+g+h" features proposed by Patil et al. [15] (blue). We observe better levels of sensibility and specificity using our 14 features. Hence, using a reasonable subset of features from the 26 extracted characteristics, we improve the prediction than using only 3.

Two of the three most important features (25th, 22th) considered by our proposal are derived from our similarity measure for phenotype and functional MIPS catalogue. We found that features related with domains have not a important

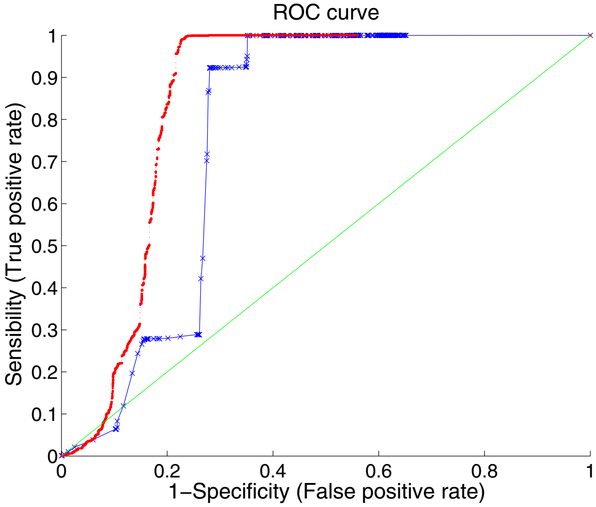


Fig. 3. ROC curves. Our proposal (red) and a RBF-SVM with features appeared in Patil et al. [15] (blue).

weight in our selection and the features related to GO annotations have a important weight. Anyway, we confirm that a combination of a subset of relevant features permit the construction of a reliable classifier. The designed SVM classifiers with the selected subset of features show to be good predictors.

Finally, we mention that the linear kernel and the RBF kernel obtain a reliable prediction in the test set. However as we expected, the RBF kernel is slightly more reliable.

6 Conclusion

The elucidation of protein-protein interactions is a important goal in the current Proteomics research. There is a large number of approaches for PPI prediction that use several genomic/proteomic features, utilizing a set of reliable examples and applying SVM [1,5].

In this work, we have proposed the usage of features based on a new similarity measures for yeast extracted from known databases: SwissPfam, GOA, MIPS, 3did, Hintdb. The data set that we have used is a set of examples extracted from a reliability source for yeast [19]. Through a combination of the outcomes of tree feature selection methods, we have performed a characteristic relevance weighting, finally selecting the features with relevance above the mean. With the selected subset we constructed a linear-SVM and a RBF-SVM as effective predictors, which may provide a confidence score in classification. Hence, we have checked that the features derived from our similarity measure improved the predictive power of the classifiers, obtaining high specificity and sensibility in prediction of PPIs.

For further work, we think that it could be interesting an analysis with several and a larger size of data since Saeed et al. [19] provide a set of more than 4 million negative examples, even extending our approach to others organisms. Moreover, it is possible to apply others variable selection methods as well as using other machine learning methodologies such as fuzzy systems that provide interpretable solutions [13]. Because of high computational needs of these algorithms, we consider the possibility of applying parallelism to our approach.

Acknowledgments

José Miguel Urquiza Ortiz was supported by FPU research grant AP2006-01748 from Spanish Ministry of Science. This work has been partially supported by the Spanish CICYT Project TIN2007-60587 and Junta de Andalucia Project P07-TIC-02768.

References

1. Ben-Hur, A., Noble, W.S.: Kernel methods for predicting protein-protein interactions. Bioinformatics (2005)
2. Bork, P., et al.: Protein interaction networks from yeast to human. Current Opinion in Structural Biology (2004)

3. Vapnik, V., Cortes, C.: Support vector network. Mach. Learn. (1995)
4. Gene Ontology Consortium. The gene ontology (go) database and informatics resource. Nucl. Acids Res. (2004)
5. Craig, L.L.: Improving protein protein interaction prediction based on phylogenetic information using a least-squares support vector machine. Ann. N Y Acad. Sci (2007)
6. Berman, H.M., et al.: The protein data bank. Nucleic Acids Res. (2000)
7. Boeckmann, B., et al.: The swiss-prot protein knowledgebase and its supplement trembl in 2003. Nucleic Acids Research (2003)
8. Camon, E., et al.: The gene ontology annotation (goa) database: sharing knowledge in uniprot with gene ontology. Nucleic Acids Research (2004)
9. Güldener, U., et al.: Cygd: the comprehensive yeast genome database. Nucleic Acids Research (2005)
10. Jansen, R., et al.: A bayesian networks approach for predicting protein-protein interactions from genomic data. Science (2003)
11. Wu, X., et al.: Prediction of yeast protein-protein interaction network: insights from gene ontology and annotations. Nucleic Acids Research (2006)
12. Herrera, L.J., Pomares, H., Rojas, I., Guillén, A., Prieto, A., Valenzuela, O.: Recursive prediction for long term time series forecasting using advanced models. Neurocomputing 70, 2870–2880 (2007)
13. Herrera, L.J., Pomares, H., Rojas, I., Valenzuela, O., Prieto, A.: Tase, a taylor series based fuzzy system model that combines interpretability and accuracy. Fuzzy Sets and Systems 153, 403–427 (2005)
14. Kira, K., Rendell, L.: A practical approach to feature selection. In: Proc. 9th International Workshop on Machine Learning (1992)
15. Patil, A., Nakamura, H.: Filtering high-throughtput protein-protein interaction data using a combination of genomic features. BMC Bioinformatics (2005)
16. Nakamura, H., Hint, P.A.: A database of annoted protein-protein interactions and their homologs. Biophysics (2005)
17. Gilad-Bachrach, A.N.R., Tishby, N.: Margin based feature selection: Theory and algorithms. In: Proc. of the 21'st ICML (2004)
18. Rojas, I., Pomares, H., Gonzáles, J., Bernier, J.L., Ros, E., Pelayo, F.J., Prieto, A.: Analysis of the functional block involved in the design of radial basis function networks. Neural Process. Lett. 12(1), 1–17 (2000)
19. Saeed, R., Deane, C.: An assessment of the uses of homologous interactions. Bioinformatics Advance Access (2007)
20. Aloy, P., Stein, A., Russell, R.B.: 3did: interacting protein domains of known three-dimensional structure. Nucleic Acids Res. (2005)

Protein Structure Prediction by Evolutionary Multi-objective Optimization: Search Space Reduction by Using Rotamers

J.C. Calvo, J. Ortega, M. Anguita, J.M. Urquiza, and J.P. Florido

Department of Computer Architecture and Technology
CITIC-UGR
University of Granada
{jccalvo,julio,manguita,jurquiza}@atc.ugr.es, jpflorido@ugr.es

Abstract. The protein structure prediction (PSP) problem is considered an open problem as there is no recognized "best" procedure to find solutions. Moreover, this problem presents a vast search space and the analysis of each protein conformation requires significant amount of computing time. We propose a reduction of the search space by using the dependent rotamer library. Also this work introduces new heuristics to improve the multi-objective optimization approach to this problem.

1 Introduction

Proteins have important biological functions such as the enzymatic activity of the cell, attacking diseases, transport and biological signal transduction, among others. They are chains of amino acids selected from a set of twenty elements. Whenever an amino acid chain is synthesized, it folds together and uniquely determines its 3D structure. Moreover, although the amino acid sequence of a protein provides interesting information, the functionality of a protein is exclusively determined by its 3D structure [1]. Thus, there is a high interest in knowing the 3D structure of any given protein. For example, as some proteins attack diseases, taking into account that protein functionality depends on their structure, it would be important to know this 3D structure to learn its correlation with the protein functionality. Thus, the protein 3D structure can aid in the design of efficient drugs. In this way, many useful applications involve determining the protein structure, and many disciplines, such as Medicine, Biochemistry, Biology and Engineering can take advantage of solutions to this problem.

It is possible to reach the 3D structure of a protein experimentally by using methods such as X-ray crystallographic and nuclear magnetic resonance (NMR). Nevertheless, these processes are quite complex and costly as they would require months of expert work and laboratory resources. This situation comes clear if considering that less than a 25% of the protein structures included in the PDB (Protein Data Bank) have been solved.

An alternative approach is to use high performance computing. Nevertheless, the computational analysis of each conformation requires a significant time and

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 861–868, 2009.
© Springer-Verlag Berlin Heidelberg 2009

this is a Grand Challenge Problem that still remains unsolved [1]. This computer approach is called protein structure prediction (PSP) and implies predicting the tertiary structure of a protein given its primary structure.

The protein structure can be divided into four levels (Fig. 1). *The primary structure* is the sequence of amino acids that define the order of the sequence. *The secondary structure* is a set of contiguous amino acids joined by some hydrogen bonds. Then, the super-secondary structure is the combination of two secondary structures by a short connecting peptide. *The tertiary structure* is a three-dimensional structure of a single sequence of a protein. All force-field atoms take part in this conformation. Finally, *the quaternary structure* refers to a protein built by two or more sequences. This structure defines the relations between the different sequences of the protein. Thus, the PSP tries to determine how the primary structure translates into the tertiary structure.

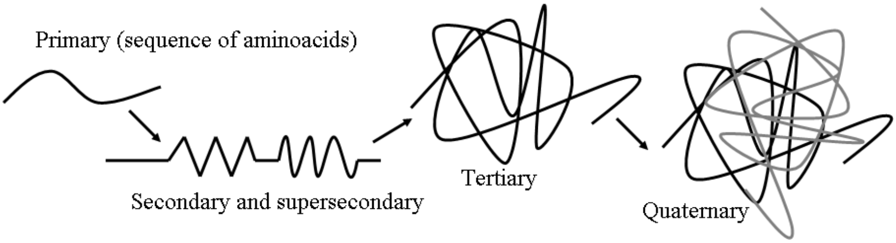

Primary (sequence of aminoacids)
Secondary and supersecondary
Tertiary
Quaternary

Fig. 1. Protein structures. From primary structure to quaternary structure.

The paper has been structured as follows. In Section II we introduce the concepts related with multi-objective optimization that we are going to use throughout of the work. The next section defines the different methods that our multi-objective optimization process needs in this problem. In the Section IV we provide the results and compare our algorithms with other approaches, and Section V exposes the conclusions.

2 PSP as an Optimization Process

Computational approaches to PSP can be divided into three main alternatives: comparative modeling, fold recognition and *ab initio* procedures [2,3]. The first two alternatives consider that similar primary structures will fold into similar ways and try to take advantage of known structures for similar amino acid sequences (homologue sequences). The *ab initio* alternative does not require any homology and can be applied when an amino acid sequence does not correspond to any other known one. This paper uses the *ab initio* approach. In this case, to predict the protein 3D structure, we need to know the relation between primary and tertiary structures. Although this relation is not trivial, and there are many factors affecting the folding process that produces the final 3D conformation, we are looking for the native tertiary conformation with minimum free energy. Therefore, we do not take into account external factors that may influence the protein folding process, such as temperature, neighbor proteins, and other conditions into the cell. We are considering Protein Structure Prediction (PSP) rather than Protein Folding (PF).

Many algorithms have been proposed to solve the PSP by optimizing an objective or energy function, usually by using evolutionary algorithms [4,5]. Nevertheless, over the last few years, some new approaches have been suggested that model the PSP problem as a multi-objective problem [1,6]. There are several reasons for trying a multi-objective approach. For example, as indicated in [6], there are works that demonstrate that some evolutionary algorithms improve their effectiveness when they are applied to multi-objective algorithms [7]. Indeed, in [1] it is argued that PSP problem can be naturally modelled as a multi-objective problem because the protein conformations could involve tradeoffs among different objectives as it is experimentally shown by analyzing the conflict between bonded and non-bonded energies. In this paper, we also apply a multi-objective approach to the PSP problem.

A multi-objective optimization problem [8] can be defined as the problem of finding a vector (1) that satisfies a given restriction set (2) and optimizes the function vector in (3). The objectives are usually in conflict between themselves, thus, optimizing one of them is carried out at the expense of the values of the others. This leads to the need of making a compromise, which implies the concept of Pareto optimality. In a multi-objective optimization problem, a decision vector x* is said to be a Pareto optimal solution if there is not any other feasible decision vector, x, that improves one objective without worsening at least one of the other objectives, given P the set of Pareto optimal solutions, (4). Usually, there are many vectors which are Pareto optimal. These solutions are called non-dominated. The set of all non-dominated solutions, in the decision space, determines the Pareto front in the objective space. Finally, a decision maker can select the solutions in its preferred front zone [6].

$$x = [x_1, x_2, ..., x_n] \tag{1}$$

$$g(x) \leq 0, h(x) = 0 \tag{2}$$

$$f(x) = \{f_1(x), f_2(x), , f_m(x)\} \tag{3}$$

$$\forall a, b \in P(\exists i, j \in \{1, 2, \cdots, n\} | (f_i(a) < f_i(b)) \wedge (f_j(a) > f_j(b))) \tag{4}$$

3 The Proposed Multi-objective Approach

This section describes the components required to solve the PSP problem by a multi-objective optimization using the *ab initio* approach. These components are: the cost function, initialization, the mutation operators and the set of variables.

Cost function: Although a realistic measure of protein conformation quality should probably imply considering quantum mechanics principles, it would be too computationally complex to become useful. Thus, as it is usual, we have used the Chemistry at HARvard Macromolecular Mechanics (CHARMM) and AMBER99 energy functions [1,9]. These are the most popular all-atom force field used for studying macromolecules and proteins respectively. We have considered its implementation at the TINKER library package [10]. As we use a multi-objective evolutionary optimization formulation of the PSP problem, the different terms of the energy function have to be transformed into several objectives. In [1] it is distinguished between bond and non-bond energies. We use this idea, although we have introduced some modifications according to the characteristics of the solution domain. Analyzing these energies we can observe that the Van Der Wall energy term has higher change range than others. Accordingly, this last energy terms can be hidden by the Van Der Wall energy term. Thus to optimize this energy appropriately, we propose a cost function with three objectives as follows: the bond energy and two more objective for the non-bond energies, one for Van Der Wall and other for the rest of non-bond terms. Also, we propose to take in account the difference between the probabilistic 3D protein structure (see next paragraph) and the current protein structure. To do that, we use another cost function configuration: bond energy, non-bond energy and difference with the probabilistic protein 3D structure.

Initialization: We have developed two initialization methods: random and probabilistic. The random method sets each variable with a random value according to the constraints of the variable. The probabilistic method uses the rotamer libraries to set each amino-acid at its most probable conformation. In this work we do not use the probabilistic initialization in the NSGA2 algorithm, because a population algorithm requires a set of different individuals.

Mutation operators: In [1] there are two different mutation operators. The first one is used to explore the search space, and the second one performs a local search optimization. We propose a third mutation operator, this method generate a mutation more conservative than the mutation to explore, and bigger than the mutation to perform a local search. This operator is executed with the same probability as the first one, and it mutates the side-chain with a gaussian function defined in the rotamer library. Doing that, we can avoid some local minima.

Set of variables: Any cost function depends on a set of variables that define the search space. These variables provide the required information to build the 3D conformation of the protein. In this paper we use torsion angles to represent the conformation of the protein, because this representation needs less variables than other alternatives. Three torsion angles are required in the backbone per each amino acid and some additional torsion angles depending on the side-chain [1] (Figure 2).

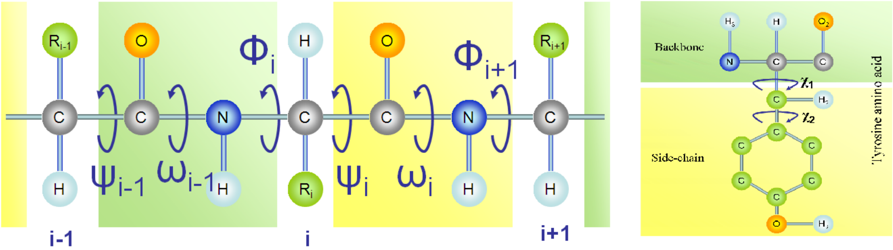


Fig. 2. (left) Backbone angles ϕ, ψ and ω. (right) Side-chain angles χ_i in the tyrosine amino acid.

In many optimization problems, once a suitable cost function is found, it would not be difficult to obtain an efficient evolutionary algorithm that generates sufficient good solutions for the problem. This is not the case in the PSP problem. As the search space is so vast, it is necessary to select an adequate set of variables and a representation space, along with the inclusion of specific information about the PSP problem in the evolutionary algorithm. Some alternatives can be taken into account to reduce the search space such as secondary and super-secondary structure prediction and inclusion of rotamers libraries [11]. Rotamers libraries have statistical information about the conformation of the amino acids in known molecules.

Thus, although the PSP problem implies to predict the tertiary structure of a given protein from its primary structure, it could be a good idea to use predictions of the secondary and super-secondary structures as they give us information about the amino acids involved in one of these structures, determining some constraints in the torsion angles of each amino acid (as shown in Table 1). In order to get the super-secondary structure given its secondary structure, we have to analyze the conformation of the residues in the short connecting peptide between two secondary structures. They are classified into five types, namely, a, b, e, l or t [12]. Sun *et al.* [12] developed a method to predict the eleven most frequently occurring super-secondary structures: H-b-H, H-t-H, H-bb-H, H-ll-E, E-aa-E, H-lbb-H, H-lba-E, E-aal-E, E-aaal-E and H-l-E where H and E are α helix and β strand, respectively. In this way a reduction in the search space of the PSP problem is obtained.

Moreover, side-chain torsion angles have interesting dependencies. Dumbrack et al. [11] give many rotamers libraries that help us to identify constraints about these torsion angles. An example of these libraries is the backbone-independent rotamer library. Given an amino acid, this library includes constraints for its side-chain torsion angles. There are dependencies between side-chain torsion angles in the same amino acid, but these dependencies are not related with the backbone torsion angles. It is difficult to find a good representation for these constraints because their mutual dependency. The backbone-dependent rotamer library is more complex than the backbone-independent rotamer library, because the former includes the dependency between side-chain torsion angles and backbone torsion angles. Therefore, the size of the library increases significantly and,

Table 1. Search space of each angle ϕ and ψ depending on the position of the super-secondary structure they are

Super-secondary structure	ϕ	ψ
H (α helix)	[-75, -55]	[-50, -30]
E (β strand)	[-130, -110]	[110, 130]
a	[-150, -30]	[-100, 50]
b	[-230, -30]	[100, 200]
e	[30, 130]	[130, 260]
l	[30, 150]	[-60, 90]
t	[-160, -50]	[50, 100]
undefined	[-180, 0]	[-180, 180]

although the information provided by this library is interesting, the time required to include this information in the optimization procedure also increases. Therefore, there are dependencies among backbone torsion angles and side-chain torsion angles, and there are also dependencies inside torsion angles of the side-chain. An optimization process tries an independent movement for each variable. Thus, if we include the backbone torsion angles and the side-chain torsion angles in the set of variables, we will ignore the cross-information between torsion angles. Therefore, a suitable way to approach the problem could be to implement a way to manage the backbone torsion angles in an optimization process and, inside it, to include another mechanism to manage the side-chain torsion angles.

Thus, In this work we propose a new method to manage torsion angles using the backbone-dependent rotamer library. In this way, we reduce the set of variables involved in the optimization process by eliminating the side-chain torsion angles. Nevertheless we can not eliminate side-chain torsion angles without adding another mechanism to take them into account. This mechanism selects the most probable conformation of the side-chain in each amino acid depending on the backbone torsion angles. Then it is possible change the side-chain conformation, considering the alternatives included in the library, to produce a better global conformation.

4 Results

In this section we provide and analyze the results obtained with our procedure. They have been obtained by using the 1CRN protein [1, 13] as a benchmark. As the cost functions used to model the PSP problem only approximate the conformation energy of the protein, it is not enough to measure the quality of the 3D structure obtained. To do so, we should accomplish a comparison between a given known protein structure and the solution obtained by optimizing the cost function that model the corresponding PSP problem. The most famous measure of similarity between predicted and known native structures is the RMSD [1] ([5]). RMSD computes the 3D difference between two structures, therefore the lower the molecule RMSD, the better is the 3D structure obtained.

$$RMSD(a,b) = \sqrt{\frac{\sum_{i=1}^{n} |\tau_{ai} - \tau_{bi}|^2}{n}} \tag{5}$$

We have executed an I-PAES and a NSGA2 with our heuristics and methods. We also have executed the original implementation of the algorithm described in [1]. We have executed each one for 21 times along 250.000 cost function evaluations and we have selected the best solution of the Pareto front of each execution. This information give us the best behavior that one approach has. We have computed the mean of these RMSDs obtaining the results of Table 2. The results shown in Table 2 do not allow us to get very clear conclusions about the best alternative for the nature of the third objective, the multi-objective evolutionary algorithm applied, and initialization procedure, or the energy functions (CHARMM or AMBER99). If we consider the obtained mean values of RMSD bellow 8.00 Å, it can be concluded that AMBER99 presents a better behavior and that the use of a third objective is useful. Nevertheless, although the best result is provided when NSGA2 is executed, I-PAES provides values below 8.00 Å in two alternatives. Similar things happen with the initialization and the third objective. This way, although the alternatives proposed in this paper allow an improvement of previous procedures in the multi-objective approach to PSP [1], it is clear that more experiments are required considering a wider set of proteins. This is the goal of or future work along with the parallel implementation of our procedures in order to reduce the times required to get the results. In [14], we provided our first results in this trend. In this work we have used 14 processors to execute parallel algorithms.

Table 2. Provided algorithms versus other approach for 1CRN protein

Algorithm	# Obj.	Init.	3rd obj.	mean	dev.	parallel	time
NSGA2-Reduced-amber99	2	Rand.	-	8.56 Å	0.99 Å	Yes	1.5 h
NSGA2-Reduced-amber99	3	Rand.	Van Der	*7.71* Å	*0.54* Å	Yes	1.5 h
NSGA2-Reduced-amber99	3	Rand.	Prob.	8.28 Å	0.96 Å	Yes	1.5 h
I-PAES-Reduced-amber99	2	Rand.	-	11.40 Å	1.78 Å	Yes	2.5 h
I-PAES-Reduced-amber99	2	Prob.	-	10.35 Å	1.50 Å	Yes	2.5 h
I-PAES-Reduced-amber99	3	Rand.	Prob.	9.72 Å	1.23 Å	Yes	2.5 h
I-PAES-Reduced-amber99	3	Prob.	Prob.	7.86 Å	1.03 Å	Yes	2.5 h
I-PAES-Reduced-amber99	3	Rand.	Van Der	8.49 Å	1.19 Å	Yes	2.5 h
I-PAES-Reduced-amber99	3	Prob.	Van Der	*7.78* Å	0.94 Å	Yes	2.5 h
I-PAES CHARMM27 [1]	2	Rand.	-	8.35 Å	1.35 Å	No	20.0 h

5 Conclusion

The PSP problem joins biological and computational concepts. It requires accurate and tractable models of the conformations energy. Thus, there is a long way to go to find useful solutions to the problem for proteins of realistic sizes. Our contribution in this paper deals with a new procedure for PSP based on a multi-objetive evolutionary algorithm. It allows a reduction in the number of variables and few heuristics to improve the quality of the solutions such as the three objectives cost function. By comparing the results of our procedures with those provided by other multi-objective approach [1], it can be shown that our methods provide conformations of comparable or even better quality. Nevertheless, more experiments are required by using more (and more complex) proteins to get more conclusive results.

Acknowledgment

This paper has been supported by the Spanish Ministerio de Educacion y Ciencia under project TIN2007-60587.

References

1. Cutello, V., Narcisi, G., Nicosia, G.: A multi-objetive evolutionary approach to the protein structure prediction problem. J. R. Soc. Interface, 139–151 (2006)
2. Branden, C., Tooze, J.: Introduction to protein structure, ISBN 0-81-532305-0
3. Handl, J., Kell, D., Knowles, J.: Multiobjective optimization in bioinformatics and computational biology. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 4(2), 279–292 (2007)
4. Krasnogor, N., Hart, W., Smith, J., Pelta, D.: Protein structure prediction with evolutionary algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference (1999)
5. Cotta, C.: Hybrid evolutionary algorithms for protein structure prediction under the hpnx model. Advances in Soft Computing 2, 525–534 (2005)
6. Day, R., Zydallis, J., Lamont, G.: Solving the protein structure prediction problem through a multiobjective genetic algorithm. Nanotech, 32–35 (2002)
7. Zydalis, J., Veldhuizen, A.V., Lamont, G.: A statistica comparison of moeas including the momga-ii. In: Proc. 1st Int. Conference on Evolutionary Multicriterion Optimization, pp. 226–240 (2001)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
9. Wathen, B.: Hydrophobic residue patterning in β-strands and implication for β-sheet nucleation, `http://qcse.queensu.ca/conferences/documents/BrentWathen.ppt`
10. INKER, Software tools for molecular design, `http://dasher.wustl.edu/tinker/`
11. Dunbrack, R.L., Cohen, F.E.: Bayesian statistical analysis of protein sidechain rotamer preferences. Protein Sci. 6, 1661–1681 (1997)
12. Sun, Z., Jiang, B.: Patterns and conformations of commonly occurring supersecondary structures (basic motifs) in protein data bank. Journal of Protein Chemistry 15(7) (1996)
13. RCSB, Protein data bank (pdb), `http://www.pdb.org`
14. Calvo, J., Ortega, J.: Parallel protein structure prediction by multiobjective optimization. In: Euromicro, Parallel, Distributed and Network-based Processing (2009)

Using Efficient RBF Networks to Classify Transport Proteins Based on PSSM Profiles and Biochemical Properties

Yu-Yen Ou and Shu-An Chen

Department of Computer Science and Engineering,
Yuan Ze University, Chung-Li, Taiwan

Abstract. Transport proteins are difficult to understand by biological experiments due to the difficulty in obtaining crystals suitable for X-ray diffraction. Therefore, the use of computational techniques is a powerful approach to annotate the function of proteins.

In this work, we propose a method based on PSSM profiles and other biochemical properties for classifying three major classes of transport proteins. Our method shows a 5-fold cross validation accuracy of 75.4% in a set of 1146 transport proteins with less than 20% mutual sequence identity.

1 Introduction

Transport proteins are difficult to understand by biological experiments due to the difficulty in obtaining crystals suitable for X-ray diffraction. Therefore, the use of computational techniques is a powerful approach to annotate the function of proteins. By using machine learning techniques, we can identify candidate transporters, predict their structure and localization in the membrane, and perform detailed functional annotation.

In recent years, several methods have been proposed for discriminating transport proteins based on amino acid conformational parameters, physical chemical properties and machine learning techniques [1,2,3,4]. However, the prediction performance and biological understanding still need further research.

In this work, we propose a method based on Position Specific Scoring Matrix (PSSM) profiles and other biochemical properties for classifying three major classes of transport proteins. Our method shows a 5-fold cross validation accuracy of 75.4% in a set of 1146 transport proteins with less than 20% mutual sequence identity.

2 Materials and Methods

2.1 Datasets

Our dataset is from major classes of TCDB (Transporter Classification Database) [5]. As Table 1 shows, the original dataset contains 1022 Channels/Pores proteins, 1363 Electrochemical Potential-driven Transporter proteins,

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 869–876, 2009.
© Springer-Verlag Berlin Heidelberg 2009

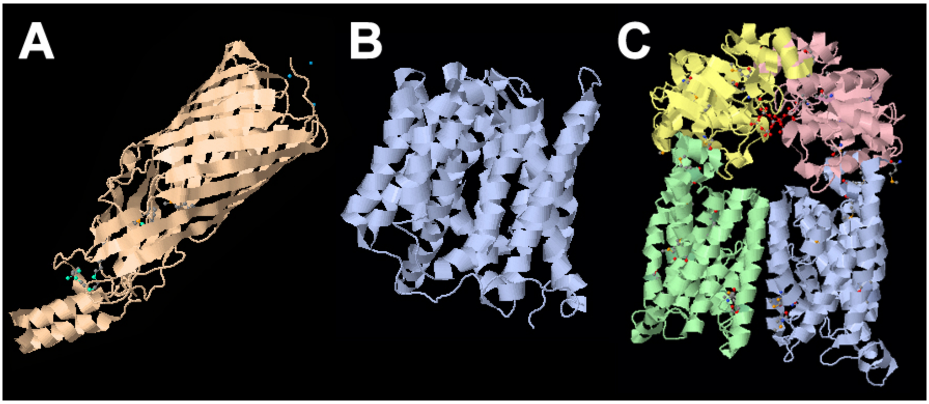


Fig. 1. Three Examples of Transport Proteins. A: Channel/Pores (1T16 from E. coli) B: Electrochemical Potential-driven Transporters (2GFP from E. coli) C: Primary Active Transporters (1L7V from E. coli).

and 1893 Primary Active Transporter proteins. We removed the sequences which are more than 20% identity with others by BLAST [6]. In addition, we also checked the e-value of each pair sequences with the criteria e-value $> e^{-50}$. Finally, the number of transport proteins in each class is 376, 208 and 562, respectively. Fig. 1 shows three sample structures from three classes of transport proteins.

2.2 Design of the Radial Basis Function Networks

Classification based on radial basis function (RBF) networks has several applications in bioinformatics. It has been widely used to predict the cleavage sites in proteins [7], inter-residue contacts [8], protein disorder [9], discrimination of β-barrel proteins [10] and so on.

In this paper, we have employed QuickRBF package [11] for constructing RBF network classifiers. Also, we adopted all default setting of QuickRBF package, and everyone can download the package from the website to verify the results.

In QuickRBF package, we do not need to select any parameters, but we have to select the centers from training data. In this work, we used all training data as hidden neurons for getting the best results.

Table 1. The Statistics of data set of this paper

Class	original data	reduced data
Channels/Pores	1022	376
Electrochemical P.-D. Transporters	1363	208
Primary Active Transporters	1893	562
total	4278	1146

The general mathematical form of the output nodes in an RBF network is as follows:

$$g_j(\mathbf{x}) = \sum_{i=1}^{k} w_{ji}\phi(||\mathbf{x} - \mu_i||\,; \sigma_i), \tag{1}$$

where $g_j(\mathbf{x})$ is the function corresponding to the *j-th* output node and is a linear combination of k radial basis functions $\phi()$ with center μ_i and bandwidth σ_i; Also, w_{ji} is the weight associated with the link between the *j-th* output node and the *i-th* hidden node. In QuickRBF package, the default value of σ_i is 1, and w_{ji} will be calculated by program and input data.

Table 2. The Normalized Value of Five Biochemical Properties in 20 Amino Acids

Property	Ala(A)	Arg(R)	Asn(N)	Asp(D)	Cys(C)	Gln(Q)	Glu(E)	Gly(G)	His(H)	Ile(I)	Leu(L)	Lys(K)	Met(M)	Phe(F)	Pro(P)	Ser(S)	Thr(T)	Trp(W)	Tyr(Y)	Val(V)
pHi	0.4	1	0.33	0	0.29	0.36	0.31	0.4	0.6	0.41	0.4	0.87	0.37	0.34	0.44	0.36	0.36	0.39	0.36	0.4
Br	0.64	0.06	0.21	0.19	1	0.13	0.09	0.64	0.51	0.98	0.87	0	0.72	0.89	0.26	0.36	0.36	0.68	0.42	0.83
aM	0.55	0.71	0.42	0.25	0.69	0.73	0.57	0.18	1	0.35	0.47	0.74	0.66	0.51	0	0.39	0.21	0.21	0.23	0.54
dG	0.24	0	0.14	0.22	0.62	0.17	0.2	0.28	0.44	0.22	0.22	0.14	0.43	1	0.23	0.11	0.16	0.87	0.56	0.23
TdS	0.49	1	0.77	0.53	0.06	0.74	0.54	0.6	0.53	0.53	0.53	0.69	0.28	0	0.55	0.53	0.52	0.24	0.22	0.51

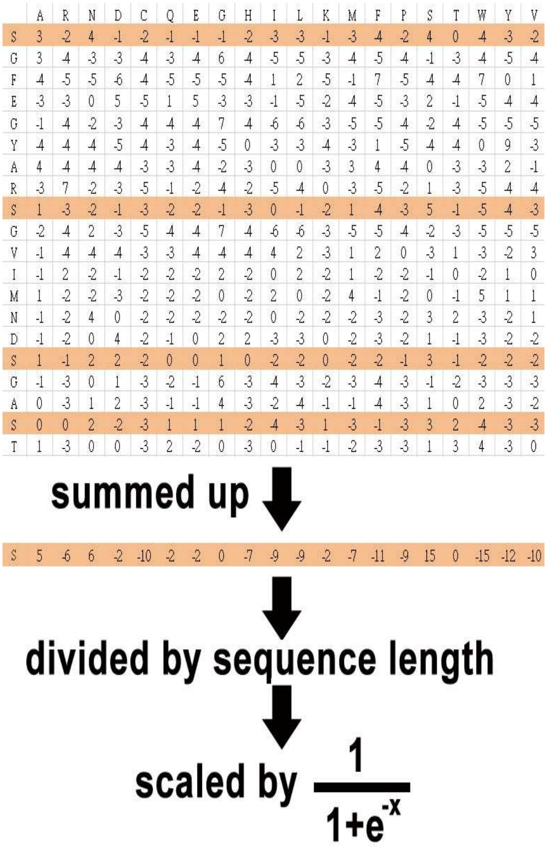
2.3 Compositions of Amino Acids and Amino Acid Pairs

If we have n proteins in the training data, we can use n vectors $\{x_i, i = 1, ..., n\}$, to represent the whole training data. Each vector has a label to show the groups those proteins belong to (e.g. electrochemical potential-driven transporters or primary active transporters).

The vector x_i has 20 dimensions for the amino acid composition and 400 dimensions for the amino acid pair composition. The 20 dimensions are the number of occurrences of 20 amino acids, and the 400 dimensions are the number of occurrences of 400 different amino acid pairs. In this paper, we also combine amino acid composition and amino acid pair composition, and then have 420 dimensions in each vector.

2.4 PSSM Profiles

In the structural point of view, several amino acid residues can be mutated without altering the structure of a protein and it is possible that two proteins have similar structures with different amino acid compositions. Hence, we have adopted the Position Specific Scoring Matrix (PSSM) profiles, which have been widely used in protein secondary structure prediction, subcellular localization and other bioinformatics problems with significant improvement [10,12,13]. The



	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
S	3	-2	4	-1	-2	-1	-1	-1	-2	-3	-3	-1	-3	-4	-2	4	0	-4	-3	-2
G	3	-4	-3	-3	-4	-3	-4	6	-4	-5	-5	-3	-4	-5	-4	-1	-3	-4	-5	-4
F	-4	-5	-5	-6	-4	-5	-5	-5	-4	1	2	-5	-1	7	-5	-4	-4	7	0	1
E	-3	-3	0	5	-5	1	5	-3	-3	-1	-5	-2	-4	-5	-3	2	-1	-5	-4	-4
G	-1	-4	-2	-3	-4	-4	-4	7	-4	-6	-6	-3	-5	-5	-4	-2	-4	-5	-5	-5
Y	-4	-4	-4	-5	-4	-3	-4	-5	0	-3	-3	-4	-3	1	-5	-4	-4	0	9	-3
A	4	-4	-4	-4	-3	-3	-4	-2	-3	0	0	-3	3	4	-4	0	-3	-3	2	-1
R	-3	7	-2	-3	-5	-1	-2	-4	-2	-5	-4	0	-3	-5	-2	1	-3	-5	-4	-4
S	1	-3	-2	-1	-3	-2	-2	-1	-3	0	-1	-2	1	-4	-3	5	-1	-5	-4	-3
G	-2	-4	2	-3	-5	-4	-4	7	-4	-6	-6	-3	-5	-5	-4	-2	-3	-5	-5	-5
V	-1	-4	-4	-4	-3	-3	-4	-4	-4	4	2	-3	1	2	0	-3	1	-3	-2	3
I	-1	2	-2	-1	-2	-2	-2	2	-2	0	2	-2	1	-2	-2	-1	0	-2	1	0
M	1	-2	-2	-3	-2	-2	-2	0	-2	2	0	-2	4	-1	-2	0	-1	5	1	1
N	-1	-2	4	0	-2	-2	-2	-2	-2	0	-2	-2	-2	-3	-2	3	2	-3	-2	1
D	-1	-2	0	4	-2	-1	0	2	2	-3	-3	0	-2	-3	-2	1	-1	-3	-2	-2
S	1	-1	2	2	-2	0	0	1	0	-2	-2	0	-2	-2	-1	3	-1	-2	-2	-2
G	-1	-3	0	1	-3	-2	-1	6	-3	-4	-3	-2	-3	-4	-3	-1	-2	-3	-3	-3
A	0	-3	1	2	-3	-1	-1	4	-3	-2	-4	-1	-1	-4	-3	1	0	2	-3	-2
S	0	0	2	-2	-3	1	1	1	-2	-4	-3	1	-3	-1	-3	3	2	-4	-3	-3
T	1	-3	0	0	-3	2	-2	0	-3	0	-1	-1	-2	-3	-3	1	3	4	-3	0

S	5	-6	6	-2	-10	-2	-2	0	-7	-9	-9	-2	-7	-11	-9	15	0	-15	-12	-10

Fig. 2. The Method to Sum the PSSM Features from Original PSSM profiles

PSSM profiles have been obtained by using PSI-BLAST and non-redundant (NR) protein database.

In the classification of transport proteins, we use PSSM profiles to generate 400D input vector as input features by summing up the same amino acid rows in PSSM profiles. Fig. 2 shows the detail of how to generate the 400 PSSM features from original PSSM profiles. Every element of 400D input vector was divided by the length of the sequence and then be scaled by $\frac{1}{1+e^{-x}}$, where x means the original summing up value of PSSM profiles.

2.5 Biochemical Properties

We have tried five additional biochemical properties in this paper, which are Isoelectric point(PHi), Buriedness(Br), Power to be at the middle of alpha helix (aM), Unfolding Gibbs free energy change (DG), and Unfolding entropy change (TdS). These five biochemical properties are picked up from 48 biochemical

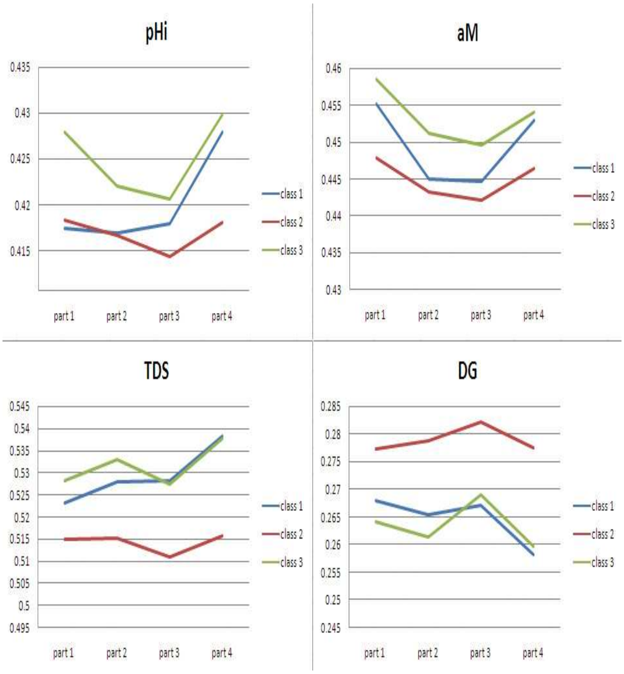


Fig. 3. The Distribution of 4 Biochemical Properties in 4 parts of protein

properties in [14]. These 48 properties are analyzed from many different physicochemical and biochemical properties to represent the characteristics of amino acids.

In this paper, we divided the amino acid sequences of each proteins into four equal parts. That is, if the length of protein is 100 amino acids, the first 25 amino acids belong to part1, 26-50 are part2, 51-75 are part3, and the others are part4. Then, the average value of different biochemical properties from four parts will be determined. The four average biochemical values from four parts are the biochemical features of the protein.

We added these 48 biochemical properties one by one into the PSSM feature sets, and kept the property in the feature set if the performance got improvement. We repeat the process until the performance can not have any improvement, and finally have added these five properties into the feature sets.

Table 2 lists the normalized values of these five properties, and Fig. 3 shows the distribution of four sample properties.

2.6 Assessment of Predictive Ability

The prediction performance was examined by 5-fold cross validation test, in which the three types of proteins were randomly divided into five subsets of approximately equal size. We have trained the data with four subsets and the remaining set was used to test the performance of the method. This process was repeated five times so that every subset was once used as the test data.

We used recall, precision, f-measure, Accuracy, MCC (Matthew's correlation coefficient) to measure the prediction performance. TP, FP, TN, FN are true positives, false positives, true negatives, and false negatives, respectively.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{F-Measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{4}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{5}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FN})(\text{TP} + \text{FP})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{6}$$

3 Results

The experiments in this section are conducted to evaluate the performance of the RBF networks classifier with different feature sets. Also, we compare the results with other classifiers, Support Vector Machine (SVM), K Nearest Neighbor (KNN), and Neural Networks (NN) from Mr. Gromiha's work in 2008 [1].

In Table 3, we listed the different results with RBF networks classifier and different feature sets, "AAC", "DPC", "AA+DP", "PSSM", and "PSSM+5".

Table 3. Comparison of Different Feature Sets with RBF network Classifier and Previous Work

	Recall			Precision			F-Measure				
Method	T1	T2	T3	T1	T2	T3	T1	T2	T3	Accuracy	MCC
KNN*	52.5	70.7	57.2	58.6	58.8	61.5	55.4	64.2	59.3	59.8	
SVM*	46.9	75.7	64.2	67.5	62.0	60.3	55.3	68.2	62.2	62.4	
NN*	54.9	71.7	64.2	63.6	65.9	61.9	58.9	68.7	63.0	63.6	
AAC	58.8	58.7	78.8	67.8	65.6	69.9	63.0	62.0	74.1	68.6	0.53
DPC	60.4	65.4	76.2	67.2	69.4	69.9	63.6	67.3	72.9	69.0	0.54
AA+DP	60.4	63.0	79.2	70.5	68.6	70.3	65.1	65.7	74.5	70.1	0.55
PSSM	67.0	63.5	81.1	73.9	69.5	74.2	70.3	66.4	77.5	73.3	0.60
PSSM + 5	71.0	64.4	82.4	75.9	73.2	75.8	73.4	68.5	79.0	75.4	0.63

* These three results are from Mr. Gromiha's work [1].

T1: Channel/Pores T2: Electrochemical Potential-driven Transporters T3: Primary Active Transporters.

The "AAC" row listed the results of using amino acid composition as feature set. The "DPC" row is the result of using amino acid pair composition as features, and the "AA+DP" row listed the results of combining amino acid composition and amino acid pair composition as feature set. Then, the "PSSM" row listed the results by using PSSM profiles as features as describe in the previous section. Also, the following row, "PSSM+5", is the results by adding five biochemical properties as features. We can see the best results are from PSSM profile with five additional biochemical features.

As Table 3 shows, the results of RBF networks classifier with PSSM profiles and five biochemical properties as feature sets perform significantly better than other results. The improvement of accuracy is more than 11%, from 63.6% to 75.4%.

4 Conclusion

We have developed a novel approach based on PSSM profiles and other biochemical properties for classifying three major classes of transport proteins, Channels/Pores, Electrochemical Potential-driven Transporters, and Primary Active Transporters.

Our proposed approach achieved a 5-fold cross validation accuracy of 75.4%, which improves the performance more than 11% by comparing with the previous work. Also, experimental results showed that PSSM profiles and additional biochemical properties can significantly improve the prediction accuracy. In addition, comparing with other neural network classifiers, like SVM and NN, the proposed method doesn't need to spend a lot of time to select the parameters, like the cost parameter and kernel parameter of SVM.

References

1. Gromiha, M.M., Yabuki, Y.: Functional discrimination of membrane proteins using machine learning techniques. BMC Bioinformatics 9, 135 (2008)
2. Lee, T., Paulsen, I., Karp, P.: Annotation-based inference of transporter function. Bioinformatics 24, i259 (2008)
3. Li, H., Dai, X., Zhao, X.: A nearest neighbor approach for automated transporter prediction and categorization from protein sequences. Bioinformatics 24, 1129 (2008)
4. Lin, H., Han, L., Cai, C., Ji, Z., Chen, Y.: Prediction of Transporter Family From Protein Sequence by Support Vector Machine Approach. PROTEINS: Structure, Function, and Bioinformatics 62, 218–231 (2006)
5. Saier Jr., M., Tran, C., Barabote, R.: TCDB: the Transporter Classification Database for membrane transport protein analyses and information. Nucleic Acids Research 34, 181 (2006)
6. Altschul, S., Madden, T., Schaffer, 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 25, 3389–3402 (1997)

7. Yang, Z., Thomson, R.: Bio-basis function neural network for prediction of protease cleavage sites in proteins. IEEE Transactions on Neural Networks 16, 263–274 (2005)
8. Zhang, G., Huang, D.: Prediction of inter-residue contacts map based on genetic algorithm optimized radial basis function neural network and binary input encoding scheme. Journal of Computer-Aided Molecular Design 18, 797–810 (2004)
9. Su, C.T., Chen, C.Y., Ou, Y.Y.: Protein disorder prediction by condensed PSSM considering propensity for order or disorder. BMC Bioinformatics 7, 319 (2006)
10. Ou, Y.Y., Gromiha, M., Chen, S.A., Suwa, M.: TMBETADISC-RBF: Discrimination of β-barrel membrane proteins using RBF networks and PSSM profiles. Computational Biology and Chemistry 32, 227–231 (2008)
11. Ou, Y.Y.: QuickRBF: a library for radial basis function networks, `http://csie.org/~yien/quickrbf/`
12. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292, 195–202 (1999)
13. Xie, D., Li, A., Wang, M., Fan, Z., Feng, H.: LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. Nucleic Acids Research 33, W105–W110 (2005)
14. Tomii, K., Kanehisa, M.: Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins (1996)

Artificial Neural Network Based Algorithm for Biomolecular Interactions Modeling

Christophe Lemetre[1], Lee J. Lancashire[2], Robert C. Rees[1], and Graham R. Ball[1]

[1] The van Geest Cancer Research Center, Nottingham Trent University, School of Science and Technology, Clifton campus, Clifton lane, Nottingham NG11 8NS, United Kingdom
[2] Clinical and Experimental Pharmacology, Paterson Institute for Cancer Research, University of Manchester, Manchester M20 4BX, United Kingdom

Abstract. With the advent of new genomic platforms there is the potential for data mining of genomic profiles associated with specific subclasses of disease. Many groups have focused on the identification of genes associated with these subclasses. Fewer groups have taken this analysis a stage further to identify potential associations between biomolecules to determine hypothetical inferred biological interaction networks (*e.g.* gene regulatory networks) associated with a given condition (termed the interactome). Here we present an artificial neural network based approach using the back propagation algorithm to explore associations between genes in hypothetical inferred pathways, by iteratively predicting the level of expression of each gene with the others, with respect to the genes associated with metastatic risk in breast cancer based on the publicly available van't Veer data set [1]. We demonstrate that we can identify a subset of genes that is strongly associated with others within the metastatic system. Many of these interactions are strongly representative of likely biological interactions and the interacting genes are known to be associated with metastatic disease.

Keywords: artificial neural networks; breast cancer; metastasis; interactions; interactome.

1 Introduction

1.1 The Interactome Problem and Its Potential

Recently, with the advent of gene expression array platform technologies, a large number of groups have focused on the profiling of a range of diseases and conditions. While huge efforts have focused on the generation of data, fewer groups have addressed issues of appropriate analysis. The data generated by microarray platform technologies is non-linear and highly dimensional with significant redundancy. This necessitates analysis strategies that appropriately identify components of functional relevance. To date, many groups have analysed microarray data for the identification of gene signatures that associate with specific clinical questions. For example van't Veer et al. [1] and West et al. [2] focused on genes associated with metastases in breast cancer. However, not all of the potential information contained within these datasets has been deeply investigated and by limiting our efforts to single biomarkers

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 877–885, 2009.
© Springer-Verlag Berlin Heidelberg 2009

we are attempting to model a real-world system that is dynamic, highly complex and correlated.

More recently groups have considered the use of microarray data for the investigation of gene regulations in terms of a more systems level view of the processes associated with disease. Barabási & Oltva stated that *"it is increasingly clear that a discrete biological function can only rarely be attributed to an individual molecule."* [3], but on a highly and complex interaction of biomolecules. While the identification of gene signatures remains important, investigating an extension of this in describing how these gene sets change in expression, and subsequently change the expression of other genes of functional relevance with a given clinical question has the potential to elucidate novel disease specific pathways that may be used to identify potential therapeutic targets.

1.2 Interaction Studies

Several techniques for the exploration of interaction networks in both proteins and genes have already been described. Examples include likelihood approaches [4] and Bayesian methods [5]. A number of groups focused their studies on the use of functionality to define interactions [6-8]. Schlitt & Brazma present a review of methodologies for gene regulatory networks [9]. However, the main disadvantage of many of these methodologies is that they provide only limited information regarding the nature of the interactions. In many cases only the presence of an interaction is considered. Furthermore, interactions are often only considered in the context of the target, and not within the whole gene marker pool.

To overcome current limitations we propose a novel approach based on back-propagation (BP) artificial neural networks (ANNs) that simultaneously considers all genes in a dataset. This will be introduced using genes associated with distant metastasis in breast cancer explored previously by the van't Veer study [1]. This is achieved by the development and comparison of a series of ANN models, each of which uses all available inputs (genes) to predict the expression of a single omitted input. This is repeated for all single inputs within the set. By examination of the weights and predictive performance of each model a matrix of interactions can be determined. The advantage of this approach over others is that the multi-factorial consideration of each input allows the magnitude of interaction to be determined for a given pair of parameters, whether it be inhibitory, stimulatory, bi or unidirectional. Once the initial screening is complete, non-significant interactions are removed using a decision threshold based upon the absolute values of the association between each input determined by the matrix of interactions. This study adopts an approach based on iterative prediction of each single input expression from all the others in a defined set. The results from each of these models are subsequently integrated into an interaction map based on the weights of each sub model. All sub models used to define the interaction map are extensively validated by Monte Carlo Cross Validation (MCCV). Previously, other groups have only investigated inferences by an analysis of large complicated ANN models associated with particular classes of expression [10-11]. This study advances upon these methods by allowing the direct prediction of associations between expressions of genes. The problems analysed are broken into small components that are unlikely to suffer from dimensionality issues associate with model complexity. Further, through a process of MCCV, early stopping and optimisation on a validation dataset, the findings are more likely to reflect a real-world solution and not an overfitting of the dataset of interest.

1.3 Artificial Neural Networks

ANNs are from the field of artificial intelligence and can "learn" from patterns by example [12-13], by adapting the connectivity between the nodes of the network. ANNs have been widely used in biology and Lisboa et al. [14] reviews their use in cancer applications. A major advantage of ANNs is their ability to cope with noisy and non-linear data, such as that found in microarray studies. Learning in ANNs occurs by adapting the weights of the connections between nodes of subsequent layers.

1.4 The Dataset

The dataset we used here consisted of the gene profiling with microarray from the seminal van't Veer breast cancer study [1]. This was, downloaded in excel format from http://www.rii.com/publications/2002/vantveer.html, and contained 78 samples, each with 24,481 inputs representing the expression ratio of each gene.

2 Materials and Methods

2.1 Data-Preprocessing

The dataset was utilized as presented by the van't Veer data set [2]. All the data were normalized to get a coherent dataset. The level of expression of every single gene was rescaled between 0 and 1 through all the cases.

2.2 Pre-screening of the Data to Determine Genes Associated with Metastatic Risk

The ANN algorithm described in [13] and [15] was successfully applied to the dataset to screen microarray data for genes of relevance to a given biological question. Here, we have utilised this approach to select the most important genes that can accurately predict the metastatic risk class described in the van't Veer study [1]. Gene microarray intensities from the arrays were used as single inputs to the ANN model. The output node represented two classes; 0 represented no distant metastases within 5 years, 1 represented cases who developed distant metastases within 5 years. The performance for each single gene model was determined based on prediction accuracy on a test dataset using cross validation. This produced a rank order of genes based on their predictive performance for the metastatic risk class. The top 100 genes defined in this analysis were further used in our interaction algorithm.

2.3 The Interaction Algorithm

The main idea of the approach lies on the iterative calculation of the influence that multiple variables may have on a single one. Indeed, if we have been able to use all the variables to explain one particular outcome, we can now use one of these variables as an outcome and all the remaining to explain the level of expression of the first one, and therefore be able to find out the influence everyone has on this particular gene expression level. Once the first input is complete, we repeat the procedure for all the other variables.

The interaction algorithm presented in this study is based on a three-layer feed forward Perceptron with a BP algorithm and a sigmoid transfer function [16]. The hidden and output layers consisted of 2 and 1 nodes respectively. Training was performed through 3000 epochs, terminated by either a window of 1000 epochs without improvement of the Mean Squared Error (MSE) on the validation subset, or an MSE threshold of 10^{-2}. Momentum and learning rate were set respectively at 0.5 and 0.1 as previously shown [13]. Training was repeated 50 times for each model, with MCCV applied prior to training to randomly split the patients in three different subsets; here the training set comprised 60% of the cases, and validation and test sets contained 20% each.

To define an interaction map for a given set of genes, the first input gene in the dataset was defined as an output. All others were set as inputs and applied to the ANN architecture and algorithm described above. The weights of the trained ANN model were stored. This process was iteratively repeated for all inputs in the dataset, treating each one in turn as an output. The weights relating a given input to a given output were then analyzed based on the sum of weights leading from an input to the output, to determine the intensity of the relationship between a source (input) and target (output). Analysis of the weights across all of the potential associations provided a rank order of their strength.

2.4 Visualisation of Interaction Maps

The concept of the interaction mapping uses the network theory as described previously [3], where a single marker is symbolised by a node and any relation between two markers is represented with a directed edge, and/or an arrow setting one of the node as a source, and the other one as a target. Cytoscape® [17] was used for interaction visualisation.

2.5 Filtering the Interaction Map to Determine the Key Interactions for the Metastatic Risk System

A matrix of interactions between variables was generated and extracted. Every association between any pair of variables contained in the dataset was investigated. Most interactions were found to be of non-significance. As this algorithm produces a huge amount of results (n inputs giving $n(n-1)$ associations), non-significant associations were filtered out so that only the highest absolute values remained. A filter value of 5 was used and the sign of the interaction was preserved so we could determine a direction for the mapped interaction indicating whether it was inhibitory or stimulatory.

3 Results

The approach was tested on the Van't Veer dataset [1]. In the first instance this consisted of screening the rank order of genes in terms of strength of predicting metastasis. The distribution of the genes ranked by their predictive error for test data was obtained (data not shown here), and clearly showed that the top genes have a higher ability to predict, and therefore are most relevant for the system being investigated by interaction analysis. The best predictor in the gene set was the mRNA sequence referred to as NM_001216 which codes for carbonic anhydrase IX (CA9), a protein involved in breast cancer prognosis and relapse [18].

To assess the interactions within the genes associated with prognostic outcome top 100 predictors were submitted to the interaction analysis. These genes had a range of predictive ability for test data between 72% for the first gene to 62.7% for the 100th. Analysis of the resulting weights from the interaction analysis produced a matrix of 9900 potential associations. It clearly appears that no relevant information can be elucidated from this interaction map due to its high dimensionality and complexity.

Figure 1 displays the distribution of the ranked interaction values. The majority of the interactions have low absolute values ranging from -10.9 to +17.2. The distribution of interaction indicates that there are very few strong links compared to the overall population of links. These warranted further investigation.

During the interaction analysis, the overall predictive performance of the models was assessed to define the interaction map by determining the correlation coefficient between actual and predicted levels. Therefore, the error values of the model were constantly assessed to produce a level of confidence for the interactions identified. In this dataset the average r^2 values was 0.665.

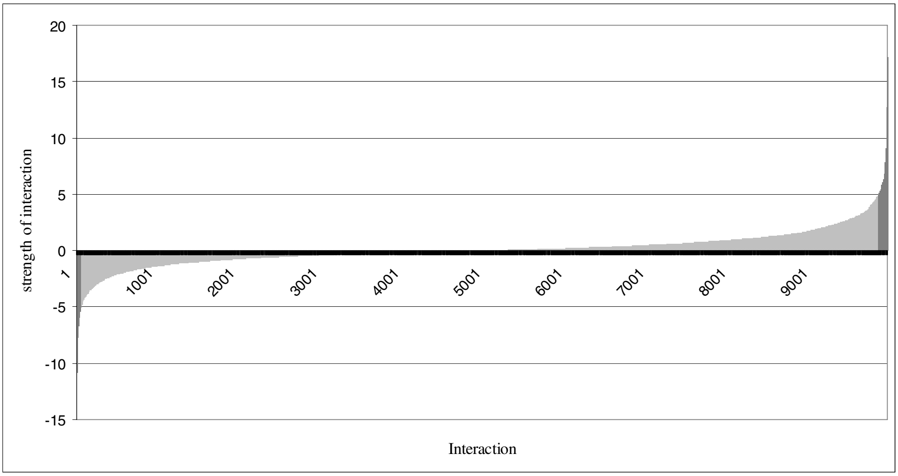


Fig. 1. Distribution of the interaction sorted by values. The interactions in dark grey at the extremities represent the strongest interactions at level 5.

A filter removing links between -5 and +5 was applied which resulted in the number of links decreasing from 10,000 to 163 (1.63% of the original interactome). This greatly simplified the interaction map, facilitating interpretation and understanding of the key features within the global interaction map as shown in Figure 2.

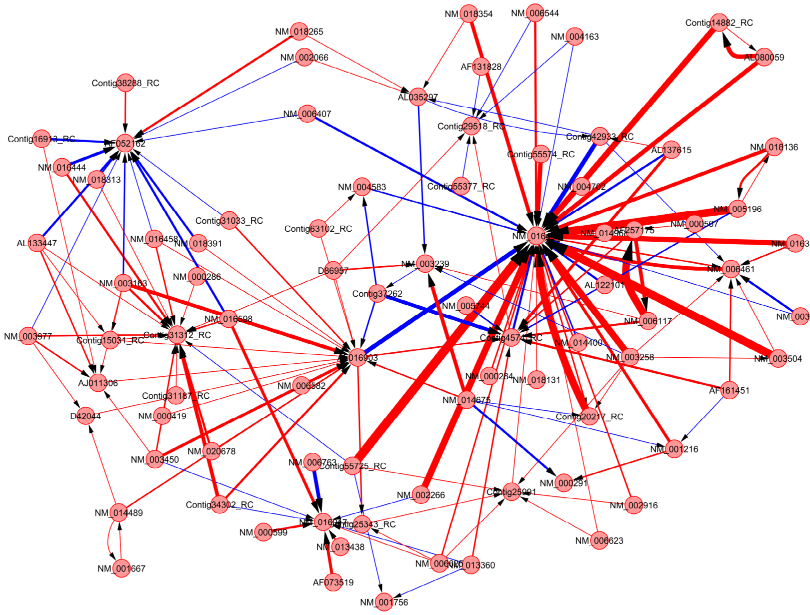


Fig. 2. Display of the interaction map reduced to level 5. Red is a positive interaction (stimulation). Blue is a negative interaction (inhibition).

From this map, we could identify the highest positive interaction involving NM_016448 (Retinoic acid-regulated nuclear matrix-associated protein) which is known to have a role in cancer cell proliferation [19], and Contig42933_RC, which corresponds to the genbank accession number R73468, referring to an unknown gene which has already been strongly associated with metastasis in breast cancer [20]. In the opposite negative direction, the strongest interaction was between NM_016448 and Contig55725_RC, the cell division cycle associated 7 (CDCA7). This has been shown to be over expressed in human tumours [21].

Moreover, we can clearly identify some key nodes involved in dense regions of interactions. Again, NM_016448 (Retinoic acid-regulated nuclear matrix-associated protein) appeared as a key feature within the whole interaction network, interacting strongly (within the strength of interactions over 5) with 35 other genes. Significantly, most of these interactions (33) have NM_016448 as the target.

According to these preliminary observations it appears that Retinoic acid-regulated nuclear matrix-associated protein has a key role in the whole interactome in this breast cancer study. This is in agreement with the literature which has demonstrated its implication in breast cancer [19] with the ability to decrease the expansion of cancer cells. Moreover, the mRNA sequence for CA9, previously identified in the screening step as the best predictor for metastasis in this case series, belongs to one of the features interacting with the Retinoic acid-regulated nuclear matrix-associated protein (NM_016448). In addition, we show an interaction between CA9 and the mRNA for phosphoglycerate kinase 1 (PGK1). Interestingly, interactions between both have already been described in relation to cancer [22].

4 Discussion

Within this study we have sought to develop an algorithm for the exploration of interactions within gene microarray data. This algorithm uses a multifactorial ANN based approach to model interactions.

4.1 Advances

This approach, incorporating an ANN based interaction mapping, offers the potential to identify key components and interactions in a multifactorial fashion. This may overcome limitations from previous studies using simple pairwise interactions, or those based upon biomolecular function, thus facilitating the identification of unidirectional influences. Moreover, the robustness of ANNs allows us to incorporate multiplatform data for exploration, and therefore not only focus on a particular system, but investigate on a wider scale for cross-platform interactions.

4.2 Validation

Earlier using the data supplied by West et al. [2], we have demonstrated that we can identify a rank order of key genes of biological relevance [15]. Similarly we have found a rank order of genes of biological relevance to metastasis here. CAIX, appeared as the strongest feature from the screening step, and was seen to have a strong interaction with PCK1, both already described to interact and to be involved in breast cancer [22]. Moreover, the retinoic acid-regulated nuclear matrix-associated protein, known to have a role in cancer cell proliferation [19], has been found to be strongly influenced within the whole interactome. We have shown other features with biological significance, confirming a level of confidence we can have for the model developed and the overall method.

However, despite these initial promising results, further validation both from the literature (and databases such as Ingenuity, or KEGG) and utilising gene silencing techniques needs to be conducted.

4.3 Limitations

The method is without its limitations. Firstly, the main limitation may come from the origin of the data itself. We are well aware that the collection of experimental data involves a certain level of bias. However, as we can only investigate and explore for relative interaction, none of the interaction is absolute, but relative to all the others within the dataset. This limitation may have an increased influence once the approach is applied to a cross-platform dataset.

Another important limitation comes from the complexity of the overall interactome obtained, and the relevance of the filtering strategy employed. It will be challenging to address a filtering strategy with an acceptable false discovery rate.

4.4 Future Work

The filtering strategy will have a key role in the future development of the approach. New approaches are under consideration, such as monitoring the deviation that multiple models can have and therefore filtering out any interactions that are not consistent over multiple runs of the process.

Further validation of the method is fundamental. This will involve databases queries and literature reviewing. Under consideration is an automatised process to query the databases and literature. In addition, comparison with other existing and validated methods will be key, in order to prove the efficiency of this approach. Several alternative methods are under considerations, such as Bayesian Networks [23].

An additional approach that could be further investigated is using the interaction mapping process to investigate the differences between the interactomes of two different classes, for example the metastatic *versus* the non-metastatic breast cancer patients. This may elucidate further features of interest associated with the breast cancer interactome.

5 Conclusion

A novel ANN based approach to predict the interactions which may exist between the components in a given dataset coming from high-throughput technologies has been described. The main advantages of the approach being its multifactorial character, and its reliance upon ANNs to allow for the inclusion of highly dimensional and non-linear data. Furthermore, the approach allows us to employ cross-platform datasets, and therefore we can apply the method to a wider scale. Preliminary results show this to be a novel and powerful tool for interactions discovery, which will allow for the investigation of biological systems and potentially provide a greater understanding of the underlying processes with a view to novel targets discovery.

Acknowledgments. We gratefully acknowledge the support of the John and Lucille van Geest foundation, and of the European Union 6th framework program (Contract nb. LSHC-CT-2004-503306).

References

1. van't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R., Friend, S.H.: Gene expression profiling predicts clinical outcome of breast cancer. Nature 415(6871), 530–536 (2002)
2. West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson Jr., J.A., Marks, J.R., Nevins, J.R.: Predicting the clinical status of human breast cancer by using gene expression profiles. PNAS 98(20), 11462–11467 (2001)
3. Barabási, A.L., Oltvai, Z.N.: Network biology: understanding the cell's; functional organization. Nat. Rev. Genet. 5(2), 101–113 (2004)
4. Liu, Y., Liu, N., Zhao, H.: Inferring protein–protein interactions through high-throughput interaction data from diverse organisms. Bioinformatics 21(15), 3279–3285 (2005)
5. Hartemink, A.J., Gifford, D.K., Jaakkola, T.S., Young, R.A.: Bayesian Methods for Elucidating Genetic Regulatory Networks. IEEE Intelligent Systems 17(2), 37–43 (2002)
6. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. PNAS 100(21), 12123–12128 (2003)

7. Shoemaker, B.A., Panchenko, A.R.: Deciphering Protein–Protein Interactions. Part II. Computational Methods to Predict Protein and Domain Interaction Partners. PLoS Comput. Biol. 3(4), e43 (2007)
8. Schwikowski, B., Uetz, P., Fields, S.: A network of protein–protein interactions in yeast. Nat. Biotechnol. 18(12), 1257–1261 (2000)
9. Schlitt, T., Brazma, A.: Current approaches to gene regulatory network modeling. BMC Bioinformatics 8, S9 (2007)
10. Hart, C.E., Mjolsness, E., Wold, B.J.: Connectivity in the Yeast Cell Cycle Transcription Network: Inferences from Neural Networks. PLoS Comput Biol 2(12), 1592–1607 (2006)
11. Xu, R., Wunsch, D.C., Frank, R.L.: Inference of Genetic Regulatory Networks with Recurrent Neural Network Models Using Particle Swarm Optimization. Computational Biology and Bioinformatics 4(4), 681–692 (2007)
12. Khan, J., Wei, J.S., Ringnér, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C., Meltzer, P.S.: Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat. Med. 7(6), 673–679 (2001)
13. Lancashire, L., Schmid, O., Shah, H., Ball, G.: Classification of bacterial species from proteomic data using combinatorial approaches incorporating artificial neural networks, cluster analysis and principal components analysis. Bioinformatics 21(10), 2191–2199 (2005)
14. Lisboa, P.J., Taktak, A.F.: The use of artificial neural networks in decision support in cancer: A systematic review. Neural Networks 19(4), 408–415 (2006)
15. Lancashire, L., Rees, R.C., Ball, G.R.: Identification of gene transcript signatures predictive for estrogen receptor and lymph node status using a stepwise forward selection artificial neural network modelling approach. Artif. Intell. Med. 43(2), 99–111 (2008)
16. Rumelhart, D.E., McClelland, J.L.: Parallel Distribution Processing: Explorations in the Microstructure of Cognition, Foundations, vol. 1. MIT Press, Cambridge (1986)
17. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 13(11), 2498–2504 (2003)
18. Span, P.N., Bussink, J., Manders, P., Beex, L.V., Sweep, C.G.: Carbonic anhydrase-9 expression levels and prognosis in human breast cancer: association with treatment outcome. British Journal of Cancer 89(2), 271–276 (2003)
19. Crowe, D.L., Kim, R., Chandraratna, R.A.S.: Retinoic Acid Differentially Regulates Cancer Cell Proliferation via Dose-Dependent Modulation of the Mitogen-Activated Protein Kinase Pathway. Molecular Cancer Research 1, 532–540 (2003)
20. Takahashi, H., Masuda, K., Ando, T., Kobayashi, T., and Honda, H.: Prognostic Predictor with Multiple Fuzzy Neural Models Using Expression Profiles from DNA Microarray for Metastases of Breast Cancer. Journal of Bioscience and Bioengineering, 98(3), 193–199, (2004)
21. Osthus, R.C., Karim, B., Prescott, J.E., Smith, B.D., McDevitt, M., Huso, D.L., Dang, C.V.: The Myc Target Gene JPO1/CDCA7 Is Frequently Overexpressed in Human Tumors and Has Limited Transforming Activity In vivo. Cancer Research 65, 5620–5627 (2005)
22. Winter, S.C., Buffa, F.M., Silva, P., Miller, C., Valentine, H.R., Turley, H., Shah, K.A., Cox, G.J., Corbridge, R.J., Homer, J.J., Musgrove, B., Slevin, N., Sloan, P., Price, P., West, C.M., Harris, A.L.: Relation of a hypoxia metagene derived from head and neck cancer to prognosis of multiple cancers. Cancer Research 67(7), 3441–3449 (2007)
23. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., Gerstein, M.: A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. Science 302(5644), 449–453 (2003)

Modelling Dengue Epidemics with Autoregressive Switching Markov Models (AR-HMM)*

Madalina Olteanu[1], Esther García-Garaluz[2], Miguel Atencia[3],
and Gonzalo Joya[2]

[1] SAMOS - CES, Université de Paris I (France)
[2] Departamento de Tecnología Electrónica
[3] Departamento de Matemática Aplicada
Universidad de Málaga (Spain)
Campus de Teatinos, 29071 Málaga, Spain
`madalina.olteanu@univ-paris1.fr`

Abstract. This work presents the Autorregresive switching-Markov Model (AR-HMM) as a technique that allows modelling time series which are controlled by some unobserved process and finite time lags. Our objetive is to bring to light the potential of this method to give valuable information about how an efficient control strategy can be performed. As a case of study, we apply the method to the dengue fever epidemics (DF) in 2001 in Havana. For this time serie, a first experiment with real data is performed in order to obtain the characterization of differents stages of the epidemics.

1 Introduction

Since the eighteen century, mathematical models of infectious diseases have been developed and used for providing information to health authorities [1]. During the 60s of the 20th century, the improvements in the prevention programs turned the research interests in the industrialized countries to other diseases. However, during the last few years dengue fever (DF) has become a major health problem affecting tropical and sub-tropical regions around the world, especially urban and suburban areas [2]. Thus, the question of modelling infectious diseases arose again. For health authorities, a well fitted mathematical model may help in figuring out new strategies for an efficient fight against the epidemics.

In this paper, a method based on hidden Markov models is described. This is a very suitable technique for vector-borne diseases since it enables characterization of unknown processes. The method presented herein allows to take into account finite-time feedback as well. This is an interesting feature since incubation and transmission rates may be considered.

---

* This work has been partially supported by the Agencia Española de Cooperación Internacional para el Desarrollo (AECID), project no. D/017218/08. The authors are grateful to Dr. Joseph Rynkiewicz for the REGRESS software.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 886–892, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The rest of the paper is organized as follows: in section 2, the autoregresive switching-Markov model (AR-HMM) is described. Section 3 presents the features of dengue fever epidemics. In section 4, AR-HMM is applied to real data of dengue fever epidemics. Finally, in section 5, conclusions and future work are proposed.

2 Autoregresive Switching-Markov Models

In some cases, time series may be explained not only by their past values, but also by some unknown, random process. Typical examples are economic cycles (recession and growth periods) or DNA identification (coding and non-coding regions). For these examples, the observed series will depend on the "state" or the "regime" of the unknown, hidden process. Usually, the hidden process is supposed to have a finite and fixed number of regimes, corresponding to the expected states of the phenomena to be modelled (for example, recession versus growth periods)

2.1 Notations and Definitions

Let us start by introducing hidden Markov chains (HMC) and recall the definition.

A bivariate process $(X_t, Y_t)_{t \in Z}$ is a hidden Markov chain if:

- (X_t) is a homogeneous Markov chain valued in a finite state space $E = \{e_1, ..., e_N\}$, $N \in N^\star$, with transition matrix $A = (a_{ij})_{i,j=1,...,N}$,

$$a_{ij} = P\left(X_{t+1} = e_j \mid X_t = e_i\right)$$

- (Y_t) valued in R^d, $d \geq 1$, conditionally independent w.r.t. (X_t) and $\mathcal{L}\left(Y_t \mid X_t = e_i\right) \equiv \mathcal{L}_i$.

The hidden Markov chain is described in Figure 1.

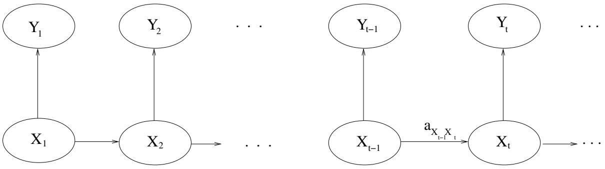


Fig. 1. Representation of Hidden Markov chains (HMC)

Hidden Markov models were introduced by [3] in 1966. They soon became essential tools in speech-recognition ([4], [6]), bioinformatics and analysis of biological sequences ([5]). Hidden Markov models have been already used for modelling epidemics data, with promising results ([8]).

Hidden Markov chains may be generalized by letting Y_t depend on its past values and on X_t. This new class of models may be defined as an autoregresive switching-Markov model (AR-HMM). AR-HMM were initially proposed in [7] and were extensively used in the sequel for modelling financial and economic time series. However, to our knowledge, there are no current applications for epidemics data.

An autoregressive switching-Markov process is a bivariate process $(X_t, Y_t)_{t\in Z}$ such that:

- (X_t) is a homogeneous Markov chain valued in $E = \{e_1, ..., e_N\}$, $N \in N^\star$, with transition matrix $A = (a_{ij})_{i,j=1,\ldots,N}$, $a_{ij} = P(X_{t+1} = e_j \mid X_t = e_i)$
- $Y_t = F_{X_t}(Y_{t-1}, ..., Y_{t-p}) + \sigma_{X_t}\varepsilon_t$
- F_{e_i} is a parametric function (linear or nonlinear) depending on some θ_i
- $\sigma_{e_i} \in \{\sigma_1, ..., \sigma_N\} \subset \left(R_+^\star\right)^N$ and (ε_t) is an iid sequence

We will suppose that the noise ϵ_t is a standard Gaussian. The parameter to be estimated contains the transition probabilities of the hidden Markov chain, the parameters in the regression functions and the standard deviations for the noise:

$$\Theta = (a_{ij}, \theta_i, \sigma_i, i, j = 1, ..., q)$$

The size of the parameter Θ is $q(q-1) + dq + q = q^2 + dq$.

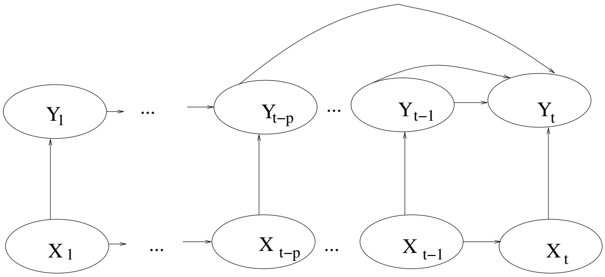


Fig. 2. Autoregresive switching-Hidden Markov Model (AR-HMM)

An autorregresive switching-hidden markov model is represented in 2, and its estimation procedure can be summarized in the following main steps:

1. First, the user has to set the number of regimes, q, the number of time lags, p (p may be chosen using some Akaike/Bayesian Information Criteria, AIC/BIC, or by cross-validation) and the type of regression functions (linear, Multi-Layer Perceptron, ...).
2. The parameter Θ is estimated using the Expectation-Maximization (EM) algorithm (see [9] for more details). A first characterization of the regimes (stability, influence of the past values,...) will be done using these results.

3. With the estimated values of the parameters, one may compute the posterior probabilities of belonging to each of the regimes, at any moment. These probabilities will allow the a posteriori identification and characterization of the regimes.

3 Dengue Epidemics

Dengue Fever (DF) and Dengue Haemorragic Fever (DHF) are infectious diseases transmitted to humans by the bites of Aedes mosquitoes. If a mosquito bites a Dengue Fever infected, it gets the disease and, after an incubation period, is capable of transmitting the infection to other people. That kind of disease, where an infectious organism that does not cause diseases itself transmits infection by contact among different hosts, is called a vector-borne disease [2].

The disease of Dengue fever shows a resurgent pattern worldwide in the last years, becoming one of the most important health problems in subtropical countries. WHO estimates [2] there may be 50 million dengue infections every year in the world, and two fifths of the world population are now at risk of Dengue. In the Americas more than 890 000 cases of Dengue fever were reported, of which 26 000 was of Haemorragic Fever.

There is not vaccine against dengue fever. Besides, people who suffer dengue are immune against that dengue strain. For these reasons, Dengue fever is frequently described as an epidemic SIR model for a short or medium-term model (under the assumption that only one strain of the virus exists). In this model the whole population is classified in three groups: susceptible, infected an recovered people. Susceptible people are those who are not immunized against the disease and do not present symptoms. Infected people are who present symptoms of the disease. Recovered are those who got the disease in the past and are immunized against it.

Several models have been proposed for dengue fever which depend on the vector population [10], [11]. This becomes an important limitation since generally it is not possible to have enough real information about it. On the other hand, dengue fever, as most biological processes, has a feedback which affects the system in a finite time way. This is the case of human and mosquito incubation periods or recovering rate.

Autoregresive switching-Markov models are specially suitable for dengue fever characterization, since they are able to model systems influenced by an unknown process and with a finite-time feedback.

4 A First Attempt to Model Dengue Epidemics Using HMM-AR

In order to perform the experiment, a 274-day register of new infections for Dengue Fever in Havana is available. From that database and the known recovering rate of the disease, the time series for the global number of infected

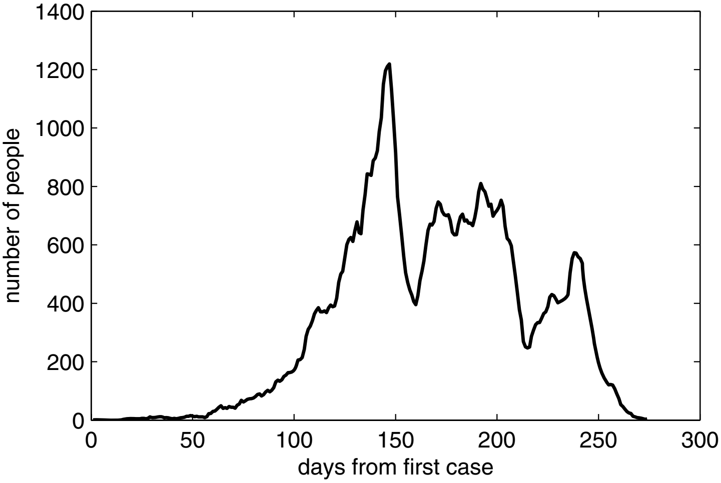


Fig. 3. Number of Dengue infected individuals in Havana (Cuba)

people was obtained (Figure 3). In it, the decrease periods are related with the fumigation programs performed by the health authorities.

An important consideration is the time lags in the experiment. Because of the incubation rate of human and mosquito lags between 8 and 15 days are considered.

For these epidemic data, we shall test three regimes: epidemics, non-epidemics and decrease periods. In all cases, each regime will be characterized a posteriori using these estimated probabilities.

In this case, the estimated transition matrix is :

$$P = \begin{pmatrix} 0.953 & 0 & 0.044 \\ 0 & 0.960 & 0.031 \\ 0.467 & 0.040 & 0.925 \end{pmatrix}$$

From this transition matrix, some aspects should be hhighlighted

- All three regimes are very stable.
- There's no transition between the first and the second regime.

Besides the transition matrix, a model for each regime was obtained with the form:

$$Y_t = aY_{t-8} - bY_{t-9} + ... + hY_{t-7} + i + \epsilon_t$$

Nevertheless, we are not showing the results for every coefficient for simplicity.

In the Figure 4 the probability of every regime is plotted.

From the whole results the obtained regimes can be characterized. The first and third regimes represent the non-epidemic and the epidemic periods respectively. Besides, it is necessary that the epidemics reach a high enough incidence to be removed. The decrease period is represented by the second regime.

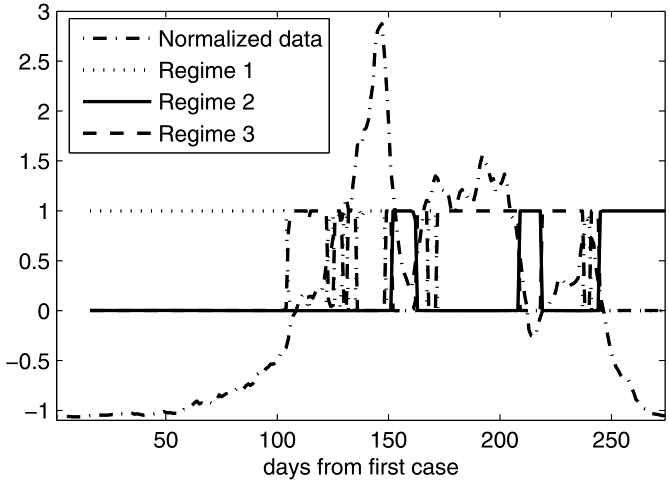


Fig. 4. Number of Dengue infected individuals in Havana (Cuba)

5 Conclusion and Future Works

In this work the Autoregresiveswitching-Markov model is described, emphasizing that it is specially intended for modelling systems with both finite feedback and the influence of a hidden process. As a first simple case of study we propose the aplication of AR-HMM to the identification of an epidemiological model of dengue fever. Thus, the analysis of an actual data-base of the dengue epidemics in Havana (Cuba) has been undertaken. The results permit the characterization of three different regimes: the epidemic, the no-epidemic and the decrease period. Some ideas can be inferred from the results, among them, that a high enough value of the epidemic prevalence is necessary before the non-epidemic state can be reached.

Some future works can be proposed. The next step is to test other configurations with different lags of time and different regression functions. The consideration of vertical transmission of dengue, that is, the transmission from the female to the eggs would be interesting as well. On the other hand, another future work line is extracting information from the available data with this technique in order to develop more efficient fumigation strategies. It would allow to perform prevention measures while the prevalence of the disease is still reduced.

References

1. Blower, S., Bernoulli, D.: An attempt at a new analysis of the mortality caused by smallpox and of the advantages of inoculation to prevent it: 1766. Reviews in Medical Virology 14, 275–288 (2004)
2. World Health Organization, `http://www.who.int`

3. Baum, L., Petrie, T.: Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Statist. 37, 1554–1563 (1966)
4. Rabiner, R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of IEEE 77(2), 257–286 (1989)
5. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological sequence analyses: Probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge (1998)
6. Bourlard, H.A., Morgan, N.: Connectionist speech recognition: A hybrid approach. Kluwer Academic Publishers, Dordrecht (1994)
7. Hamilton, J.D.: A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. Econometrica 57(2), 357–384 (1988)
8. Rath, T.M., Carreras, M., Sebastiani, P.: Automated Detection of Influenza Epidemics with Hidden Markov Models. In: R. Berthold, M., Lenz, H.-J., Bradley, E., Kruse, R., Borgelt, C. (eds.) IDA 2003. LNCS, vol. 2810, pp. 521–532. Springer, Heidelberg (2003)
9. Rynkiewicz, J.: Estimation of Hybrid HMM/MLP models. In: Proceedings of ESANN 2001, pp. 383–390 (2002)
10. Esteva, L., Vargas, C.: A Model for Dengue Disease with Variable Human Population. J. Math. Biol. 38, 220–240 (1999)
11. Coutinho, F.A.B., Burattini, M.N., Lopez, L.F., Massad, E.: Threshold Conditions for a Non-Autonomous Epidemic System Describing the Population Dynamics of Dengue. Bulletion of Mathematical Biology 68, 2263–2282 (2006)

A Theoretical Model for the Dengue Epidemic Using Delayed Differential Equations: Numerical Approaches

Andrés Sánchez Pérez, Héctor de Arazoza Rodríguez, Teresita Noriega Sánchez,
Jorge Barrios, and Aymee Marrero Severo

Universidad de La Habana, San Lázaro y L. Vedado.
Plaza de La Revolución. Ciudad de La Habana. 14000, Cuba
asp@matcom.uh.cu

Abstract. We formulate an autonomous dynamical system to model an epidemic outbreak of Dengue fever in which the population of mosquitoes is not directly present. In their place, we consider delayed differential equations. Our model is mainly based on vertical transmission. We found equilibrium points, studied its stability and gave some possible interpretations of the results. Numerical work is present too because we try to fit the parameters with data from a real epidemic.

1 Introduction

Dengue fever and dengue hemorrhagic fever (DF/DHF) are caused by the dengue virus, a flavivirus, and is transmitted by mosquitoes of the genus Aedes (Stegomyia), mainly by the species Aedes aegypti and Aedes albopictus [7]. The mosquito A. aegypti inhabits mainly in houses and bites at any time during the day, which makes it a very efficient vector [4]. A. aegypti is infected by sucking infected human blood, after which follows a short incubation period of approximately eight days. Afterwards the virus begins a replication process inside the mosquito's salivary glands and is ready to infect susceptible humans through any future bite. The mosquito never recovers from the infection, their infective period ends when they die. When an infected mosquito bites a susceptible human, the virus is injected into his or her blood stream and begins an incubation period. In subtropical regions, the disease shows a resurgent pattern with yearly epidemics, which starts typically in the months characterized by heavy rains and heat, peaking some 3 or 4 months after the beginning of the rainy season. In the dry months, the number of cases drop essentially to zero due to the virtual disappearance of the vector. Since the infection reappears in the same regions, it is natural to ask how the virus survives the dry season and one of the possible reasons considered is the vertical transmission in the mosquito population with a fraction of the eggs remaining infected and hatching later into infected females. This vertical transmission has already been suggested by field studies [3]. In addition, in a recent publication, Crochu et al. (2004) demonstrated the integration of flavivirus DNA sequences in the genome of Aedes spp. mosquitoes. This may, in the future, provide further evidence in favor of vertical transmission influencing the pattern of dengue epidemics.

In this work we present and analyse a new model for Dengue's epidemic. Its objective is to avoid dealing with values of mosquitoes' population, which are unknown. With

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 893–900, 2009.
© Springer-Verlag Berlin Heidelberg 2009

this purpose, we assume the hypothesis that reject the transmission of the fever by direct bite of a mosquito, in front of vertical transmission by means of eggs of this mosquito. Since a mathematical point of view, this consideration implies the use of delay differential equations because the infections produced at moment t are due to mosquitoes that have inherited the infection of their mother, infected at moment $t - \tau$. The article is organized in the followimg way: section 2 describes the model stating the variables and the coefficients we use and their meaning in the context of the epidemic, section 3 is dedicated to the study of the stability of the system. In section 4 we fit the values of the parameters to simulate a real epidemic: the Havana 2001 epidemic. Section 5 gives the conclusions of our study.

2 Discussion of a Model

Mathematical modeling of infectious diseases becomes an interesting tool for the understanding of these diseases and for the proposition of strategies. The formulation of the model and the possibility of simulations with parameter estimation, allow tests for sensitivity and comparison of conjectures.

We propose a model based mainly on the idea of delayed differential equations [5], [16] because we do not have any information about the population of infected mosquitoes at the moment of an epidemic. The presence of local periodical maxima in the graphics of some epidemics [3] also suggested us to consider the delay.

We propose that the mosquitoes that bite sick persons at moment $t - \tau$ and become infectious, will have adult and infective descendent at the moment t. This fact is the main one in our concept for the expansion of the epidemic: vertical transmission. In our proposal τ is the time that it takes for the eggs to hatch and for larvae to reach the adult stage. This is a hypothesis proposed by some epidemiological evidence as the main cause for an epidemic to appear and develop.

So, the variation of infected people at moment t, will be proportional to those infected at moment $t - \tau$ and to the susceptibles at t. To indicate this we will use a parameter β.

We shall assume that the population is constant (N) with b the per capita birth rate that we will assume equals the death rate. This assumption in epidemiological models, is relatively valid for diseases of short duration with limited effect on mortality [6]. By simplicity, we will consider the effect of mortality due to Dengue equal to zero. That is we are modeling an epidemic where Dengue Hemorrhagic Fever is not present.

Let S, I, and R be the total number of susceptibles, infectives, and immune or recovered in the human population, respectively [8], [14]. The model is represented schematically by the following diagram.

All those assumptions, lead us to the following equations:

$$\begin{aligned}
\frac{dS}{dt}(t) &= -\beta S(t) I(t-\tau) - bS(t) + bN \\
\frac{dI}{dt}(t) &= \beta S(t) I(t-\tau) - (b+r) I(t) \\
\frac{dR}{dt}(t) &= rI(t) - bR(t)
\end{aligned} \tag{1}$$

Where r is the recovery rate.

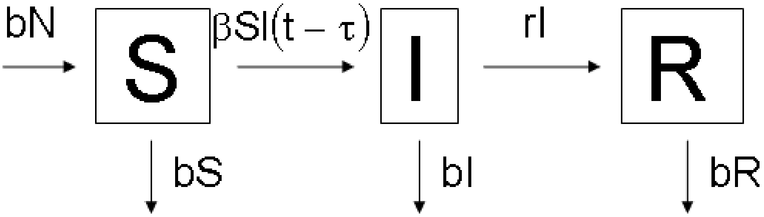


Fig. 1. Flow's diagram of a SIR delay epidemics model

3 Equilibrium Points

The behavior of each group of the population in this case, will be modeled by each component of the solution.

The region of biological interest is $\omega = \{(S, I, R) : S \geq 0, I \geq 0, R \geq 0\}$.

We shall use the following threshold parameter: $R_0 = \dfrac{\beta N}{b + r}$ in the study of equilibrium, stability and the interpretation of results. Next section is devoted to this.

Equilibrium points are constant solutions of a system of differential equations [9]. We can find them by solving the system obtained when we equal to zero all the equations of the differential system. The following theorem give us the equilibrium points with biological interpretation of the system (1).

Theorem 1

1. *System (1) always has the disease free equilibrium* $P_1 = (N, 0, 0)$.
2. *If* $R_0 > 1$ *there exists an endemic equilibrium point,* $P_2 = \left(\dfrac{N}{R_0}, \dfrac{b}{\beta}(R_0 - 1), \dfrac{b}{\beta}(R_0 - 1)\right)$ *and it is unique.*

3.1 Basic Reproduction Number

R_0 is the quantity of new infected caused by an infectious while he remain in this state.

The mean duration of the disease for all the people, that starts our study infected is $\dfrac{1}{r}$ days and the mean time of the infectious period is $\dfrac{1}{b + r}$, taking into account the mortality rate.

If we take the system (1) as a discreet time system, we perceive that one infectious individual causes βN new infections every day in a completely healthy population. And now, it is clear that $R_0 = \dfrac{\beta N}{b + r}$, is the number of new infections produced by one infected person during the period that he/she remains infectious. This is called the basic reproduction index [11].

3.2 Stability

A very useful tool for the study of the stability of equilibrium points [12] is the characteristic equation [13], [10], [2]. Taking $P = (S, I, R)$, we could write our system in a compact way like: $P'(t) = f(P(t), P(t-\tau))$. Its characteristic equation for the equilibrium point $P^* = (S^*, I^*, R^*)$ is

$$\begin{aligned}
\det\left(\lambda I - \frac{\partial f(u,v)}{\partial u} - \frac{\partial f(u,v)}{\partial v} e^{-\lambda\tau}\right)_{u=v=P^*} &= 0 \\
\det\left(\begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix} - \begin{pmatrix} -\beta I^* - b & 0 & 0 \\ \beta I^* & -(b+r) & 0 \\ 0 & r & -b \end{pmatrix} - \begin{pmatrix} 0 & -\beta S^* & 0 \\ 0 & \beta S^* & 0 \\ 0 & 0 & 0 \end{pmatrix} e^{-\lambda\tau}\right) &= 0 \\
\det\begin{pmatrix} \lambda + \beta I^* + b & \beta S^* e^{-\lambda\tau} & 0 \\ -\beta I^* & \lambda + b + r - \beta S^* e^{-\lambda\tau} & 0 \\ 0 & -r & \lambda + b \end{pmatrix} &= 0 \\
(\lambda + b)\det\begin{pmatrix} \lambda + \beta I^* + b & \beta S^* e^{-\lambda\tau} \\ -\beta I^* & \lambda + b + r - \beta S^* e^{-\lambda\tau} \end{pmatrix} &= 0 \\
(\lambda + b)\left[\lambda^2 + \lambda(2b + r + \beta I^*) - \lambda\beta S^* e^{-\lambda\tau} - b\beta S^* e^{-\lambda\tau} + b^2 + br + \right. & \\
\left. + b\beta I^* + r\beta I^*\right] &= 0 \qquad (2)
\end{aligned}$$

Stability without delay. We shall present some results for the system without delay, or delay zero. First we will consider the non epidemic point $P_1 = (N, 0, 0)$. The jacobian for the system (1) with $\tau = 0$ at P_1, has the eigenvalues $-b$ (double) and $(b+r)(R_0 - 1)$. If we find the jacobian at point P_2 we get one eigenvalue equal to $-b$ and the other two have sum equal to $-b(\beta(R_0 - 1) + 1)$, and product equal to $(b + r)\beta b(R_0 - 1)$. Thus we have:

Theorem 2

1. *If $R_0 \leq 1$ then the disease free equilibrium is the only one in the region of interest and it is locally asymptotically stable.*
2. *If $R_0 > 1$ then P_1 is unstable, P_2 exists in the region of interest and it is locally asymptotically stable.*

The interpretation is clear: if an epidemic starts with a relatively little number of cases and each one of those leaves less than one new sick person, in average, and this happens with each new infective, then the number of new infections (the incidence) decreases. On the contrary, if each one of the infectious persons, leaves more than one new infectious person, then the number of new infections increases.

Stability with delay. Let us analyze now the system in a more general way, with any fixed delay. Its behavior depends, like in the previous section, on the roots of a determinant of order 2.

$$\lambda^2 + \lambda(2b + r + \beta I^*) - \lambda\beta S^* e^{-\lambda\tau} - b\beta S^* e^{-\lambda\tau} + b^2 + br + b\beta I^* + r\beta I^* = 0 \qquad (3)$$

For the equilibrium point, P_1, (3) becomes:

$$\lambda^2 + \lambda(2b + r) - \lambda\beta N e^{-\lambda\tau} - b\beta N e^{-\lambda\tau} + b^2 + br = 0$$

Reagrouping:

$$\begin{aligned}\lambda^2 + \lambda b + \lambda b + b^2 + \lambda r + br - \lambda\beta N e^{-\lambda\tau} - b\beta N e^{-\lambda\tau} &= 0 \Leftrightarrow\\ \lambda(\lambda+b) + b(\lambda+b) + r(\lambda+b) - \beta N e^{-\lambda\tau}(\lambda+b) &= 0 \Leftrightarrow\\ (\lambda+b)\left(\lambda + b + r - \beta N e^{-\lambda\tau}\right) &= 0\end{aligned}$$

Again we have a root equal to $-b$ that we will factor from the equation.

We should focus on the transcendent equation

$$\lambda + b + r - \beta N e^{-\lambda\tau} = 0$$

Let $\lambda = x + iy$, where $x, y \in \mathbb{R}$.

$$\begin{gathered}x + iy + b + r - \beta N e^{-x\tau - iy\tau} = 0\\ (x + b + r + iy)\left[\cos(y\tau) + i\sin(y\tau)\right] = \beta N e^{-x\tau}\end{gathered}$$

It is sufficient to find the roots of

$$g^{+}(x) = x + b + r - \beta N e^{-x\tau}$$

or

$$g^{-}(x) = x + b + r + \beta N e^{-x\tau}$$

Using differential calculus and upper and lower bounds for trigonometric and exponential functions, we get the following result:

Theorem 3. *If* $\max\{\beta N\tau, R_0\} < 1$, *then* P_1 *is asymptotically stable.*

We could say that the epidemic is under control if the basic reproduction number of the epidemic, is less than one, and the number of new infections produced during τ days is less than one too.

4 Simulations

We used the data from the 2001 epidemic in the city of Havana to estimate the values of the parameters β, b and r. For this purpose, we worked with the "Parameter Estimation Tool"[1], [15], a very useful program made for Matlab. The initial values, as well as its bounds where obtained from [3]. We obtained the following estimate for the unknown parameter β:

$$\beta = 1.56 \cdot 10^{-7}.$$

The curves that correspond to estimated values (using this value for β) for the number of susceptibles and infected persons in the population, are too far from the data values. In figures 2 and 3 we present the curves of real data and the curves determined by the optimal parameter found.

Figure 4 is the complete epidemic curve of the Havana 2001 data.

The failure to fit the model to the data indicates that we need to reformulate the problem or that one of our hypothesis is not true. The main problem from our point

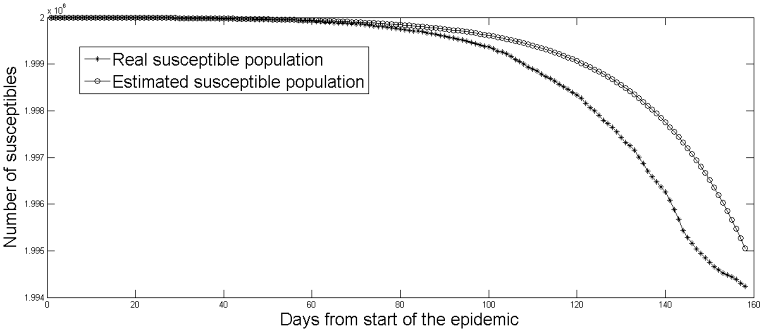


Fig. 2. Results of the simulation for the susceptible population

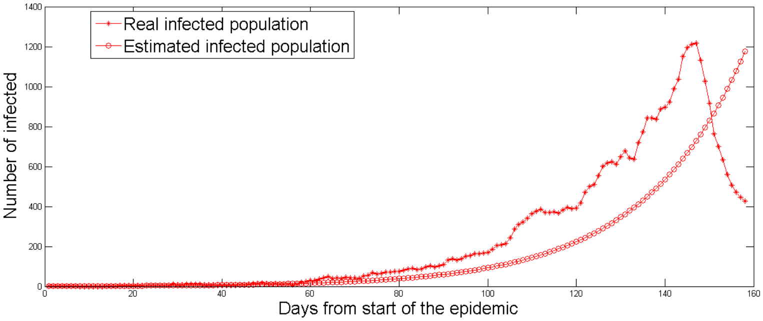


Fig. 3. Results of the simulation for the infected population

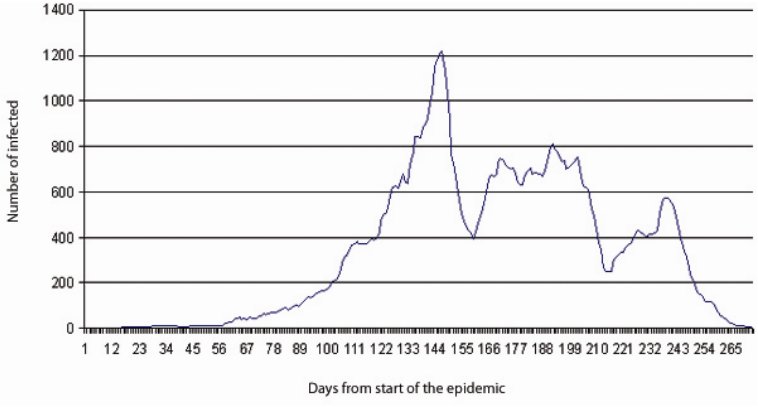


Fig. 4. Infected cases reported in Havana 2001

of view is the value for τ, the delay and to consider the parameter β, specifically, like a constant. We took τ as the time it takes the mosquito eggs to hatch and the larvae to reach adult stage. This might not be the case if the percentage of infected eggs is not as high as suggested in [3]. In the model, the intervention of the Health System (fumigation and other) to control and eliminate the epidemic is not taken into account. This is reflected in the model by the continuous increase of the infected populations.

In the future more simulation work is needed, also to consider different values for the parameters at different moments of the epidemic. Also a more extensive search for the bounds of the parameters, and the possibility of a nonconstant delay τ.

5 Concluding Remarks

The non existence of data on the mosquito population during an epidemic of dengue fever, still represents an obstacle for the modelling and evaluation of the parameters involved in the dynamics of the epidemic, mainly the rate at which new infections are produced. We presented a model based on the hypothesis that the epidemic is maintained basically through the vertical transmission of infection in the mosquitoes. The bad fit of the model to the data suggests that this is not the case, that the dynamics of the epidemic is sustained not by vertical transmission in the mosquito population, but by another factor.

Further tests on the values for τ would give us an idea of where to look for the "motor"of the epidemic. The model should be enlarged to consider several possible values for the delay and even a non constant value for the delay and the parameter β. This will be done in future work.

Nevertheless the stability results obtained for the model that we proposed are valid.

Acknowledgments. This work has been partially supported by the AECID (Agencia Española de Cooperación Internacional para el Desarrollo), Project D/017218/08.

Dr. Héctor de Arazoza was Invited Professor at the Laboratoire Paul Painlevé of the University of Lille, France, while preparing the manuscript.

The authors wishes to thank Dr. Gonzalo Joya Caparrós for valuable remarks and discussions during the preparation of this paper.

References

1. Barrios, J.: Estimación de parámetros en modelos epidemiológicos de VIH/SIDA en Cuba. M.Sc. thesis. University of Havana (2007)
2. Busenberg, S., Cooke, K.: Vertically Transmitted Diseases, Models and Dynamics. Springer, Heidelberg (1993)
3. Coutinho, F.A.B.: Threshold Conditions for a Non-Autonomous Epidemic System Describing the Population Dynamics of Dengue. Bulletin of Mathematical Biology 68, 2263–2282 (2006)
4. Derouich, M.: A model of dengue fever. BioMedical Engineering OnLine 2, 4 (2003)
5. Driver, R.D.: Ordinary and Delay Differential Equations. Springer, Heidelberg (1977)

6. Esteva, L., Vargas, C.: A model for dengue disease with variable human population. J. Math. Biol. 38, 220–240 (1999)
7. Esteva, L., Vargas, C.: Coexistence of different serotypes of dengue virus. J. Math. Biol. 46, 31–47 (2003)
8. Forde, J.E.: Delay Differential Equation Models in Mathematical Biology (2005)
9. Hirsch, M.W., Smale, S.: Differential Equations, Dynamical Systems and Linear Algebra. Academic Press, Inc., San Diego (1974)
10. Liz, E.: Sobre ecuaciones diferenciales con retraso, dinámica de poblaciones y números primos (2001)
11. Lounes, R.: Modèles d'épidémie: point de vue déterministe. Notas de curso, Université Paris Descartes (2006)
12. Marín, J.: Propiedades Cualitativas de un Modelo Depredador-Presa con Retardo. Divulgaciones Matemáticas 9(2), 123–136 (2001)
13. Roussel, M.R.: Delay-differential equations (2004)
14. Murray, J.D.: Mathematical Biology, 2nd edn. Biomathematics Texts, vol. 19. Springer, Berlin (1980)
15. Shampine, L.F., Thompson S.: Solving Delay Differential Equations with dde23 (2000)
16. Stépán, G.: Retarded dynamical systems: stability and characteristic functions. Longman Scientific and Technical, UK (1989)

System Identification of Dengue Fever Epidemics in Cuba*

Esther García-Garaluz[1], Miguel Atencia[2], Francisco García-Lagos[1],
Gonzalo Joya[1], and Francisco Sandoval[1]

[1] Departamento de Tecnología Electrónica
[2] Departamento de Matemática Aplicada
Universidad de Málaga (Spain)
Campus de Teatinos, 29071 Málaga, Spain
`megg@uma.es`

Abstract. The objective of the work described in this paper is twofold. On the one hand, the aim is to present and validate a model of Dengue fever for the Cuban case which is defined by a delay differential system. Such a model includes time-varying parameters, which are estimated by means of a method based upon Hopfield Neural Networks. This method has been successfully applied in both robotic and epidemiological models described by Ordinary Differential Equations. Therefore, on the other hand, an additional aim of this work is to assess the behaviour of this neural estimation technique with a delay differential system. Experimental results show the ability of the estimator to deal with systems with delays, as well as plausible parameter estimations, which lead to predictions that are coherent with actual data.

1 Introduction

System identification can be defined as the determination of the internal properties of a system from measurable outputs [1]. In models that contain parameters, fitting the values of the model parameters is referred to as parametric identification or parameter estimation [2]. The herein proposed method to perform the estimation is based on Hopfield Neural Networks. This method, which allows for a time-variant estimation of the parameters, has been successfully applied in previous works to models for robotics [3] and the AIDS/HIV epidemics in Cuba [4,5]. In this work, the estimator is applied to a model with a new feature, a delay differential system.

The application of Mathematics to Epidemiology is a task that has been performed for a long time [6,7]. Building models of infectious diseases, validating them and estimating their parameters, allows, on one hand, to predict their evolution, and on the other hand, to develop sanitary policies in order to fight the

---

* This work has been partially supported by the Agencia Española de Cooperación Internacional para el Desarrollo (AECID), project no. D/017218/08. Thanks are due to Dr. Hector de Arazoza for his useful suggestions and, in particular, for providing the model of Dengue epidemics.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 901–908, 2009.
© Springer-Verlag Berlin Heidelberg 2009

diseases. Dengue fever is an infectious disease which has reappeared in subtropical regions of the American and African continent in the last decades. Therefore, the identification of the Dengue fever system has become an interesting study to be performed.

The work presented in this paper has several objectives. Firstly, it defines a model of the Dengue fever epidemics in Havana (Cuba). Conventional models of Dengue take into account the infection vector by defining separate populations of humans and mosquitoes. The comparison of these models to real data is hindered by the difficulty to measure the population of mosquitoes. The novelty of the proposed models is that only human populations are comprised, whereas the mosquito incubation time is accounted for by means of a delay. Thus, all populations are, in principle, measurable, which is accomplished at the cost of an increased mathematical complexity of the model. Therefore, the results of the estimation allow for assessing the validity of the model, as long as the model simulations show an acceptable adjustment to real population data. Furthermore, the ability of the Hopfield Neural Networks estimator to identify delay differential systems is analysed by means of simulation results.

The structure of the rest of the paper is as follows. In Section 2, a model of Dengue fever for the Cuban case is described; the model being based on delay differential equations. Section 3 describes the system identification method by means of Hopfield Neural Network, and the way the estimation problem is solved by an optimization method is presented. Simulation results and their meaning are presented in Section 4. Finally, conclusions and future works put an end to the paper in Section 5.

2 Model of Dengue Fever in Cuba

Dengue fever is a viral infectious disease which is mainly transmitted by the *Aedes Aegypti* mosquitoes. Several models have been previously proposed to describe Dengue epidemics [7,8]. These models include mosquitoes populations, which is a severe drawback for both model validation and prediction, because of the lack of data. In order to model the Dengue fever a compartmental model has been figured out, where the population is divided into people who are susceptible (S) to the disease, people who are infected (I) and people who have recovered (R) and are thus immune to the Dengue fever. Therefore, such a model is referred to as a SIR model. In the case of Dengue, the use of a SIR model is appropriate, since there is no need to consider the mortality caused by Dengue fever, which is mainly caused by Haemorrhagic Dengue fever. Indeed, for short time periods, only one strain of the virus can be considered, and in that case the infected people become immune, whereas Haemorrhagic Dengue is thought to be associated to different virus strains.

The Dengue fever in Cuba has been modelled as a population dynamics defined by the delay differential equation system in Equation (1), which avoids the use of mosquito populations by including infection information in a time-dependent parameter and through the delay term:

$$
\begin{aligned}
\frac{dS}{dt} &= -\beta(t)\frac{S(t)}{N}\sum_{\tau=t_1}^{\tau=t_2} I(t-\tau)\, e^{-\mu_M \tau} + b\, N(t) - \mu_H\, S(t) \\
\frac{dI}{dt} &= \beta(t)\frac{S(t)}{N}\sum_{\tau=t_1}^{\tau=t_2} I(t-\tau)\, e^{-\mu_M \tau} - r\, I(t) - \mu_H\, I(t) \\
\frac{dR}{dt} &= r\, I(t) - \mu_H\, R(t)
\end{aligned}
\tag{1}
$$

where the parameter $\beta(t)$ is the infection rate, μ_H is the natural human mortality, μ_M is the mosquito mortality, b is the natural birthrate and r is the Dengue recovery rate. The parameters in the model are related to the transition rate among the different populations. The parameter β carries a special importance because it provides information about mosquitoes population, and it is the tool that allows to circumvent the dependence on mosquitoes population data. Contrarily to the rest of the parameters in the model, it is not possible to estimate the β parameter with statistical methods, so it constitutes the target of this work and it will be estimated with the neural network. This model has an intuitive rationale which allows to analyse the meaning of the equations:

- S equation. The growth of the susceptible population is caused by the births $+b\, N$ and its decrease is given by the human mortality $-\mu_M\, S(t)$ and the new infections $-\beta(t)\frac{S(t)}{N}\sum I(t-\tau)e^{-\mu_M \tau}$.
- I equation. The growth of this population is caused by the new infections $+\beta(t)\frac{S(t)}{N}\sum I(t-\tau)e^{-\mu_M \tau}$ and the decrease is established by the recovery $-r\, I(t)$ and the human mortality $-\mu_H\, I(t)$.
- R equation. The growth of this population is caused by the recently recovered individuals $+r\, I(t)$ and its decrease is caused mortality of recovered people $-\mu_H\, R(t)$.

The use of delay equations results from considering the mosquito incubation period of Dengue virus, which is known to be about 8 days. After the incubation, an infected mosquito can transmit the virus every time it bites a susceptible human. The maximum life span of the mosquito is about 15 days. Therefore, the summation extends from $t_1 = 8$ to $t_2 = 15$ days, where each component is weighted by an exponential term to adjust it to the mosquito survival rate. In other words, each term in the summation of the form $I(t-\tau)\, e^{-\mu_M \tau}$ accounts for a fraction of the infected mosquitoes, which stemmed from biting an infected human τ days ago. Therefore, mosquito populations, which cannot be measured, appear implicitly in the model through the time-dependent parameter β, thanks to the introduction of delay terms.

3 System Identification with Hopfield Neural Networks

3.1 Parametric Identification with Continuous Hopfield Networks

The estimation method that is used in this work is based upon the combination of two ideas: on one hand, estimation can be formulated as an optimization problem [2]; on the other hand, Hopfield Neural Networks [9] are able to solve optimization problems [10]. The proposed method, in particular, stems from the Abe formulation for Hopfield Neural Networks [11,12], which is defined by the system shown in Equation (2):

$$\frac{du_i}{dt} = \sum_i \omega_{ij} s_j - I_i \,; \qquad s_i = tanh\left(\frac{u_i(t)}{\gamma}\right) \tag{2}$$

where s_i is the state of the neuron i, ω_{ij} is the weight of connection between the i-th and the j-th neuron, I_i is the bias of the i-th neuron, and γ controls the slope of the hyperbolic tangent. The system resulting from this formulation has been proved to be stable, due to the existence of the Lyapunov function shown in Equation (3).

$$V(s) = -\frac{1}{2}\sum_i \sum_j \omega_{ij} s_i s_j + \sum_i I_i s_i \tag{3}$$

The application of Hopfield Neural Networks to optimization is carried out by matching the Lyapunov function to the target function to be optimized, thus the trend of the network state is to reach a minimum of the function.

Consider a system defined in a generic Linear In the Parameters (LIP) form $y = A(x)\theta$, where $A(x)$ is a matrix of non-linear functions depending on the states (represented by x), y is a vector depending on the states and their derivatives and θ is the parameter vector. Estimation of the parameter vector θ can be formulated as the obtention of a value $\widehat{\theta}$ that minimizes the prediction error $e = y - A(x)\widehat{\theta}$. After some algebra, the target function shown in Equation (4) is obtained

$$V = \frac{1}{2} e^\top e = \frac{1}{2}\widehat{\theta}^\top A^\top A \widehat{\theta} + \widehat{\theta}^\top(-A^\top y) + \frac{1}{2}\|y\|^2 \tag{4}$$

Matching Lyapunov function in Equation (3) and the target function in Equation (4), the weights and the biases for the estimator are obtained, as shown in Equation (5).

$$W = -A^\top A \,; \qquad I = -A^\top y \,; \qquad s = \theta \tag{5}$$

The development of this estimation method deserves a number of remarks. Firstly, contrarily to most of the conventional methods, time-variant parameters can be estimated in a natural way. Besides, the presence of hyperbolic tangent forces the estimations to be confined in a unit hypercube $[-1,1]^n$. This is not a restrictive condition because if the parameters to be estimated are in another range, a linear transformation can be applied, although the range of the parameter value must be known in advance. Therefore, the estimation is automatically bounded, as opposed to the gradient methods which often needs to be complemented with bounding algorithms leading to a higher computational cost.

3.2 Application of the Neural Estimator to Discrete Systems

The estimation method defined in Section 3.1 is described for continuous systems. However, the epidemiological system can not be assumed to follow this restriction because of the discrete nature of population data. A discrete version of the above method is obtained by means of an explicit discretization of the model, as shown in Equation 6.

$$x_{n+1} = x_n + \Delta t\, A(x_n)\theta \tag{6}$$

Following the same procedure as above, the weights and biases of the network equation are obtained, which correspond to the discrete version of the estimator. These weights are shown in Equation (7):

$$W = -\Delta t^2\, A^\top A\,; \qquad I = -\Delta t\, A^\top (x_{n+1} - x_n) \tag{7}$$

The estimation method for discrete systems is able to deal with time-varying parameters, as its continuous counterpart.

The equations of the estimation method must be generalized in order to deal with a delay differential model. Paradoxically, discretization helps, rather than hinders, the consideration of delays. Indeed, the only change consists in the generalization of the A matrix, so that it includes past states, i.e. $A = A(x_n, x_{n-i}, x_{n-k}, ...)$ where i and k are the lags. Therefore, the system is modelled by Equation (8).

$$x_{n+1} = x_n + \Delta t A(x_n, x_{n-i}, x_{n-k} ...)\theta \tag{8}$$

In this way, the estimation method with Hopfield Neural Network is completely defined for an epidemiological system that comprises delays, which is a delay differential equation from the mathematical point of view, by using the discrete equations of the method and the generalization of A matrix.

4 Experimental Results

In order to apply the estimator with Hopfield Neural Networks to the Dengue fever model in Equation (1), a data series of the Dengue populations in Cuba is available. It consists of a 274-day long database of new infections by Dengue fever in Havana. From this register and the known characteristics of the disease (incubation period of Dengue fever in humans and recovery time) it was possible to build a complete database of all the populations in the model: susceptible, infected and recovered people.

Some consideration must be made to describe the performed experiments. One of them is that most of the parameters in the model are supposed to be known, because they are characteristic either to the epidemic or to the population and can be easily estimated by other methods (i.e. human mortality and birth rate, or dengue recovery rate can be estimated by statistical methods). The only parameter being estimated is thus $\beta(t)$. Another consideration to be made

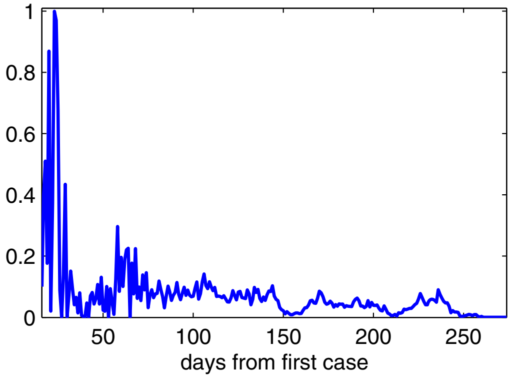


Fig. 1. Estimation of parameter β with Hopfield Neural Networks for Dengue model in Havana (Cuba)

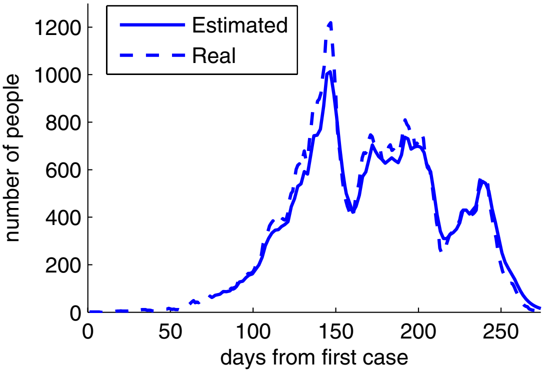


Fig. 2. Real and predicted evolution of infected population of Dengue fever in Havana (Cuba)

is that, in this case, the global population is considered constant for this time period.

The results of the estimation for the parameter β are represented in Figure 1. As it was mentioned in the comments below Equation (2), the parameter β is related to the infected mosquitoes population, so an approximation to that population can be figured out. The shape of the estimated values for β, in coherence with the model definition, follows the same behaviour as the infected population, but its trajectory is smoother. For instance, the valleys, near the days 160th and 220th, in the number of infections can be traced to corresponding falls in the estimation of β, which can be interpreted as a decrease in the number of infected mosquitoes due to the success of fumigation campaigns.

In order to make a validation for both the model and the parameter estimation, a test was performed. It consists on integrating the model equations from a real initial point using the estimation values. In order to choose an initial point with enough significance, the integration was started from the instant when the infected population value is 10% of the average value of the serial data. The validation results for infected population are shown in Figure 2 where the real and predicted population are plotted together.

Although further validation tests are needed, the obtained results are satisfactory since the estimation provides a good fit in the validation test between the real and the predicted infected populations. Indeed, these results support the need to model the infected mosquito population can be implicitly represented by a time-varying parameter. Based on this rationale, it would be interesting to extend the model in future works in order to obtain an explicit estimation of infected mosquito population.

5 Conclusions and Future Works

In this work, a new model for Dengue fever has been presented. This model is focused on Dengue epidemics in Havana (Cuba). From the epidemiological point of view, the interest of the model relies on the absence of mosquitoes populations, which are impossible to measure. Instead, the model comprises a time-varying parameter, which is an implicit indicator of the mosquitoes population. From the mathematical point of view, the introduction of delay differential equations poses a challenging setting for estimation. This feature suggests the need to prove the estimator that was designed from Hopfield Networks Neural, assessing its ability to deal with generalized differential systems. The obtained results and the validation test are satisfactory and encourage a new line of work.

Some directions of future research can be proposed from the herein described work. First, the use of delay differential equations with a Neural Network estimator well deserves a deeper mathematical analysis. On the other hand, the sensitivity analysis of the model would provide a powerful tool which would complement the theoretical research. Besides, additional validation tests could be performed and a model extension could lead to an explicit estimation of mosquito population.

References

1. Ljung, L.: System Identification. Theory for the User. Prentice-Hall, Englewood Cliffs (1999)
2. Unbehauen, H.: Some New Trends in Identification and Modeling of Nonlinear Dynamical Systems. Applied Mathematics and Computation 78, 279–297 (1996)
3. Atencia, M.A., Joya, G., Sandoval, F.: Parametric Identification of Robotic Systems with Stable Time-varying Hopfield Networks. Neural Computing & Applications 13(4), 270–280 (2004)
4. Atencia, M.A., Joya, G., Sandoval, F.: Modelling the HIV-AIDS Cuban Epidemics with Hopfield Neural Networks. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2687, pp. 449–456. Springer, Heidelberg (2003)

5. Atencia, M.A., Joya, G., García-Garaluz, E., Arazoza, H., Sandoval, F.: Estimation of the Rate of Detecction of Infected Individuals in an Epidemiological Model. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 948–955. Springer, Heidelberg (2007)
6. Bailey, N.T.: The mathematical theory of infectious diseases and its applications. Ch. Griffin and Company LTD (1975)
7. Esteva, L., Vargas, C.: A model for dengue disease with variable human population. Journal of Mathematical Biology 38(3), 220–240 (1999)
8. Coutinho, F., Burattini, M., Lopez, L., Massad, E.: Threshold conditions for a non-autonomous epidemic system describing the population dynamics of dengue. Bulletin of Mathematical Biology 68(6), 2263–2282 (2006)
9. Hopfield, J.: Neural Networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79, 2554–2558 (1982)
10. Tank, D., Hopfield, J.: 'Neural' Computation of Decisions in Optimization Problems. Biological Cybernetics 52, 141–152 (1985)
11. Abe, S.: Theories on the Hopfield Neural Networks. In: Proc. IEE International Joint Conference on Neural Networks, vol. I, pp. 557–564 (1989)
12. Atencia, M.A., Joya, G., Sandoval, F.: Dynamical Analysis of Continuous Higher Order Hopfield Networks for Combinatorial Optimization. Neural Computation 17(8), 1802–1819 (2005)

HIV Model Described by Differential Inclusions

Jorge Barrios[1,2,⋆], Alain Piétrus[1], Aymée Marrero[2], and Héctor de Arazoza[2]

[1] Laboratoire Analyse, Optimisation, Contrôle
Université des Antilles et de la Guyane
Département de Mathématiques et Informatique
Campus de Fouillole
F-97159 Pointe-à-Pitre–France
`apietrus@univ-ag.fr`
`http://www.univ-ag.fr/aoc`
[2] Universidad de La Habana
Facultad de Matemática y Computación
San Lázaro y L, Vedado, La Habana–Cuba
`{jbarrios,aymee,arazoza}@matcom.uh.cu`
`http://www.matcom.uh.cu`

Abstract. Infected population size estimation is a common problem in HIV/AIDS epidemic analysis and it is the most important aspect for planning appropriate care and prevention policies. Some Ordinary Differential Equations models of HIV epidemic in Cuba considering the Contact Tracing strategy have been described in previous works. In this paper we present a HIV/AIDS model described by Differential Inclusions. Also, we establish a mathematical framework allowing us to make suitable prediction of the size HIV infected population at a future time.

Keywords: Reachable Sets, Differential Inclusion, Contact Tracing, HIV, AIDS.

1 Introduction

The Human Immunodeficiency Virus (HIV) infection, leads to Acquired Immunodeficiency Syndrome (AIDS). When antibodies to HIV are detected the patient is infected and said to be seropositive or HIV-positive.

HIV moves into its third decade and continues to be a worldwide health problem. In the early 1980s, an AIDS diagnosis meant a death sentence. But with the advent of HIV medications, people can stay healthy, live longer, and have hope after their HIV diagnosis. One of the principals drawbacks in this direction is the high costs of treatment. Estimating how many people are living with HIV is an important question for appropriate epidemiological interventions and efficient resource allocation.

In Cuba, the first AIDS case was diagnosed in April of 1986. This signaled the starting point of HIV epidemic in the country, although some HIV-positive

---

[⋆] This work has been partially supported by the AECID (Agencia Española de Cooperación Internacional para el Desarrollo), Project D/017218/08.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 909–916, 2009.
© Springer-Verlag Berlin Heidelberg 2009

persons had been detected at the end of 1985. At the early stages the Cuban Government had initiated several preventive measures to try to contain the possible outbreak of the epidemic. Once the first cases were confirmed, a program based on the experience with other sexually transmitted diseases was started. The AIDS program also had, among other measures, the tracing of sexual contacts of known HIV-positive persons, to prevent the spreading of HIV. When a person is detected as living with HIV, an epidemiological interview is carried out by the Epidemiology Department of his municipality or by his family doctor as part of the Partner Notification Program (or Contact Tracing Program). After this interview the Epidemiology Department tries to locate the sexual partners of the person through the network of Healt System. The person living with HIV usually does not participate in this process, though they normally help in notifying their present partners. Trying to locate the sexual partners is a very complex job and one that in some cases take a lot of time. This task is one of high level of priority for the Health System, and it is under constant supervision to try to determine how effective it is in the prevention of the spread of HIV.

In [1,2] and other papers, mathematical models of HIV epidemic allowing for Contact Tracing have been presented. The objective was to model the Contact Tracing aspect of HIV detection system, to try to obtain some information that could be useful to the Health System in Cuba in evaluating the way the program is working, and to ascertain its usefulness in terms of intervention and treatment of HIV.

Models appearing in [2] are described by ordinary differential equations and the unknown parameters involved in them are assumed constant over time. Generally, in reality some of these parameters are not constant and can depend on time and/or the system state. From that point of view, we are necessarily faced with a problem of uncertainty in the dynamics.

In [7], an adaptive estimation technique is used, which is base on Recurrent Hopfield Neural Networks; this technique follows to considers some parameters as non-constant under the hypothesis of a relatively slow variability. However, using this method it is not possible to find all possible solutions of the dynamics under the uncertainties. In addition, for this method to work, data on all the state variables of the model is needed, and this is not the case of the system presented in this paper.

The problems of uncertainties in ordinary differential equations can be addressed as a differential inclusions approach. Differential inclusions first appeared in papers from Zaremba (1936) and Marchaud (1938). Although a relatively new theory has been widely described and developed in the comprehensive work by Aubin and Cellina in 1984 [3] and others.

The objective of this paper is to reformulate an HIV/AIDS model from [2] as a Differential Inclusion. We establish a mathematical framework that will allow us to answer the initial question indicated above, and make a suitable prediction of the HIV infected persons, that do not know they are infected, at a future time t. We do not give numerical results, they will be presented in a future work.

The paper is organized as follows. Section 2 presents an HIV/AIDS model described by a differential inclusion with the respective epidemiological meaning of the parameters and variables in the model. Section 3 presents an approach based in a discrete approximation of the reachable set of the model to make predictions. Section 4 concludes the paper.

2 The Model

We will consider in the model the following state variables as function of time t:

- $X(t)$: is the number of HIV-infected persons that do not know they are infected at time t.
- $Y_1(t)$: is the number of HIV-infected persons that know they are infected at time t and were detected in a random type search.
- $Y_2(t)$: is the number of HIV-infected persons that know they are infected at time t and were detected through contact tracing.
- $Z(t)$: is the number of persons with AIDS at time t.

With the following constant coefficients:

- N: sexually active population.
- α: the rate of recruitment of new HIV-infected persons infected by X.
- α': the rate of recruitment of new HIV-infected persons infected by $Y_1 + Y_2$.
- k_1: the rate at which the unknown HIV-infected persons are detected by the system, independently of other HIV-positive persons (through "random" screening).
- β: the rate at which the undetected HIV-positive persons develop AIDS, the reciprocal of the mean incubation.
- β': the rate at which the detected HIV-positive persons develop AIDS, the reciprocal of the mean time it takes to go from $Y_1 + Y_2$ to Z.
- μ: the mortality rate of the sexually active population.
- μ': the mortality rate of the population with AIDS.

And, to reflect an non-exactly known parameter we assume the following uncertainty as function of time t:

- $k_2(t)$: the rate at which unknown HIV-infected persons are detected by the system through contact tracing.

Parameter k_2 is taken as an uncertainty because it depends on the effort displayed by the Health System in searching and locating the sexual partners that have been reported by those already detected as HIV positive.

Considering t_0 and T the initial and final time were the model is defined, with $T > t_0$, since $k_2(t)$ is unknown but bounded in a known real interval $K = [a, b]$ (taken from [2]), and

$$\mathcal{K} = \{\ k_2(\cdot) \mid \quad k_2(t) \in K \ for\ almost\ all\ elements\ t \in [t_0, T]\ \}$$

is the set of all measurable functions with values in K, the ordinary differential equation model presented in [2] becomes the dynamics described by the differential inclusion

$$\underbrace{\begin{bmatrix} \frac{\partial X}{\partial t} \\ \frac{\partial Y_1}{\partial t} \\ \frac{\partial Y_2}{\partial t} \\ \frac{\partial Z}{\partial t} \end{bmatrix}}_{=\dot{x}} \in \left\{ \underbrace{\begin{matrix} \alpha N X + \alpha^{'} N(Y_1+Y_2) - (k_1+\mu+\beta)\,X - k_2 \frac{X(Y_1+Y_2)}{X+Y_1+Y_2} \\ k_1 X - \left(\mu+\beta^{'}\right) Y_1 \\ k_2 \frac{X(Y_1+Y_2)}{X+Y_1+Y_2} - \left(\mu+\beta^{'}\right) Y_2 \\ \beta X + \beta^{'}(Y_1+Y_2) - \mu^{'} Z \end{matrix}}_{=g(x,k_2)} \right\}_{k_2 \in K}, \tag{1}$$

were $x : [t_0, T] \to \mathbb{R}^4,\ g : \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}^4$.

We also know, the initial value vector

$$\begin{bmatrix} X \\ Y_1 \\ Y_2 \\ Z \end{bmatrix} (t_0) = \begin{bmatrix} 230 \\ 68 \\ 26 \\ 3 \end{bmatrix} \tag{2}$$

for the model, corresponding to the data at the beginning of the HIV epidemic in Cuba in 1986 of the respective groups. The value of $X(t_0)$ was estimated from the number of HIV positives that were found after 1986 and were already infected at that time.

The system is only considered in the region

$$D = \{X > 0, Y_1 > 0, Y_2 > 0, Z > 0\}\ .$$

The value $k_2(\tau)$ of each function $k_2(\cdot) \in \mathcal{K}$ at time $\tau \in [t_0, T]$, $t_0 < \tau \leq T$ makes right part of (1) a map which associates a point with a set in all $\tau \in [t_0, T]$. This kind of map is called a set valued map or multifunction. A clear introduction to these versatile and powerful concept can be found in [4].

Data for variables Y_1, Y_2 and Z are available from 1986 to 2005, but of course we do not have data for X. We will assume here as constants the estimated values of $N, \alpha, \alpha^{'}, k_1, \beta, \beta^{'}, \mu, \mu^{'}$, available from [2] and [1], that were obtained following different strategies using the available data.

For simplicity we will assume here only k_2 as uncertainty although others parameters as $k_1, \alpha, \alpha^{'}$, could also be considered in such a way.

3 Approximating the Reachable Set of the Model

Consider the initial value problem (IVP) formed by (1) and (2). By definition, a solution of this IVP is any absolutely continuous function $x : [t_0, T] \to \mathbb{R}^4$ that satisfies the initial condition (2) and the differential inclusion (1) for almost all $t \in [t_0, T]$ (see [3]). In this case for each different measurable realization $k_2(\cdot)$ we have a solution $x(\cdot)$ for the IVP (see [3],p.91).

Let us denote by $\mathcal{X}$ the set of all solutions or trajectories $x(\cdot)$ of the IVP for the differential inclusion (1). We defined the reachable set $R(\tau)$ at the time τ as

$$R(\tau) \doteq \{x(\tau) : x(\cdot) \in \mathcal{X}\} . \tag{3}$$

In order to numerically solve the IVP described above, some effective numerical methods are needed. In the following, a set-valued version of a specific Runge-Kutta scheme, known as Heun scheme (see e.g. [5]) is reviewed.

HEUN's METHOD. *For $N \in \mathbb{N}' \subset \mathbb{N}$ choose a grid*

$$t_0 = t_0 < t_1 < ... < t_N = T \tag{4}$$

with a stepsize

$$h = \frac{T - t_0}{N} = t_j^N - t_{j-1}^N \quad (j = 1, ..., N) . \tag{5}$$

Let the starting approximation y_0 equal to the initial value vector (2) and for $j = 0, ..., N-1$ solve the difference inclusion

$$y_{n+1} \in y_n + \frac{h}{2}\{g(y_n, k_2) + g(y_n + hg(y_n, k_2), k_2) : k_2 \in K\} \quad n = 0, ..., N . \tag{6}$$

Here, as usual $\mathbb{N}'$ denotes a subsequence of $\mathbb{N}$ converging to infinity.

Let us denote by $\mathcal{Y}_N$ the set of all trajectories $y^N(\cdot)$ of the discrete inclusion (6), the reachable set $R_N(n)$ of (6) at stage n is

$$R_N(n) = \{y_n : y^N = (y_0, ..., y_N) \in \mathcal{Y}_N\} . \tag{7}$$

The reachable set $R_N(N)$ of (6) can be considered as resulting from the recurrent procedure

$$R_N(n+1) = R_N(n) + \frac{h}{2}\{g(R_N(n), k_2) + g(R_N(n) + hg(R_N(n), k_2), k_2) : k_2 \in K\}, \tag{8}$$

$$n = 0, ..., N-1,$$

that is, $R_N(\cdot)$ can be considered as the trajectory of a set-valued map discrete -time dynamical system (see e.g.[5]).

Now, rewrite the differential inclusion (1) as follows:

$$\underbrace{\begin{bmatrix} \frac{\partial X}{\partial t} \\ \frac{\partial Y_1}{\partial t} \\ \frac{\partial Y_2}{\partial t} \\ \frac{\partial Z}{\partial t} \end{bmatrix}}_{=\dot{x}} \in \left\{ \underbrace{\begin{bmatrix} \alpha N X + \alpha^{'} N (Y_1 + Y_2) - (k_1 + \mu + \beta) X \\ k_1 X - \left(\mu + \beta^{'}\right) Y_1 \\ - \left(\mu + \beta^{'}\right) Y_2 \\ \beta X + \beta^{'} (Y_1 + Y_2) - \mu^{'} Z \end{bmatrix}}_{=f(x)} + \underbrace{\begin{bmatrix} -\frac{X(Y_1 + Y_2)}{X + Y_1 + Y_2} \\ 0 \\ \frac{X(Y_1 + Y_2)}{X + Y_1 + Y_2} \\ 0 \end{bmatrix}}_{=F(x)} k_2 \right\}_{k_2 \in K} \tag{9}$$

with the same initial condition (2), where $f : \mathbb{R}^4 \to \mathbb{R}^4,\ F : \mathbb{R}^4 \to \mathbb{R}^4$. We show that by the following theorem, that it is possible to obtain convergence results and estimations for the reachable set of the IVP described by (1) and (2), where $haus(\cdot, \cdot)$ denotes the Hausdorff distance between two sets.

Theorem 1. *Supposing that all trajectories of the model (9) are contained in a positive bounded set $S \subset \mathbb{R}^4$, and k_2 belongs to the real interval $K = [a, b]$. Given N, let the approximating set-valued dynamical system (8) where*

$$g(x, k_2) = f(x) + F(x) k_2 \ .$$

Then, there exists a constant C such that

$$haus(R_N(N), R(T)) \le C h^{1.5} \ .$$

In addition, for those N for which the sets $R_N(N)$ are convex

$$haus(R_N(N), R(T)) \le C h^{2} \ .$$

Proof. It is obvious that the components of f and F are differentiable and the components of the following Jacobian matrix of f and F

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \alpha N - (k_1 + \mu + \beta) & \alpha^{'} N & \alpha^{'} N & 0 \\ k_1 & -\left(\mu + \beta^{'}\right) & 0 & 0 \\ 0 & 0 & -\left(\mu + \beta^{'}\right) & 0 \\ \beta & \beta^{'} & \beta^{'} & -\mu^{'} \end{bmatrix} \tag{10}$$

$$\frac{\partial F}{\partial x} = \begin{bmatrix} -\frac{(Y_1+Y_2)^2}{(X+Y_1+Y_2)^2} & -\frac{X^2}{(X+Y_1+Y_2)^2} & -\frac{X^2}{(X+Y_1+Y_2)^2} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{(Y_1+Y_2)^2}{(X+Y_1+Y_2)^2} & \frac{X^2}{(X+Y_1+Y_2)^2} & \frac{X^2}{(X+Y_1+Y_2)^2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

are Lipschitz continuous on S.

Moreover the lie bracket structural condition assumption of the Theorem 4.1 of [5] is always fulfilled since the dimension of k_2 is equal to one.

Then all assumptions of Theorem 4.1 [5] are satisfied (or the Theorems 4.1 and 4.2 of [6]), its application gives the conclusion and both estimations follow.□

With this result, the approach described above allows us to make predictions in time for all state variables of the model of HIV/AIDS described by the differential inclusion (1) using projections of the corresponding reachable set in each of the state variable axes. Of particular interest is the resulting interval for the variable X in the model, because with this interval we get a prediction of the HIV infected persons that do not know that they are infected by HIV at time t. Also, with the corresponding prediction intervals for Y_1, Y_2 and Z obtained in the same way, we can have a suitable estimate of the number of people living with HIV at a future time with the interval resulting from the sum of predicted intervals for the state variables X, Y_1, Y_2, Z, of (1).

The practical viability of the above theorem through Differential Inclusions solvers is an actual and interesting field of research. Main dificulties in that problem are the computational complexity and solution representation.

4 Conclusions

In this paper, we formulated an HIV/AIDS model that describes the detection by Contact Tracing in Cuba as a Differential Inclusion assuming parameter k_2 as an uncertainty, because it depends on the effort displayed by the Health System in searching and locating the sexual partners that have been reported by those already detected as HIV-positive.

We have also proved that it is possible to calculate an approximation of the reachable set of our model using the set valued version of Heun's method, with estimations of order 1.5 or 2, respectively, in order to achieve reliable predictions of the size of the HIV/AIDS epidemic in Cuba that can be used to assist in successful health policy decisions.

Acknowledgments. The author wishes to thank Dr. Gonzalo Joya Caparrós for valuable remarks and discussions during the preparation of this paper.

References

1. Lounes, R., De Arazoza, H.: A two-type model for the Cuban national programme on HIV/AIDS. IMA. J. Math. Appl. Med. Biol. 16, 143–154 (1999)
2. De Arazoza, H., Lounes, R.: A non-linear model for sexually transmitted disease with contact tracing. IMA. J. Math. Appl. Med. Biol. 19, 221–234 (2002)
3. Aubin, J.-P., Cellina, A.: Differential Inclusions, Set-Valued Maps and Viability Theory. Springer, Heidelberg (1984)
4. Aubin, J.-P., Frankowska, H.: Set-Valued Analysis. Birkhäuser, Basel (1990)
5. Lempio, F., Veliov, V.: Discrete approximation of differential inclusions. Bayreuther Mathematische Schriften 54, 149–232 (1998)
6. Veliov, V.: On the time-discretization of control systems. SIAM J. Cont. Optim. 35(5), 1470–1486 (1997)
7. Atencia, M., Joya, G., García-Garaluz, E., De Arazoza, H., Sandoval, F.: Estimation of the Rate of Detection of Infected Individuals in an Epidemiological Model. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 948–955. Springer, Heidelberg (2007)

Data Mining in Complex Diseases Using Evolutionary Computation

Vanessa Aguiar, Jose A. Seoane, Ana Freire, and Cristian R. Munteanu

Facultad de Informática, Universidade da Coruña (UDC)
{vaguiar,jseoane,afreirev}@udc.es, muntisa@gmail.com

Abstract. A new algorithm is presented for finding genotype-phenotype association rules from data related to complex diseases. The algorithm was based on Genetic Algorithms, a technique of Evolutionary Computation. The algorithm was compared to several traditional data mining techniques and it was proved that it obtained similar classification scores but found more rules from the data generated artificially. In this paper it is assumed that several groups of SNPs have an impact on the predisposition to develop a complex disease like schizophrenia. It is expected to validate this in a short period of time on real data.

Keywords: Association studies, SNPs, evolutionary computation, genetic algorithms, data mining.

1 Introduction

Association studies are those in which DNA from patients that haven't developed the disease is compared to DNA from affected patients in order to find relationships between mutation in the DNA and the disease studied. Usually, mutations such as SNPs (Single Nucleotide Polymorphisms) [1] are studied. A SNP is a single nucleotide site where two (of four) different nucleotides occur in a high percentage of the population, that is, at least in 1% of the population. Since there exist 14 million of SNPs in human beings then a huge amount of data obtained from DNA genotyping needs to be dealt with, thus many variables have to be taken into account.

These studies can be done on complex diseases [2]. In these diseases, apart from the influence of genetic predisposition, environmental factors also affect, in such a way that a person could be genetically predisposed but never develop the disease. Due to the nature of these diseases, it is hard to establish a relationship between a gene and the disease since, in general, this type of disease is caused by combination of effects of several sets of SNPs which, separately, have a low effect. There is a high prevalence and impact of complex diseases like cancer, mental disorders and cardiovascular diseases.

The goal of this work is to propose an algorithm which is able to extract rules composed of sets of SNPs which represent genetic predisposition to develop the complex disease studied.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 917–924, 2009.
© Springer-Verlag Berlin Heidelberg 2009

2 State of the Art

2.1 Introduction to Genetic Algorithms

Genetic Algorithms (GAs) [3, 4] is an artificial intelligence technique. It is a search method based on Charles Darwin's Theory of Evolution [5]. As a result, Genetic Algorithms are inspired in biological evolution and genetic-molecular base. These algorithms make a population evolve through random actions similar to the actions existing in biological evolution (mutations and genetic recombination), as well as selections with a certain criteria called fitness. The fitness is used to decide which individuals are selected, i.e., the more suitable individuals are the higher likelihood they will reproduce.

2.2 Recent Work

During recently years, several techniques have been developed to analyse genotype-phenotype association studies. Most of these techniques are statistically based. However, Evolutionary Computation (EC) has proved to be a useful tool when searching for this type of associations. The Chinese Schizophrenia Consortium has carried out a study [6], using EC as a search heuristic, where the GA chromosome was coded as a list of SNPs and the fitness function was the number of hits in the sample population. Other similar approaches have been developed based on GAs, such as Jourdan's rule selection algorithm [8, 9], in which a GA-based method capable of extracting IF-THEN association rules is presented, or tree-based approaches. Shah et al [7] used a GA to perform feature selection in two different approaches in order to build decision trees and Clark et al [10, 11] used a GA to build trees that consist of boolean expressions made of blocks of SNPs. In addition, there are other approaches in which genetic programming and heuristic search are used to learn sets of classification rules capable of distinguishing between case and control subjects (Goerzel [12]).

3 Methods

An algorithm based in EC has been developed. Given that this type of technique has been designed to solve optimization problems, an application of GAs is proposed with the aim of making rule extraction possible.

3.1 Structure of the Association Rules

It is assumed that the influence of SNPs in genetic predisposition to develop a complex disease can be represented as several rules extracted from the data. These rules have the following structure:

If the following conditions are met

$$SNP1=X \text{ AND } SNP2=Y \text{ AND } \ldots \text{ AND } SNPn=Z$$

the individual is genetically predisposed to develop the complex diseased studied.

Each rule represents an individual of the GA population.

3.2 Iterative Algorithm

This algorithm (figure 1) consists of a loop that executes, in each iteration, a GA.

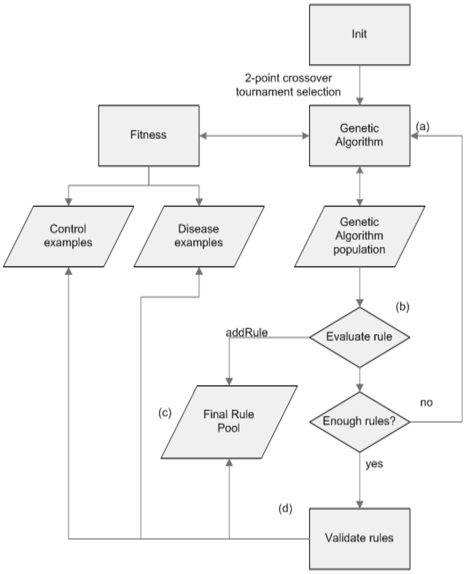


Fig. 1. Iterative algorithm structure

Every time the GA is executed, a list of candidate rules is obtained. These rules, which correspond to the final population of the GA, are filtered in order to choose the best rule between those who classify a case subject and cover at least 5% of the samples from the training set. From the iterations of the iterative algorithm a set of rules covering the search space is obtained. That set will be called rule pool.

The algorithm developed is capable of finding logical expressions. These expressions classify a subject based on the values that have the alleles of certain SNPs.

3.3 Genetic Algorithm

3.3.1 GA Individual

The chromosome of the GA, shown in figure 2, represents a candidate rule containing n SNPs plus a position for the phenotype. Each SNP needs two genes. One represents if the SNP is part, or not, of the rule (ACT) and the other one represents its real value (VALUE). This value can be: homozygous for the first allele, homozygous for the second allele, heterozygous or unknown. The last position of the chromosome, which represents the phenotype, can take 1 if the subject is a case or 0 if it is a control. Hence, the structure of an individual of the GA consists of 2n+1 genes that represent a chromosome.

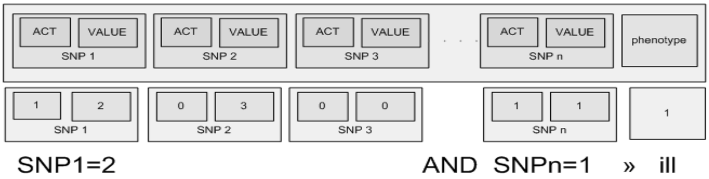


Fig. 2. A possible rule

3.3.2 Fitness Function

The fitness function used, which has to be minimized, has the following structure:

IF *the rule doesn't classify correctly any example* THEN
fitness = Number of examples of the training set + 0.1 x (Number of SNPs part of the rule / Number of SNPs in the examples)

ELSE
fitness = Number of examples incorrectly classified + (Number of examples of the training set – Number of examples correctly classified)

In the first part (the IF part) individuals are penalized adding the number of SNPs part of the rule divided by the total number of SNPs and multiplied by a coefficient to the number of examples. This addend is added in order to avoid the tendency of obtaining rules with all the SNPs. It is multiplied by 0.1 so that it won't have too much impact on the fitness value and to avoid obtaining rules with only one SNP. The value used as coefficient has been determined after a large number of experiments.

To sum up, the GA generates individuals with random values. The GA is executed until individuals change less than a certain threshold or until a maximum number of generations is reached. As explained above, individuals obtained as a result of the execution of the GA are the candidate rules then filtered by the iterative algorithm.

This approach allows obtaining different solutions, since the GA is executed several times. The aim of the algorithm presented is to find several rules that associate complex diseases to a case phenotype.

4 Generation of Artificial Input Data

In order to observe how the algorithm presented works data sets have been generated. This is done so that a pattern simulating real clinical data exists in the data created.

First of all, 360 random data sets have been generated. Each data set was divided in two groups: case subjects and control subjects. Each group represents 50% of the data set. After that, rules with two, three or four SNPs have been randomly generated. Finally, some examples representing case subjects have been modified in different percentages (between 20% and 80%). This has been done introducing the values of the SNPs of one, two or three rules generated previously. This makes it possible to verify if the algorithm is able to find the rules introduced artificially.

Four groups of sets will be available for testing the algorithm. In each group 10%, 20%, 30% or 40% of the data has been modified. Also, in each group, one third of the sets have been modified by rules of two SNPs, one third by rules of three SNPs and one third by rules of three SNPs.

5 Results

To validate the algorithm 10-fold cross validation has been used. Therefore, the original data was divided into 10 subsets. Only one subset was used as test data, while the other 9 subsets were used for training. The cross validation is repeated ten times, using each time, only, each subset. The ten results obtained after doing the 10-fold cross validation have been combined in order to obtain global classification measures.

The GA has been executed until the fitness value changed less than a certain threshold or until a maximum number of generations was reached. Deterministic tournament selection has been used, as well as a two-real-point crossover algorithm with a crossover rate of 90% and random replacement for mutation with a mutation rate of 1%.

The algorithm proposed has been compared to 29 traditional data mining techniques. These techniques can be classified in the following categories: decision rule learning, classification trees, bayesian nets and other techniques such as neural networks, metaclassifiers, regression-based techniques… The data mining suite Weka [13] was used and the 29 techniques were applied to the 360 sets created artificially.

The evolution of the percentages of correctly classified samples obtained for the different groups is shown in figure 3. Although it wasn't the aim of the algorithm proposed, it obtains better results than certain methods.

These results are similar to the mean and even better when 80% of the case subjects is modified. Classification scores are shown in table 1.

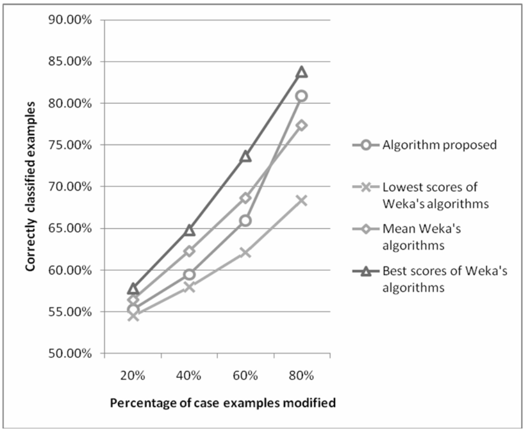


Fig. 3. Comparison of classification percentages

Table 1. Classification scores

	20%	40%	60%	80%
Algorithm proposed	55.27%	59.49%	65.94%	80.87%
Lowest scores of Weka's algorithms	54.51%	58.00%	62.14%	68.41%
Mean Weka's algorithm	56.47%	62.30%	68.66%	77.37%
Best scores of Weka's algorithms	57.84%	64.83%	73.72%	83.81%

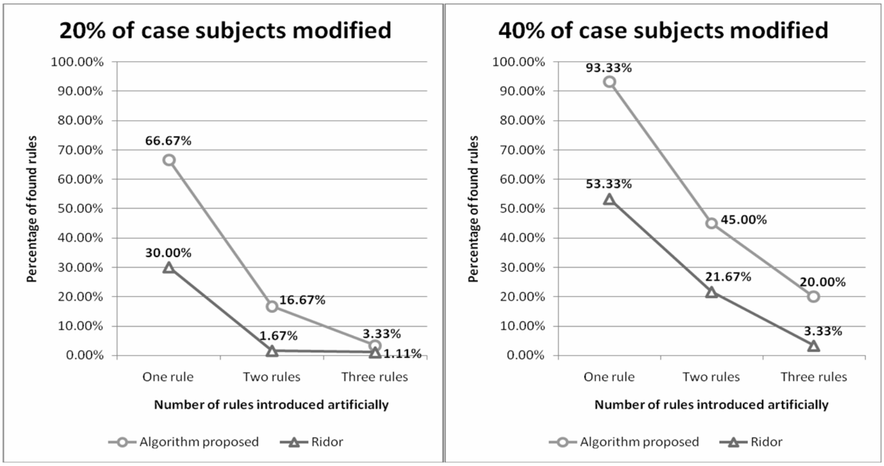


Fig. 4. Comparison of found rules

Ridor [14] is the algorithm that usually gets the best classification scores. However, when 20% or 40% of the case subjects is modified, the number of rules found by Ridor is clearly less than the number of rules found by the algorithm proposed in this paper (figure 4).

Moreover, in most of the situations where the algorithm presented doesn’t find the exact rule, it finds a part of it (for example, for a rule made of four SNPs it could find a rule with three SNPs), while Ridor, in many situations, is unable to find any of the rules introduced artificially.

6 Conclusions and Future Work

In complex diseases, the factors that increase the risk of developing a certain disease do not correspond to the values of one or more genes. There is then a likelihood of several combinations of values of different sets of SNPs affecting the increase of the risk factor. The algorithm proposed is capable of finding several sets of SNPs related to control or case subjects.

When comparing to other existing techniques, it classifies better than several of them and it was able to find a higher percentage of the rules introduced artificially.

We expect to validate the previous hypothesis with real clinical data of Galician schizophrenic patients in a near future.

Acknowledgments

This work was partially supported by the Spanish Ministry of Science and Innovation (Ref TIN2006-13274) and the European Regional Development Funds (ERDF), grant (Ref. PIO52048 and RD07/0067/0005) funded by the Carlos III Health Institute, grant (Ref. PGDIT 07TMT011CT) and (Ref. PGDIT08SIN010105PR) from the General Directorate of Research, Development and Innovation of the Xunta de Galicia and grant (2007/127 and 2007/144) from the General Directorate of Scientific and Technologic Promotion of the Galician University System of the Xunta de Galicia. The work of Vanessa Aguiar is supported by a grant from the General Directorate of Quality and Management of Galicia's University System of the Xunta. The work of José A. Seoane is supported by an Isabel Barreto grant and the work of Cristian R. Munteanu is supported by a Parga Pondal grant, both from the General Directorate of Research, Development and Innovation of the Xunta.

References

1. Den Dunnen, J.T., Antonarakis, S.E.: Mutation Nomenclature Extensions and Suggestions to Describe Complex Mutations: A Discussion. Human Mutation 15, 7–12 (2000)
2. Motulsky, A.G.: Genetics of complex diseases. Journal of Zhejiang University Science B 7(2), 167–168 (2006)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
4. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
5. Darwin, C.: On the Origin of Species by Means of Natural Selection (1859)
6. Xu, Q., Jia, Y.-B., Zhang, B.-Y., Zou, K., Tao, Y.-B., Wang, Y.-P., Qiang, B.-Q., Wu, G.-Y., Shen, Y., Ji, H.-K., Huang, Y., Sun, X.-Q., Ji, L., Li, Y.-D., Yuan, Y.-B., Shu, L., Yu, X., Shen, Y.-C., Yu, Y.-Q., Ju, G.-Z.: Association study o fan SNP combination pattern in the dopaminergic pathway in paranoia schizofrenia: a novel strategy for complex disorders. Journal of Molecular Psychiatry 9, 510–521 (2004)
7. Shah, S., Kusiak, A.: Data mining and genetic algorithm based gene/SNP selection. Artificial Intelligence in Medicine 31, 183–196 (2004)
8. Jourdan, L., Dhaenens, C., Talbi, E.-G.: A Genetic Algorithm to Exploit Genetic Data. In: Corne, D., Fogel, G. (eds.) Evolutionary Computation and Bioinformatics, pp. 297–316. Morgan Kaufmann, San Francisco (2002)
9. Jourdan, L., Dhaenens, C., Talbi, E.-G.: Rules Extraction in Linkage Disequilibrium mapping with an adaptative genetic algorithm. In: Proceedings of European Conference on Computational Biology ECCB 2003, pp. 29–32 (2003)
10. Clark, T.G., De Lorio, M., Griffiths, R.G., Farrall, M.: Finding Association in Dense Genetic Maps: A Genetics Algorithm Approach. Human Heredity 60, 97–108 (2005)
11. Clark, T.G., De Lorio, M., Griffiths, R.G.: An Evolutionary Algorithm to Find Associations in Dense Genetic Maps. IEEE Transactions on Evolutionary Computation 12(3), 297–306 (2008)

12. Goertzel, B., Coelho, L., Pennachin, C.: Identifying Potencial Biomarkers for Chronic Fatigue Syndrome via Classification Model Ensemble Mining. In: VI Intenational Conference for the Critical Assessment of Microarray Data Analysis. Durham (2006)
13. Sierra Araujo, B.: Conceptos básicos y avanzados. Aspectos prácticos utilizando el software Weka. Pearson, Prentice Hall, London (2006)
14. Gaines, B.R., Compton, P.: Induction of Ripple-Down Rules Applied to Modeling Large Databases. J. Intell. Inf. Syst. 5(3), 211–228 (1995)

Using UN/CEFACT'S Modelling Methodology (UMM) in e-Health Projects*

P. García-Sánchez, J. González, P.A. Castillo, and A. Prieto

Department of Computer Architecture and Computer Technology, CITIC-UGR,
University of Granada, Spain
`{pgarcia,jesus,pedro,aprieto}@atc.ugr.es`

Abstract. UN/CEFACT's Modelling Methodology (UMM) is a methodology created to capture the business requirements of inter-organizational business processes, regardless of the underlying technology. An example of how to apply UMM to an inter-enterprise e-health project is presented in this paper.

1 Introduction

Nowadays, Ambient Assisted Living (AAL) and e-health are in a continuous growing, so several international initiatives applying IT technologies to welfare and health are emerging [1]. One example is the I+D+i project AmIVital, which instead of focusing on home-systems and network sensors for e-health, focuses on the research and development of services that will be the basis of an interoperable platform able to communicate with different systems of e-health, e-government or others. In this sense, the B2B (Business To Business) paradigm facilitates the information exchange among organizations (social services, hospitals, government, etc.) using the Internet.

In this paper we show how to use UMM to model a complete Ambient Assisted Living scenario, describing all views and sub-views of this methodology. The paper is arranged as follows. First, the state of art, concerning all the relevant technologies and design methodologies related to B2B applications, is described in Section 2. Then, Section 3 introduces all the design views UMM uses to model a B2B system, and Section 4 describes how to apply it to model an example of an e-health use case within AmIVital, describing all its views and sub-views in relation with the proposed example. Finally, some concluding remarks and future work are pointed out in Section 5.

2 State of the Art

In the previous years, several B2B technologies have emerged, being *ebXML* and *Web Services* the most extended. Web Services have been quickly adopted by

---

* This work has been supported by the Spanish Ministry of Commerce, under project AmIVital (CENIT2007-1010).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 925–932, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the IT industry, while ebXML has gained acceptance in different B2B communities because of its specification of standard business communication scenarios. ebXML defines a set of standards that allows the enterprises negotiate their products through the Internet. It is based on a well-defined documents interchange using a contract-based approach [2], providing a specification for messaging, registry/repositories and business processes description, and unlike other approaches, it is an horizontal standard (it is not oriented towards a specific industry sector). On the contrary, Web Services expose any kind of applications to the Web, so anyone can call them (service approach). Another main difference between Web Services and ebXML is that the former is based on BPEL, which can only describe the scenario inside a company, due to it has not all the information of the services being orchestrated, while the latter can be used to model a global choreography among several companies.

Regardless of the chosen B2B framework, the business processes of the platform must be analyzed and modelled. So it is necessary to use a consistent and well-defined methodology to design a model based on a machine-readable description. *Business-Centric Methodology (BCM) for Enterprise Agility and Interoperability* [3] is a roadmap for the development and implementation of procedures to create effective, efficient, and sustainable interoperability mechanisms. It has been developed by OASIS, the same consortium that created BPEL or UDDI, among others, and it is complementary to other existent architectures and technologies designed to build business oriented services, like ebXML or Web Services. BCM is formed by a set of model layers with a step-guide process, and an information pyramid to align the semantic information of partners. This allows the participation of business experts and the creation of a very large documentation repository. Nevertheless, this methodology has some disadvantages: it has a very large learning curve and it is not very extended yet.

SOMA [4] is an architecture proposed by IBM to model service oriented processes. It lets the identification, specification and implementation of services inside the SOA paradigm. To achieve this tasks, it proposes a top-down modelling oriented to intra-enterprise services (service-oriented instead of business-oriented).

UN/CEFACTs Modelling Methodology (UMM) [5] is an approach to model the business services that each partner must provide in order to perform a B2B collaboration. It has a complete meta-model about business processes and business information, including a process analysis methodology. It is interesting to show that UMM provides and supports components to capture the knowledge about the business processes, and that it is independent of the underlying implementation technology (ebXML, Web Services, CORBA or EDI). Furthermore, because UMM extends UML, we could say that this methodology is more easily adaptable, due to the high development, acceptance and maturity of UML. In fact, a survey of B2B modelling languages show that UMM is the most complete approach [6], thus, this is the methodology the authors have chosen to model the AAL scenario presented in this paper.

3 Usage of UMM

To build a complete B2B information system it is necessary to obtain all the business knowledge, as well as all the technical information about the business domain to be modelled, being this information the basis of the system ambit, its requirements and its specifications. UMM proposes the usage of worksheets to do this work, being them the bridge between business experts and business analysts. Thus, every view and sub-view of the model requires a set of these worksheets in order to obtain the relevant information. The main views of this methodology are:

Business Domain View: It is used to discover, but not to build, collaborative business processes. Using interviews with the business experts, together with the worksheets, the process analyst is guided in order to capture business knowledge.

Business Requirements View: It is used to identify the business processes and business entities candidates to perform a business collaboration. First the workflow processes are modelled, then the relevant *Business Entities* and their lifecycles are identified, and finally the requirements of every transaction are described.

Business Transaction View: This view describes the whole choreography, the interactions and the exchanged information between business partners.

Business Service View: This last view details the composed services, the agents, and the necessary data interchange in order to execute and validate a business collaboration. This view is expressed in the language and technical concepts chosen by the software developers (i.e. BPEL or BPSS), and thus, it is not included in the UMM guidelines.

At the moment, the most viable option to apply UMM is using the Enterprise Architect tool together with the UMM Add-In [7]. This plug-in is widely tested, includes a whole user guide, and is completely free. It also includes tools to manage the worksheets, which are not standardized by the UN/CEFACT. But its most important features are that it is able to validate the model, and that it can generate BPEL or BPSS descriptions from a valid model.

4 Using UMM in AmIVital

The general objective of project AmIVital is to develop a new generation of technology and information tools to model, design and implement an Ambient Intelligence (AmI) platform to support the independent life, welfare and health of its users. As this objective is completely integrated in the European concept of Ambient Assisted Living (AAL), the Spanish Ministry of Commerce, inside the CENIT program, has awarded a grant of more than 20 million euro to this project.

In a large e-health project like AmIVital, where several institutions must create functional services to be choreographed, and many different technologies must be integrated, it is a good idea to use an independent-technology based modelling methodology like UMM. As an example of its application we could think in the following scenario: A virtual calendar launches an event and notifies the monitorization system to create a reminder for the user, sending him a notice. If the user is not localizable, or he does not accept the notice, the monitorization ends. Otherwise, the monitorization is accepted and a monitorization context, containing information about the devices and instructions about how to use them, is created. Then the user activates the monitorization, which sends the data to the monitorization context, where they are checked. Finally a report is sent to the user and the doctor (or an alarm if there have been any problem).

In the next subsections all the UMM views of this example are described.

4.1 Business Domain View

This view allows to split the business in two ways: *Business Areas* (divisions of an enterprise: sales, personnel,...) that encompass *Business Process Areas* (common operations) or *Business Categories*, that may encompass other *Business Categories*. As the last one is a more general modelling strategy, it is the proposed in this work to model the AmIVital platform.

Inside every leaf of the created structure a use case diagram is created to identify the relevant business partners and business processes. As each process may be composed by other processes, we may have to use the *include* association. Business partners must also be added and linked with their participating processes with *participates* relations. Finally, stakeholders can also be added with an *isInterestedOf* relation to describe that, although they are interested in a business process, they do not take active part in its execution.

4.2 Business Requirements View

The requirements of the collaborative business processes, business entities and transactions, are collected in this view. It is composed by the following sub-views:

Business Process View: Every business process can be divided into subprocesses, and each of these, into activities. For example, the previous whole scenario has been divided into three sub-processes: *Create reminder*, *Create monitorization context* and *Perform monitorization*. Every sub-process is modelled from each participant's point of view, and finally, a collaborative point of view is generated (an example of this view can be seen in Figure 1).

Business Entity View: This sub-view describes the relevant *Business Entities* affected by a business process and models their lifecycle. A Business Entity is a real-world thing with significance to the partners. In our example we define entities like Reminder, Monitorization or Localization. Each one of them can change its state during their lifecycle (an example is shown in Figure 2a). This information is also detailed in the worksheets.

Transaction Requirements View: This step focuses on the atomic interactions between the partners of a collaboration flow, describing the exchange of

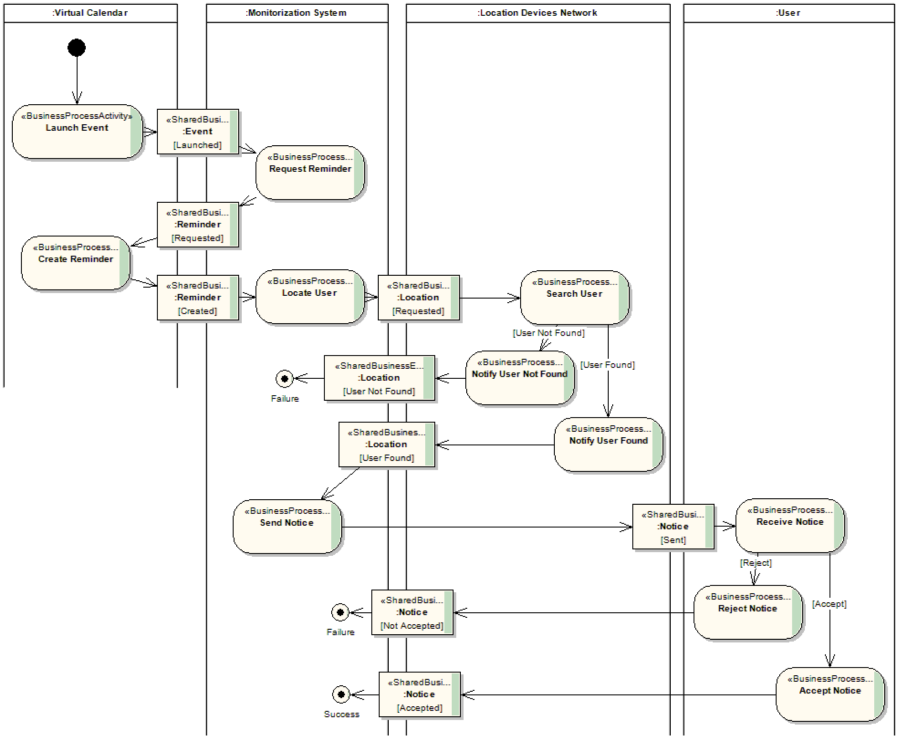
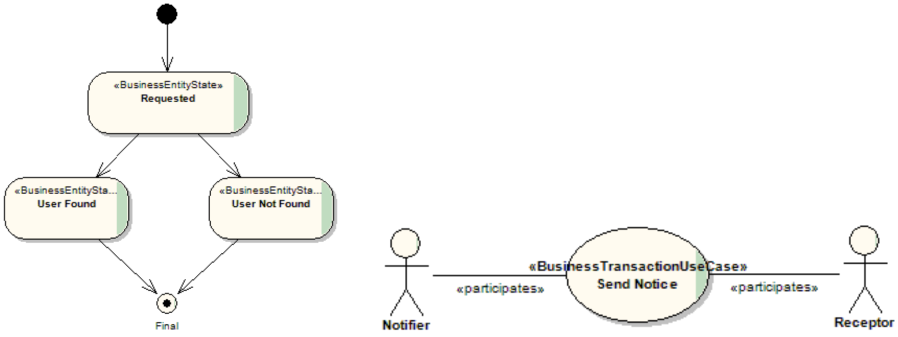

:Virtual Calendar
:Monitorization System
:Location Devices Network
:User
«BusinessProcessActivity» Launch Event
«SharedBusi... :Event [Launched]
«BusinessProcess... Request Reminder
«SharedBusi... :Reminder [Requested]
«BusinessProcess... Create Reminder
«SharedBusi... :Reminder [Created]
«BusinessProcess... Locate User
«SharedBusi... :Location [Requested]
«BusinessProcess... Search User
[User Not Found]
[User Found]
«BusinessProcess... Notify User Not Found
«SharedBusinessE... :Location [User Not Found]
Failure
«BusinessProcess... Notify User Found
«SharedBusines... :Location [User Found]
«BusinessProcess... Send Notice
«SharedBusi... :Notice [Sent]
«BusinessProcess... Receive Notice
[Reject]
[Accept]
«BusinessProcess... Reject Notice
«SharedBusi... :Notice [Not Accepted]
Failure
«BusinessProcess... Accept Notice
«SharedBusi... :Notice [Accepted]
Success

Fig. 1. Business Process View of the sub-process *Manage Reminder* of our example (collaborative view). Each column is a partition to describe the behaviour of each partner.


«BusinessEntityState» Requested
«BusinessEntitySta... User Found
«BusinessEntitySta... User Not Found
Final
Notifier
«participates»
«BusinessTransactionUseCase» Send Notice
«participates»
Receptor

(a) (b)

Fig. 2. (a) Business Information Entity Lifecycle. (b) Business Transaction Use Case.

information between exactly two roles. The worksheets describe who starts the transaction, the activities performed, the pre and post-conditions, and the possible exceptions. An example can be seen in Figure 2b.

Business Collaboration Requirements View: All the participants and requirements of each collaboration are detailed in this sub-view, so every collaboration is divided into Business Transactions Uses Cases (created on the previous view) and business partners are mapped to the roles in every transaction (for example, the Business Partners *User* and *Monitorization System* are respectively mapped to the *Receptor* and the *Notifier* roles in the transaction use case shown in Figure 2b). Worksheets are used to specify which Business Entities and Business Partners participate in the collaboration, along with all the pre and post-conditions, exceptions and performed actions.

Business Collaboration Realization View: This last sub-view is used to map the business partners defined in the *Business Domain View* into the business collaborations defined in the Business Collaboration Requirements View. This view allows that different partners perform the same collaboration, increasing the re-usability and scalability of the design.

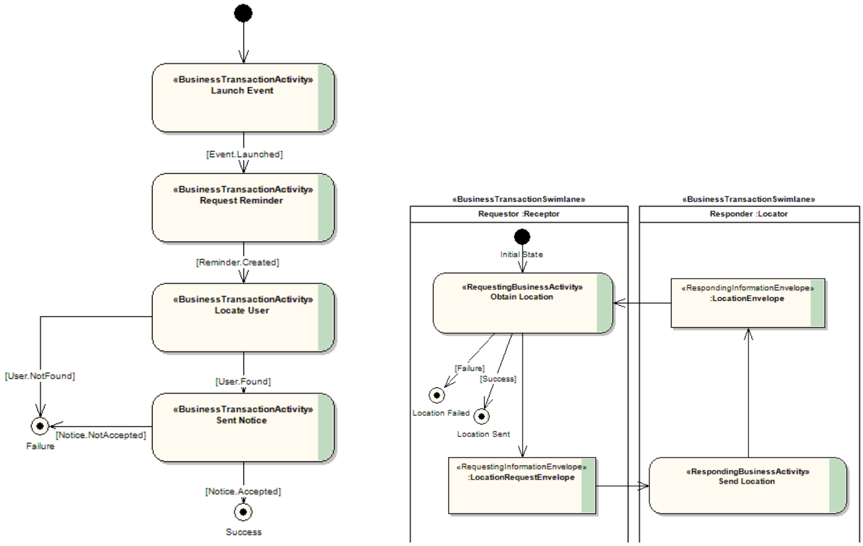
4.3 Business Transaction View

This last view in the UMM workflow is used to model the business collaborations, from the analysts's point of view, in order to reach the requirements defined in the previous steps. It describes the whole choreography of interactions and defines the information that is exchanged among these interactions. It is composed by the following sub-views:

Business Choreography View: For each one of the Business Collaborations defined in the Business Collaboration Requirements View, a Business Choreography is created. This choreography is composed by Business Activities, being each one of them linked to the business transactions defined in the *Business Interaction View* (next sub-view). Constraints are used in order to guide this execution flow, as can be seen in Figure 3a).

Business Interaction View: This view details the sequence of steps that each business transaction performs, identifying the information exchanged between exactly two roles. These exchanges are modelled as requests and optional responses. It is interesting to show that these Business Transactions are also described in worksheets that specify the transaction pattern (Request/Response, Query/Response, Notification...) and several legal and security requisites. For example, Figure 3b shows a request/response transaction to locate the user.

Business Information View: This view models the information entities that are exchanged in each transaction. These information entities are encapsulated in *information envelopes*, which have a *header* and a *body*. The information entities are modelled using UML artefacts. In fact, there are not restrictions to model this documents, although UN/CEFACT recommends using the Core Components Technical Specification (CCTS) [8].



(a) (b)

Fig. 3. (a) Business Choreography for the sub-process *Manage Reminder*. Every transaction is refined within a Business Interaction View. One of these transactions can be seen in (b).

Nevertheless, as CCTS is mainly focused on commercial transactions, it is not appropiated to model the information entities of AmIVital.

4.4 Business Service View

Functional services for e-health are modelled more efficiently using SOA, mainly because this kind of services (location, vital signs monitoring, video-conference, etc.) are more centred on the user rather than on the business process. Thus, as UMM does not include a modelling guide to create or reuse functional services, we propose the use of SOMA to model different low-coupling services for each organization involved in AmIVital.

Like UMM, SOMA is also based on a top-down modelling, being more oriented towards intra-enterprise services. SOMA can also be used to model a micro-soa, for instance, to describe the set of domotic devices inside a house, providing a level of abstraction and a service interface more adequate to integrate them in the whole platform using UMM.

5 Conclusions and Future Work

Within a wide inter-organizational research project like AmIVital, where several enterprises and universities work together to develop services for e-health, a common modelling methodology is mandatory. As a proof of concept, UMM has been used to model a simple e-health business process, where several systems collaborate in an intelligent way with the user[1]. We have chosen UMM due to its growing usage in the development of B2B applications, the fact that its learning curve is not very large, and because there exist completely functional tools to create and validate the models.

During the development of this work, the UMM authors have published the 2.0 specification of this methodology [9], with some changes in the artefacts and packet structure to make UMM more simple, extensible and scalable. At the moment it is only a public draft version. Their authors are also working in a new UMM Add-In for Enterprise Architect, being this add-in in beta phase.

The next step in our project will be to migrate the use case described in this paper to UMM 2.0, and to continue working with this new version of the methodology, extending its usage to other e-health services, as telecare, medication management and domotic services among others.

References

1. Kleinberger, T., Becker, M., Ras, E., Holzinger, A., Müller, P.: Ambient intelligence in assisted living: Enable elderly people to handle future interfaces. In: Stephanidis, C. (ed.) UAHCI 2007 (Part II). LNCS, vol. 4555, pp. 103–112. Springer, Heidelberg (2007)
2. Patil, S., Newcomer, E.: ebXML and Web Services. IEEE Internet Computing 7(3), 74–82 (2003)
3. OASIS BCM TC: Business-Centric Methodology for Enterprise Agility and Interoperability. Executive White Paper (2003), `http://businesscentricmethodology.com/`
4. Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., Holley, K.: SOMA: A method for developing service-oriented solutions. IBM Systems Journal 47(3), 377–396 (2008)
5. Hofreiter, B., Huemer, C., Liegl, P., Schuster, R., Zapletal, M.: UN/CEFACT'S modeling methodology (UMM): A UML profile for B2B e-commerce. In: Roddick, J.F., Benjamins, V.R., Si-said Cherfi, S., Chiang, R., Claramunt, C., Elmasri, R.A., Grandi, F., Han, H., Hepp, M., Lytras, M.D., Mišić, V.B., Poels, G., Song, I.-Y., Trujillo, J., Vangenot, C. (eds.) ER Workshops 2006. LNCS, vol. 4231, pp. 19–31. Springer, Heidelberg (2006)
6. Folmer, E., Bastiaans, J.: Methods for Design of Semantic Message-Based B2B Interaction Standards. Volume Enterprise Interoperability III (2008)
7. Hofreiter, B., Huemer, C., Liegl, P., Schuster, R., Zapletal, M.: UMM add-in: A UML extension for UN/CEFACT's modeling methodology. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 618–619. Springer, Heidelberg (2007)
8. United Nations Centre for Trade Facilitation and Electronic Business: Core Components Technical Specification (2003), `http://www.unece.org/cefact/`
9. Huemer, C., Liegl, P., Motal, T., Schuster, R., Zapletal, M.: The development process of the UN/CEFACT modeling methodology. In: ICEC 2008: Proceedings of the 10th international conference on Electronic commerce, pp. 1–10. ACM, New York (2008)

---

[1] Available at `http://tinyurl.com/aa2bf8`

Matrix Metric Adaptation Linear Discriminant Analysis of Biomedical Data

M. Strickert[1,⋆], J. Keilwagen[1], F.-M. Schleif[2], T. Villmann[3], and M. Biehl[4]

[1] Research Group Data Inspection, IPK Gatersleben, Germany
`stricker@ipk-gatersleben.de`
[2] Research Group Computational Intelligence, University of Leipzig, Germany
[3] Dept. of Computer Science, University of Applied Sciences Mittweida, Germany
[4] Intelligent Systems Group, University of Groningen, The Netherlands

Abstract. A structurally simple, yet powerful, formalism is presented for adapting attribute combinations in high-dimensional data, given categorical data class labels. The rank-1 Mahalanobis distance is optimized in a way that maximizes between-class variability while minimizing within-class variability. This optimization target has resemblance to Fisher's linear discriminant analysis (LDA), but the proposed formulation is more general and yields improved class separation, which is demonstrated for spectrum data and gene expression data.

Keywords: adaptive Mahalanobis distance, LDA.

1 Introduction

Proper characterization of attribute dependence in high-dimensional data is a key issue for the analysis of data from high-throughput screening technologies like gene expression arrays and spectrometers. By the experimental design often treatment and control cases of biological studies are available, and explanatory models are sought helping to separate these cases at data level. For that purpose, test statistics like the t-test or U-test are commonly used for the identification of potentially interesting genes, molecular masses, or outstanding experiments, for example. Alternatively, algorithms for variable ranking and feature subset selection can be used [5,9]. Typical challenges related to the analysis of high-dimensional data sets from biochemistry and medicine concern the exponential number of tested attribute combinations which make greedy optimization or other heuristics and approximations such as genetic algorithms necessary for splitting attributes properly in decision trees or for attribute-tuned k-nearest neighbor classification [15].

Feature weighting, rather than in-depth ordering, can be approached by parallelized methods. Supervised metric adaptation approaches have made strong progress in the last couple of years [17,19], and also generalized learning vector quantizers were shown to profit from adaptive metrics [6,16]. More recently,

---

[⋆] Corresponding author.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 933–940, 2009.
© Springer-Verlag Berlin Heidelberg 2009

matrix-based adaptive metric have been investigated along that line in order to extend single attribute rating to assessing the influence of pairs of attributes on the classification accuracy [14,18]. Traditionally, linear discriminant analysis (LDA) is used for a structurally related task of extracting class-separating directions in multi-class data sets [10]. LDA features intuitive interpretation of linear class separation and allows attribute-reducing data projections without the need for creating an explicit classifier. We present a new approach based on adaptive Mahalanobis distance (AMDi) preserving these beneficial features while removing a structural problem of classical LDA and enhancing the separation capabilities. Faithful data projections and attribute dependence graphs can be obtained, as shown for a spectrum data set and a gene expression data set.

2 Method

The underlying idea of AMDi is related to LDA in which within-class variances get minimized, while between-class variances get maximized. LDA is thus a good starting point for which an extended view is derived for the rank-1 Mahalanobis distance. For C different classes, the LDA-optimal direction $\boldsymbol{v}$ in the space of data items $\mathbf{X} \subset \mathbb{R}^M$, $|\mathbf{X}| = N$, with class-specific sub-sets $(\mathbf{X}_1, \ldots, \mathbf{X}_C)$ is obtained by maximizing the LDA stress function [8]:

$$S_{LDA} = \frac{\boldsymbol{v}^\top \cdot \left[\sum_{i=1}^{C} N_i \cdot (\boldsymbol{\mu_i} - \boldsymbol{\mu}) \cdot (\boldsymbol{\mu_i} - \boldsymbol{\mu})^\top\right] \cdot \boldsymbol{v}}{\boldsymbol{v}^\top \cdot \left[\sum_{i=1}^{C} \boldsymbol{\Sigma}_i\right] \cdot \boldsymbol{v}} \,. \tag{1}$$

Therein, $N_i = |\{\mathbf{x}^j : \mathrm{c}(j) = i\}|$ is the size of the data set of the i-th class, $\boldsymbol{\mu_i} := 1/N_i \cdot \sum_{k \in [1,N]: c(k)=i} \mathbf{x}^k$ is the center of class i, $\boldsymbol{\mu} := 1/N \cdot \sum_{k=1}^{N} \mathbf{x}^k$ is the center of the complete data set, and $\boldsymbol{\Sigma}_i := \sum_{j:\mathrm{c}(j)=i} (\mathbf{x}^j - \boldsymbol{\mu_i}) \cdot (\mathbf{x}^j - \boldsymbol{\mu_i})^\top$ denotes the covariance matrix of class i. The numerator contains the between-class variation as squared difference between class centers, while the denominator describes the within-class variation over all classes i. In case of more than one optimized direction, $\boldsymbol{v}$ becomes a matrix $[\boldsymbol{v}^1, \ldots, \boldsymbol{v}^k]$ that leads to algebraically ill-defined ratio in Eqn. 1, making a lossy mapping of numerator and denominator to matrix determinants necessary [8]. For simplicity, we stick to single directions here.

Mahalanobis distance describes attribute dependences originally by their inverse covariance values. Replacing these covariances by adaptive parameters helps realizing flexible data transformations for emphasizing class-specific data distributions. The outer self-product of a parameter vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_M)^\top$ can be used to create an adaptive matrix $\boldsymbol{\Lambda} = \boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^\top$ of rank one. Such a positive-definite matrix is used as adaptive constituent of a rank-1 Mahalanobis metric:

$$\mathrm{d}^v(\boldsymbol{x}, \boldsymbol{y} | \boldsymbol{\lambda}) = (\boldsymbol{x} - \boldsymbol{y})^\top \cdot \boldsymbol{\Lambda} \cdot (\boldsymbol{x} - \boldsymbol{y}) = \left(\sum_{m=1}^{M} \lambda_m \cdot (x_m - y_m)\right)^2 \,. \tag{2}$$

This distance is an appealingly simple case, because the underlying model has, like LDA, only as many parameters as the data dimensionality is, while being able to express the most interesting scaling direction. Thereby, no constraints need to be put upon the parameters for minimizing the proposed stress function

$$S_{\mathrm{d}^v} = \frac{W\left(\mathbf{X}|\mathrm{d}^v\right)}{B\left(\mathbf{X}|\mathrm{d}^v\right)} \tag{3}$$

with

$$W\left(\mathbf{X}|\mathrm{d}^v\right) = \sum_{k=1}^{C} \sum_{n_1=1}^{N_k-1} \sum_{n_2=n_1}^{N_k} \mathrm{d}^v\left(\boldsymbol{x}_{k,n_1}, \boldsymbol{x}_{k,n_2}|\boldsymbol{\lambda}\right) ,$$

$$B\left(\mathbf{X}|\mathrm{d}^v\right) = \sum_{k_1=1}^{C-1} \sum_{k_2=k_1+1}^{C} \sum_{n_1=1}^{N_{k_1}} \sum_{n_2=1}^{N_{k_2}} \mathrm{d}^v\left(\boldsymbol{x}_{k_1,n_1}, \boldsymbol{x}_{k_2,n_2}|\boldsymbol{\lambda}\right) .$$

This ratio involves within-class compression $W\left(\mathbf{X}|\mathrm{d}^v\right)$ and between-class expansion $B\left(\mathbf{X}|\mathrm{d}^v\right)$. Like vector $\boldsymbol{v}$ in LDA, $\boldsymbol{\lambda}$ can be interpreted as projection vector useful for mapping data vectors to scalar values by calculating $\boldsymbol{x}^{\mathsf{T}} \cdot \boldsymbol{\lambda}$. In contrast to the LDA criterion in Eqn. 1 the stress function in Eqn. 3 remains valid also for multiple direction vectors $[\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^k]$, because Eqn. 2 always ensures scalar distance values. Plainly speaking, LDA optimizes the projection, while AMDi rescales the original data space.

Despite the simplicity of the rank-1 adaptive Mahalanobis distance, the minimization of the stress function in Eqn. 3 is much harder than the matrix-algebra solution of LDA. For getting reliable optimization results, we make use of the memory-limited quasi-Newton method [12]. For such an optimization approach we need to supply the derivative of Eqn. 2:

$$\frac{\partial \mathrm{d}^v\left(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\lambda}\right)}{\partial \lambda_i} = 2 \cdot (x_i - y_i) \cdot \sum_{m=1}^{M} \lambda_m \cdot (x_m - y_m) . \tag{4}$$

Throughout this work, optimization is stopped, if the improvement of subsequent stress function evaluations falls below 10^{-10}.

The Mahalanobis distance in Eqn. 2 can be separated into two parts, the difference vector $(\boldsymbol{x} - \boldsymbol{y})^{\mathsf{T}}$ and a projection of this vector $\boldsymbol{\Lambda} \cdot (\boldsymbol{x} - \boldsymbol{y})$. If the original data vectors $\boldsymbol{x}, \boldsymbol{y}$ are very different, although they are of the same class, then their difference vector and its projection should be orthogonal for damping this undesired discrepancy between class membership and closeness of data points. For unequal classes, the difference should be more emphasized by a collinear amplification of both difference vectors. Thus, proper class assignment is driven by the parameters $\boldsymbol{\lambda}$ of the distance creating an auxiliary projection space $\boldsymbol{\Lambda} \cdot \boldsymbol{x}$ for each data vector.

3 Applications

AMDi-based attribute rating is illustrated in two biomedical tasks targeting the extraction of interesting channel combinations from a spectrum data set and for target gene pairs from cDNA arrays.

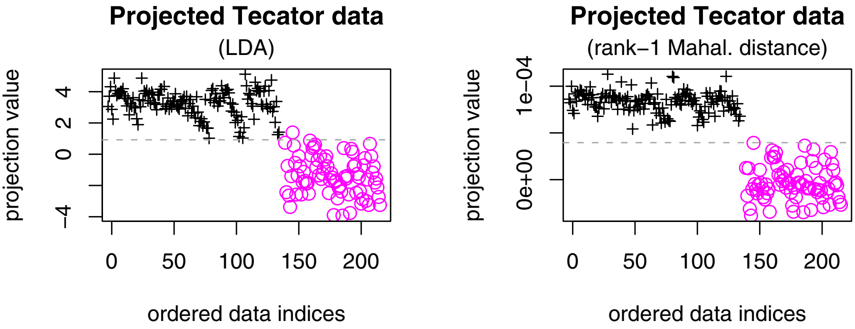


Fig. 1. Comparison of projections of 100-dimensional Tecator spectra to the ordinate, plotted vs. data index. Left, projection to LDA direction. Right, projection to optimized rank-1 Mahalanobis vector $\boldsymbol{\lambda}$. Crosses denote low fat content and circles high fat content. Horizontal dashed lines indicate the midpoints of the projected class averages.

3.1 Tecator Benchmark Data

Initial experiments are done with a well-known benchmark data set containing 215 samples of 100-dimensional infrared absorption spectra recorded on a Tecator Infratec Food and Feed Analyzer working in the wavelength range 850–1050nm by the Near Infrared Transmission (NIT) principle [1]. The original regression problem was changed into a classification task of low and high fat content in meat probes according to [13] for feature identification [16]. The task is to distinguish between highly correlated spectra corresponding to fat content, being complicated by the strong spectral overlap [13].

The optimized AMDi distance vector for class separation is used for the data projection, shown in the right panel of Fig. 1. For comparison, the LDA projection, obtained by the R package MASS, is shown in the left panel. Although the LDA separation is quite good, it fails to generate a one-dimensional projection without overlapping classes on the training set. In contrast, AMDi achieves perfect separation along the ordinate, yielding a powerful and simple classifier. This observation is also supported by the area under precision-recall curve ($\mathrm{AUC_{PR}}$) [2] which is 0.9991 for LDA and at its maximum of 1.0 for AMDi.

These results demonstrate the principle use of the new method for a well-known benchmark set. In the following case study, we investigate the generalization performance on a more complex and interesting biological data set.

3.2 Gene Expression Data

The leukemia gene expression data set [4] studied in the following has become a quasi-benchmark for testing feature selection methods. The original research aimed at the identification of the most informative genes for modeling and classification of two cancer types, acute lymphoblastic leukemia (ALL) and acute

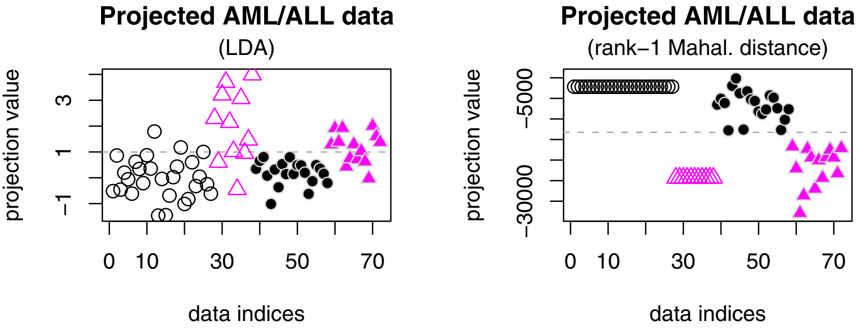


Fig. 2. Comparison of projections of the AML/ALL data set. Left, projection of cancer gene expression data (7129-dimensional) to the LDA direction $\boldsymbol{v}$. Right, projection to the best rank-1 Mahalanobis vector $\boldsymbol{\lambda}$. Horizontal dashed lines indicate the midpoints of the projected class averages. Both approaches were trained on the test set consisting of 27 ALL cases (open circles) and 11 AML cases (open triangles) (first 38 indices). The unseen cases (20 ALL, filled circles; 14 AML, filled triangles) can be classified by a midpoint threshold (dashed horizontal line) between the projected points of the trained classes.

myeloid leukemia (AML). The public data set covers 7129 genes and comes with pre-defined splits into training and test samples: the training data contains 27 cases of ALL and 11 cases of AML, whereas the test data set contains 20 cases of ALL and 14 cases of AML. Complementary to existing studies on this cancer data set [4,16], the adaptive rank-1 Mahalanobis distance allows to extract relevances of dependent gene pairs.

Parameter optimization on the training data set by AMDi leads to unique and stable results. Using the test data set, we observe a good generalization performance, as shown in the right panel of Fig. 2. A canonical thresholding by the midpoint of the projected class averages of the training data yields a perfect classification for the training set and also for the test set. By visual comparison, the LDA result in the left panel of Fig. 2 shows inferior class separation to AMDi on both training and test sets. These findings are also supported quantitatively. We find good classification performance in terms of $\mathrm{AUC_{PR}}$ for training and test data set for LDA of 0.852 and 0.898, respectively, and of 1.0 in both cases for AMDi. The increased test performance of LDA over the training might not be obvious from the left panel of Fig. 2, because this plot is visually dominated by the projection training set. The $\mathrm{AUC_{PR}}$ values though are calculated separately for both sets related to open and closed glyphs. Especially for imbalanced data sets the advantage of $\mathrm{AUC_{PR}}$ over other related measures, such as the area under ROC curve, has been described recently [2]. Considering the class separation illustrated by Fig. 2 and the $\mathrm{AUC_{PR}}$ measure, AMDi possesses valuable generalization capabilities despite of the high data dimensionality.

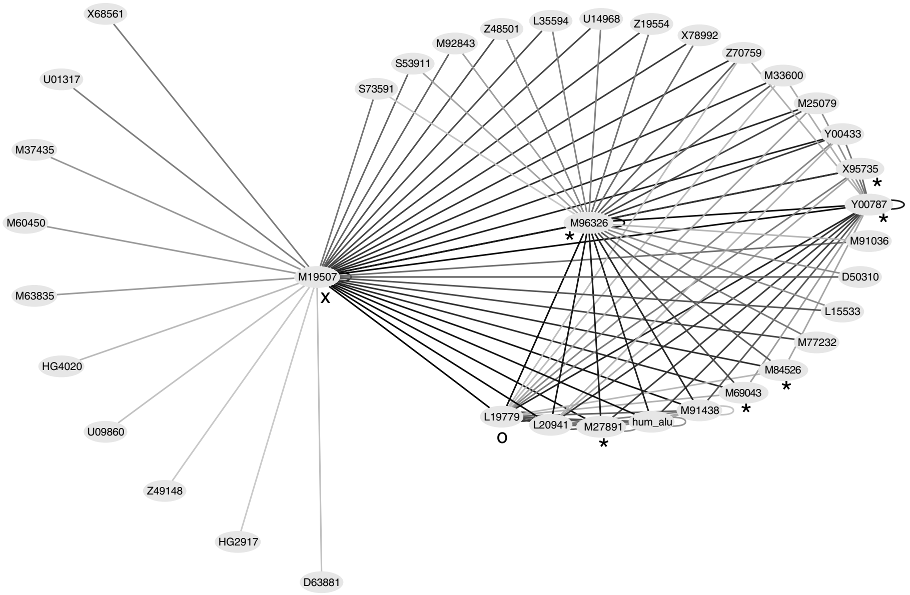


Fig. 3. Gene pair dependency network derived from the top 100 magnitude values in the 7129x7129 rank-1 Mahalanobis matrix $\boldsymbol{\Lambda}$. The network contains 37 nodes corresponding to 37 genes and 100 edges corresponding to the top 100 entries in $\boldsymbol{\Lambda}$. The darker an edge connection, the stronger the gene association. Genes with self-connections possess solitary separating quality. Extra marked genes refer to publications: (∗):[4], (x):[7], (o):[11].

Comparing the good class separation of AMDi to a projection by identical weights (not shown) or to LDA projection indicates that the parameter vector $\boldsymbol{\lambda}$ contains relevant differential information about the two classes. This information can be assessed by the corresponding matrix $\boldsymbol{\Lambda}$. Instead of plotting the cumbersome 7129x7129 matrix, we use an illustration of its 100 entries with largest magnitudes. These pairs of genes are illustrated as network in Fig. 3. This plot only scratches the very surface of a wealth of yet unvisited interactions coded in the matrix $\boldsymbol{\Lambda}$. It contains some of the candidates of Golub et., namely M96326 (26 edges), Y00787 (15 edges), M27891 (6 edges), M69043 (4 edges), M84526 (4 edges), and X95735 (4 edges). In addition to these genes, a number of other important players could be identified: particularly M19507 (36 edges) was not mentioned in their study, but it has been identified recently as interesting gene encoding myeloperoxidase (MPO), an enzyme known to be strongly involved in regulation and termination of inflammatory processes [7]. Similarly, L19779 (13 edges), which is related to the production of H2A proteins and to nucleosome structure, was not mentioned in the original study, but it has been confirmed in a decision tree approach [11]. A number of further interesting genes could be found, as for instance gene L20941 (7 edges) which has close homology to ferritin H and is a well-known cancer gene [3].

This case study indicates two ways how matrix learning assists attribute relevance assessment. On one hand, the absolute value of the matrix entry indicates the importance of a specific attribute pair; thereby, self-connections represented by entries on the diagonal denote the subset of individual attribute weights. On the other hand, the number of specific attribute combinations provides an impression of its global involvement, this way, helping to assess its potential role as a network hub.

4 Conclusions

A mathematical framework has been presented for data-driven pairwise attribute weighting based on supervised adaptation of a rank-1 formulation of Mahalanobis distance with its weight matrix expressed by an outer product of adaptive vectors. Data clouds belonging to the same class category get collapsed while class-separating contrast is enhanced.

We regard AMDi with its adaptive Mahalanobis distance as interesting alternative to Fisher's LDA with its inverse covariance matrix. Both are based on linear projection models, but AMDi shows better qualitative and quantitative class separation results than LDA. As demonstrated for spectrum and gene expression data, the amount of improvement depends on the data complexity. The stress function of AMDi is structurally different from the LDA optimization criterion: while LDA ratio requires a special treatment for more than one direction, this is not needed for AMDi which utilizes (scalar) sums of distances. The minimization of the new stress function involves non-linear optimization and is therefore computationally more demanding than classical LDA. For the one-dimensional case, projections of labeled data get well enhanced by AMDi both for plain visual inspection and for the area under precision-recall curve measure of classification performance. The resulting candidate genes of the biological cancer study are found to be in good accordance with a several publications available on that topic.

Non-trivial adaptive full-rank Mahalanobis distances are of interest in future research, because these offer more comprehensive perspectives on complex attribute dependency analysis, crucial in the domain of systems biology and other scientific fields.

Acknowledgments

The work is supported by grant XP3624HP/0606T, Ministry of Culture Saxony-Anhalt.

References

1. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
2. Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: ICML 2006: Proceedings of the 23rd international conference on machine learning, pp. 233–240. ACM, New York (2006)
3. Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16(10), 906–914 (2000)
4. 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 286(5439), 531–537 (1999)
5. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction: Foundations and Applications. Springer, Berlin (2006)
6. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Networks 15, 1059–1068 (2002)
7. Hu, S., Rao, J.: Statistical redundancy testing for improved gene selection in cancer classification using microarray data. Cancer Informatics 2, 29–41 (2007)
8. Lee, E., Cook, D., Klinke, S., Lumley, T.: Projection pursuit for exploratory supervised classification. Journal of Computational and Graphical Statistics 14(4), 831–846 (2005)
9. Massachusetts Institute of Technology. Journal of Machine Learning Research – Special Issue on Variable and Feature Selection, vol. 3 (2003)
10. McLachlan, G.: Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience, Hoboken (2004)
11. Molla, M., Waddell, M., Page, D., Shavlik, J.: Using machine learning to design and interpret gene-expression microarrays. AI Magazine 25(1), 23–44 (2004)
12. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Heidelberg (1999)
13. Rossi, F., Villa, N.: Support vector machine for functional data classification. Neurocomputing 69(7–9), 730–742 (2006)
14. Schneider, P., Biehl, M., Hammer, B.: Matrix adaptation in discriminative vector quantization. IfI Technical Report Online IFI-08-08, Clausthal University of Technology (2008)
15. Soto, A.J., Cecchini, R.L., Vazquez, G.E., Ponzoni, I.: A wrapper-based feature selection method for admet prediction using evolutionary computing. In: Marchiori, E., Moore, J.H. (eds.) EvoBIO 2008. LNCS, vol. 4973, pp. 188–199. Springer, Heidelberg (2008)
16. Strickert, M., Seiffert, U., Sreenivasulu, N., Weschke, W., Villmann, T., Hammer, B.: Generalized relevance LVQ (GRLVQ) with correlation measures for gene expression data. Neurocomputing 69, 651–659 (2006)
17. Sun, Y.: Iterative relief for feature weighting: Algorithms, theories, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 1035–1051 (2007)
18. Weinberger, K., Saul, L.: Fast solvers and efficient implementations for distance metric learning. In: McCallum, A., Roweis, S. (eds.) Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008), pp. 1160–1167. Omni Press (2008)
19. Yang, L.: Distance Metric Learning: A Comprehensive Survey. Report, Michigan State University (2006), http://www.cse.msu.edu/~yangliu1/

SPECT Image Classification Techniques for Computer Aided Diagnosis of the Alzheimer Disease

J. Ramírez, R. Chaves, J.M. Górriz, M. López,
D. Salas-Gonzalez, I. Álvarez, and F. Segovia

Dept. of Signal Theory, Networking and Communications
University of Granada, Spain

Abstract. Alzheimer disease (AD) is a progressive neurodegenerative disorder first affecting memory functions and then gradually affecting all cognitive functions with behavioral impairments. As the number of AD patients has increased, early diagnosis has received more attention for both social and medical reasons. Currently, accuracy in the early AD diagnosis is below 70% so that AD does not receive a suitable treatment. Functional brain imaging including single-photon emission computed tomography (SPECT) is commonly used to guide the clinician's diagnosis. However, conventional evaluation of SPECT scans often relies on manual reorientation, visual reading and semiquantitative analysis of certain regions of the brain. This paper evaluates different pattern classifiers for the development of a computer aided diagnosis (CAD) system for improving the early AD detection. Discriminant template-based normalized mean square error (NMSE) features of several coronal slices of interest (SOI) were used. The proposed system yielding a 97% AD diagnosis accuracy, reports clear improvements over existing techniques such as the voxel-as-features (VAF) which yields just a 78% classification accuracy.

1 Introduction

Alzheimer disease (AD) [1] is a progressive neurodegenerative disease associated with disruption of neuronal function and gradual deterioration in cognition, function, and behavior. It affects approximately 2-4 million individuals in the United States and more than 30 million worldwide. With the growth of the older population in developed nations, the prevalence of AD is expected to triple over the next 50 years. The major goals in treating AD currently are to recognize the disease early in order to initiate appropriate therapy and delay functional and cognitive losses.

Cerebral single-photon emission computed tomography (SPECT), which is based on brain uptake of a technetium 99m-based lipid-soluble radionuclide, is a widely available technique for brain perfusion assessment with a rotating gamma camera. AD patients typically demonstrate a relative paucity of activity in the temporoparietal regions, compared with the activity in control subjects [2]. However, early detection of AD still remains a challenge since conventional

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 941–948, 2009.
© Springer-Verlag Berlin Heidelberg 2009

evaluation of SPECT scans often relies on manual reorientation, visual reading and semiquantitative analysis.

This paper evaluates different pattern classifiers for the development of an early AD SPECT-based computer aided diagnosis (CAD) system [3]. The proposed methods combining pattern recognition and advanced feature extraction schemes are developed with the aim of reducing the subjectivity in visual interpretation of SPECT scans by clinicians, thus improving the accuracy of diagnosing Alzheimer disease in its early stage.

2 Overview of Classifiers

The goal of a binary classifier is to separate a set of binary labeled training data consisting of, in the general case, N-dimensional patterns $\mathbf{v}_i$ and class labels y_i:

$$(\mathbf{v}_1, y_1), (\mathbf{v}_2, y_2), ..., (\mathbf{v}_l, y_l) \in (R^N \times \{\text{Normal}, \text{ATD}\}), \tag{1}$$

so that a classifier is produced which maps an unknown object $\mathbf{v}_i$ to its classification label y_i. Several different classifiers are evaluated in this work.

2.1 Discriminant Analysis

We suppose that $\mathbf{v}$ denotes a p-component random vector of observations made on any individual; $\mathbf{v}_0$ denotes a particular observed value of $\mathbf{v}$, and π_1, π_2 denote the two populations involved in the problem. The basic assumption is that $\mathbf{v}$ has different probability distributions in π_1 and π_2. Let the probability density of $\mathbf{v}$ be $f_1(\mathbf{v})$ in π_1, and $f_2(\mathbf{v})$ in π_2. The simplest intuitive argument, termed the likelihood ratio rule, classifies $\mathbf{v}_0$ as π_1 whenever it has greater probability of coming from π_1 than from π_2. This classification rule can be written as:

$$\mathbf{v} \in \pi_1 \text{ if } f_1(\mathbf{v})/f_2(\mathbf{v}) > 1 \qquad \mathbf{v} \in \pi_2 \text{ if } f_1(\mathbf{v})/f_2(\mathbf{v}) \leq 1. \tag{2}$$

Multivariate Normal model: quadratic discriminant function. The most general form of the model is to assume that π_i is a multivariate normal population with mean $\boldsymbol{\mu}_i$ and dispersion matrix $\boldsymbol{\Sigma}_i$ for $i = 1, 2$. Thus $f_i(\mathbf{v}) = (2\pi)^{-p/2}|\boldsymbol{\Sigma}_i|^{-1/2} \exp\{\frac{1}{2}(\mathbf{v} - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{v} - \boldsymbol{\mu}_i)\}$, so that

$$\begin{aligned} f_1(\mathbf{v})/f_2(\mathbf{v}) = |\boldsymbol{\Sigma}_2|^{1/2}|\boldsymbol{\Sigma}_1|^{-1/2} \exp[-\frac{1}{2}\{\mathbf{v}'(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})\mathbf{v} - \\ 2\mathbf{v}'(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2) + \boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2'\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\}] \end{aligned} \tag{3}$$

Hence, by taking logarithms in equation (2), the classification rule for this model is reduced to allocating $\mathbf{v}$ to π_1 if $Q(\mathbf{v}) > 0$, and otherwise to π_2, where $Q(\mathbf{v})$ is the discriminant function

$$\begin{aligned} Q(\mathbf{v}) = \frac{1}{2}\log\{|\boldsymbol{\Sigma}_2|/|\boldsymbol{\Sigma}_1|\} - \frac{1}{2}\{\mathbf{v}'(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})\mathbf{v} - \\ 2\mathbf{v}'(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2) + \boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2'\boldsymbol{\Sigma}\boldsymbol{\mu}_2\}. \end{aligned} \tag{4}$$

Since the terms in $Q(\mathbf{v})$ include the quadratic form $\mathbf{v}'(\mathbf{\Sigma}_1^{-1} - \mathbf{\Sigma}_2^{-1})\mathbf{v}$, which will be a function of the squares of elements of $\mathbf{v}$ and cross-products between pairs of them, this discriminant function is known as the *quadratic discriminant*. In any practical application, the parameters $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$, $\mathbf{\Sigma_1}$ and $\mathbf{\Sigma_2}$ are not known. Given two training sets, $\mathbf{v}_1^{(1)}, ..., \mathbf{v}_{n_1}^{(1)}$ from π_1, and $\mathbf{v}_1^{(2)}, ..., \mathbf{v}_{n_2}^{(2)}$ from π_2 we can estimate these parameters by:

$$\boldsymbol{\mu}_j = \frac{1}{n_j}\sum_{i=1}^{n_j} \mathbf{v}_i^{(j)}, \quad \mathbf{\Sigma}_j = \frac{1}{n_j - 1}\sum_{i=1}^{n_j}(\mathbf{v}_i^{(j)} - \boldsymbol{\mu}_j)(\mathbf{v}_i^{(j)} - \boldsymbol{\mu}_j)' \quad j = 1, 2 \qquad (5)$$

Multivariate Normal model: linear discriminant function. The presence of two different population dispersion matrices makes more difficult the test about the population mean vectors. Since the assumption $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$ can be reasonable in many practical situations then

$$f_i(\mathbf{v}) = (2\pi)^{-p/2}|\mathbf{\Sigma}|^{-1/2}\exp\{-\frac{1}{2}(\mathbf{v} - \boldsymbol{\mu}_i)'\mathbf{\Sigma}^{-1}(\mathbf{v} - \boldsymbol{\mu}_i)\} \qquad (6)$$

so that the classification rule reduces to allocating $\mathbf{v}$ to π_1 if $L(\mathbf{v}) > 0$, and otherwise to π_2, where $L(\mathbf{v}) = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\mathbf{\Sigma}^{-1}\{\mathbf{v} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)\}$. Note that, no quadratic terms now exist in the discriminant function $L(\mathbf{v})$, which is therefore called the *linear discriminant function*. In that case, contrary to the quadratic case, we estimate the pooled covariance matrix:

$$\mathbf{\Sigma} = \frac{1}{n_1 + n_2 - 2}\{\sum_{i=1}^{n_1}(\mathbf{v}_i^{(1)} - \boldsymbol{\mu}_1)(\mathbf{v}_i^{(1)} - \boldsymbol{\mu}_1)' + \sum_{i=1}^{n_2}(\mathbf{v}_i^{(2)} - \boldsymbol{\mu}_2)(\mathbf{v}_i^{(2)} - \boldsymbol{\mu}_2)'\} \qquad (7)$$

Mahalanobis distance. The Mahalanobis distance differs from Euclidean distance in that it takes into account the correlations of the data set and its scale-invariant nature. We allocate $\mathbf{v}$ to π_1 if $\Delta_1 > \Delta_2$, and otherwise to π_2, where Δ_1, Δ_2 are the Mahalanobis distance between $\mathbf{v}$ and π_1, π_2 respectively:

$$\Delta_1^2 = (\mathbf{v} - \boldsymbol{\mu}_1)'\mathbf{\Sigma_1}(\mathbf{v} - \boldsymbol{\mu}_1), \quad \Delta_2^2 = (\mathbf{v} - \boldsymbol{\mu}_2)'\mathbf{\Sigma_2}(\mathbf{v} - \boldsymbol{\mu}_2) \qquad (8)$$

2.2 k-Nearest Neighbor

An object is classified by a majority vote of its neighbors, with the object being assigned to the most common class amongst its k nearest neighbors. k is a positive integer, typically small. For instance, if $k = 1$, then the object is simply assigned to the class of its nearest neighbor.

2.3 Classification Trees

Classification trees is a non-parametric technique that produces classification of categorical dependent variables [4]. Binary tree structured classifiers are constructed by repeated splits of subsets of X into two descendant subsets, beginning

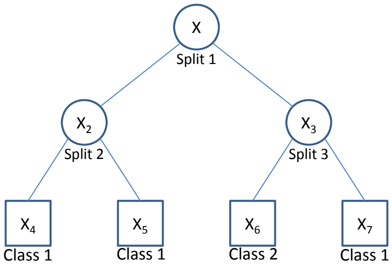


Fig. 1. Hypothetical two-class tree

with X itself. This process is plotted in Fig. 1 for a two classes tree. In the figure, X_2 and X_3 are disjoint with $X = X_2 \cup X_3$. Similarly, X_4 and X_5 are disjoint with $X_4 \cup X_5 = X_2$. Those subsets which are not split, in this case X_4, X_5, X_6 and X_7 are called terminal nodes. Each terminal node is designated by a class label. There may be more than one terminal subset with the same class label.

The first problem in tree construction is how to use the learning sample to determine the binary splits of X into smaller pieces. In order to build a classification tree, three questions need to be solved: how to select the splits, when to declare a node terminal or split and how to assign a class to each terminal node. Once a good split of X is found, then a search is made for good splits of each of the two descendant nodes.

2.4 Support Vector Machines

Support vector machines (SVM) separate a given set of binary labeled training data with a hyperplane that is maximally distant from the two classes (known as the maximal margin hyper-plane). The objective is to build a function $f : R^N \longrightarrow \{\pm 1\}$ using training data that is, N-dimensional patterns $\mathbf{x}_i$ and class labels y_i:

$$(\mathbf{v}_1, y_1), (\mathbf{v}_2, y_2), ..., (\mathbf{v}_l, y_l) \in R^N \times \{\pm 1\}, \tag{9}$$

so that f will correctly classify new examples $(\mathbf{v}, y)$. When no linear separation of the training data is possible, SVM can work effectively in combination with kernel techniques so that the hyperplane defining the SVM corresponds to a non-linear decision boundary in the input space. If the data is mapped to some other (possibly infinite dimensional) Euclidean space using a mapping $\Phi(\mathbf{v})$, the training algorithm only depends on the data through dot products in such an Euclidean space, i.e. on functions of the form $\Phi(\mathbf{v}_i) \cdot \Phi(\mathbf{v}_j)$. If a "kernel function" K is defined such that $K(\mathbf{v}_i, \mathbf{v}_j) = \Phi(\mathbf{v}_i) \cdot \Phi(\mathbf{v}_j)$, it is not necessary to know the Φ function during the training process. In the test phase, an SVM is used by computing dot products of a given test point $\mathbf{v}$ with $\mathbf{w}$, or more specifically by computing the sign of

$$f(\mathbf{v}) = \sum_{i=1}^{N_S} \alpha_i y_i \Phi(\mathbf{s}_i) \cdot \Phi(\mathbf{v}) + w_0 = \sum_{i=1}^{N_S} \alpha_i y_i K(\mathbf{s}_i, \mathbf{v}) + w_0, \tag{10}$$

where $\mathbf{s}_i$ are the support vectors.

3 Image Acquisition, Preprocessing and Feature Extraction

SPECT scans are registered after injecting the patient a gamma emitting ^{99m}Tc-ECD radiopharmaceutical by means of a three-head gamma camera Picker Prism 3000. 3D brain perfusion volumes are reconstructed from projection data using the filtered backprojection (FBP) algorithm in combination with a Butterworth noise removal filter. The SPECT images are spatially normalized using the SPM software [5] in order to ensure that the voxels in different images refer to the same anatomical positions in the brain [6], giving rise to images of voxel size 69×95×79. Finally, intensity level of the SPECT images is normalized to the maximum intensity, following a procedure similar to [7]. The images were initially labeled by experienced clinicians of the Virgen de las Nieves hospital (Granada, Spain), using 4 different labels: normal (NOR) for patients without any symptoms of ATD and possible ATD (ATD-1), probable ATD (ATD-2) and certain ATD (ATD-3) to distinguish between different levels of the presence of typical characteristics for ATD. In total, the database consists of 79 patients: 41 NOR, 20 ATD-1, 14 ATD-2 and 4 ATD-3.

Similarity measures between the functional activity of normal controls and each subject were used as features. First, the expected value of the voxel intensity of the normal subjects was computed by averaging the voxel intensities of all the normal controls in the database. Then, the Normalized Mean Square Error

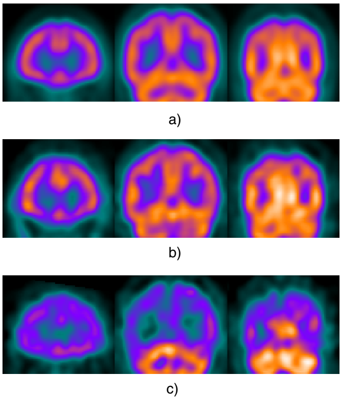
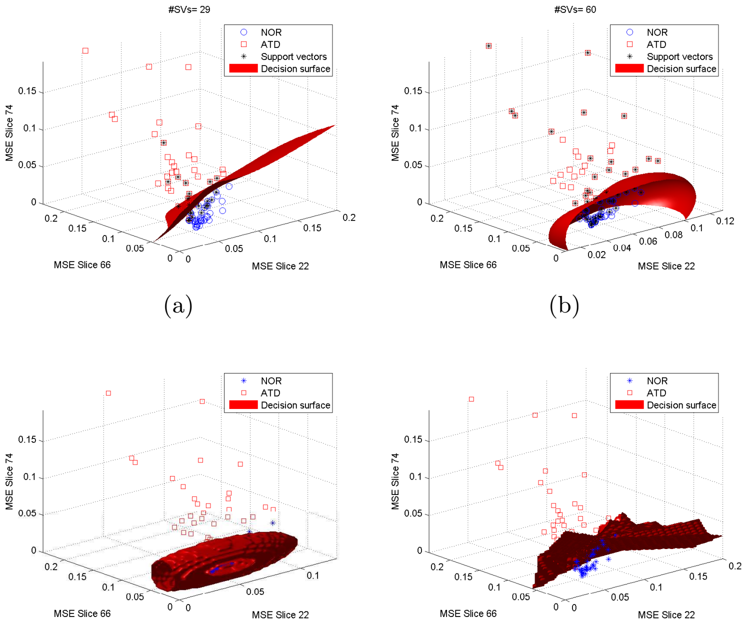


Fig. 2. Coronal slices of: a) Template, b) Normal subject, and c) AD patient



(a) (b)

(c) (d)

Fig. 3. Decision surfaces for different classifiers: a) SVM (polynomial kernel), b) SVM (RBF kernel), c) Discriminant analysis (diagonal quadratic), and d) kNN (k= 3)

(NMSE) between slices of each subject and the template, and defined for 2-D slices of the volume to be:

$$NMSE = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [f(m,n) - g(m,n)]^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [f(m,n)]^2} \tag{11}$$

where $f(m, n)$ defines the reference template and $g(m, n)$ the voxel intensities of each subject, was computed for coronal, transaxial, and sagittal slices.

A study was carried out in order to identify the most discriminating slices of interest (SOI) by means of a SVM-classified trained on NMSE features of transaxial, coronal and sagittal slices. The analysis showed the high discrimination ability of specific NMSE features of coronal slices. Fig. 2 shows the differences in regional cerebral blood flow (rCBF) provided by these SOI. It shows three different coronal slices of a template brain obtained by averaging the functional SPECT of 41 controls (Fig. 2.a) together with the corresponding coronal slices of a normal subject (Fig. 2.b) and an AD patient (Fig. 2.c). It can be concluded that the rCBF of patients affected by Alzheimer disease is significantly reduced when compared to the expected value for a normal subject. This reduction affects more to some specific cerebral regions. This result is in agreement with many other studies that have shown the temporo-parietal region to be practical for the early detection of the disease in patients that are no longer characterized by specific cognitive impairment but by general cognitive decline.

4 Evaluation Results

First of all, a baseline classifier using voxel-as-features (VAF) [8] was developed for reference. The dimension of the 95×69×79-voxel volume representing the rCBF of each subject was reduced by decimating the original 3D volume and a SVM-based classifier was trained and tested based on a leave-one-out cross validation strategy. Accuracy of the baseline system was 78.5%. The reference VAF system was compared to different classifiers using the proposed NMSE features of the three most significant slices for AD detection that were shown and discussed in Fig. 2. Fig. 3 shows the 3-D input space defined by means of these three coronal NMSE features and the ability of different classifiers (discriminant, analysis, kNN, SVM, ...) to separate the two classes (normal controls in blue *vs.* patients affected by DTA in red) by means of decision surfaces defined by defining classifiers. The shape of the decision rule strongly depends on the method for formulating the decision rule and its associated parameters. Among the different classification techniques considered, SVM with almost linear polynomial kernels is the one that better separates the two classes.

Table 1. Diagnosis accuracy for the different classifiers evaluated. Comparison to the VAF baseline.

Discriminat Analysis	Linear	Diag-Linear	Quadratic	Diag-Quadratic
NMSE (Coronal SOI)	84.8	87.3	91.1	74.7
kNN k =	3	5	7	9
NMSE (Coronal SOI)	96.2	93.7	94.9	93.7
Decision trees				
NMSE (Coronal SOI)	88.6			
SVM kernel	Lineal	Quadratic	RBF	Polynomial
Baseline VAF	78.5	68.4	51.9	53.2
NMSE (All)	93.7	69.6	35.4	83.5
NMSE (Coronal SOI)	97.5	93.7	94.9	97.5

Table 1 shows the accuracy of the proposed and reference VAF systems evaluated by a leave-one-out cross-validation strategy. Results for the system using the NMSE features of all the coronal slices as well as the system using just the three most discriminative ones are included. It can be concluded that the proposed NMSE features using carefully selected coronal slices improves the performance of the system using information of all the brain volume corroborating the evidence that only selected brain areas are mainly affected by hypo-perfusion in patients suffering the Alzheimer disease. The best results are obtained for an almost linear kernel SVM system yielding a 97.5% accuracy and outperforming the VAF approach which obtains the best results for linear kernels (78.5%).

5 Conclusions

This paper showed an study for the selection of the optimum classification technique for the development of an AD CAD system. The analysis considered classifiers based on discriminant analysis, k-nearest neighbor, classification trees and support vector machines. Template-based features defined in terms of the so called Normalized Mean Square error, which measures the difference in regional cerebral blood flow of several coronal SOI, were found to be very effective for discriminating between AD patients and normal controls. With these and other innovations, the proposed system yielded a 97.5% diagnosis accuracy for almost linear SVM kernels, thus outperforming the 78.5% accuracy of the classical baseline VAF approach .

Acknowledgments. This work was partly supported by the Spanish Government under the PETRI DENCLASES (PET2006-0253), TEC2008-02113, NAPOLEON (TEC2007-68030-C02-01) projects and the Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía, Spain) under the Excellence Project (TIC-02566).

References

1. Petrella, J.R., Coleman, R.E., Doraiswamy, P.M.: Neuroimaging and early diagnosis of Alzheimer disease: A look to the future. Radiology 226, 315–336 (2003)
2. Holman, B.L., Johnson, K.A., Gerada, B., Carvaiho, P.A., Sathn, A.: The scintigraphic appearance of Alzheimer's disease: a prospective study using Tc-99m HMPAO SPECT. Journal of Nuclear Medicine 33(2), 181–185 (1992)
3. Ramírez, J., Górriz, J.M., López, M., Salas-Gonzalez, D., Álvarez, I., Segovia, F., Puntonet, C.G.: Early detection of the Alzheimer disease combining feature selection and kernel machines. In: ICONIP 2008. LNCS. Springer, Heidelberg (2008)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Chapman and Hall, Boca Raton (1993)
5. Ramírez, J., Górriz, J.M., Gómez-Río, M., Romero, A., Chaves, R., Lassl, A., Rodríguez, A., Puntonet, C.G., Theis, F.J., Lang, E.: Effective emission tomography image reconstruction algorithms for SPECT data. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 741–748. Springer, Heidelberg (2008)
6. Salas-González, D., Górriz, J.M., Ramírez, J., Lassl, A., Puntonet, C.G.: Improved Gauss-Newton optimization methods in affine registration of SPECT brain images. IET Electronics Letters 44(22), 1291–1292 (2008)
7. Saxena, P., Pavel, D.G., Quintana, J.C., Horwitz, B.: An automatic threshold-based scaling method for enhancing the usefulness of Tc-HMPAO SPECT in the diagnosis of Alzheimers disease. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 623–630. Springer, Heidelberg (1998)
8. Stoeckel, J., Malandain, G., Migneco, O., Koulibaly, P.M., Robert, P., Ayache, N., Darcourt, J.: Classification of SPECT images of normal subjects versus images of Alzheimer's disease patients. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 666–674. Springer, Heidelberg (2001)

Automatic System for Alzheimer's Disease Diagnosis Using Eigenbrains and Bayesian Classification Rules

M. López[1,⋆], J. Ramírez[1], J.M. Górriz[1], I. Álvarez[1], D. Salas-Gonzalez[1],
F. Segovia[1], and C.G. Puntonet[2]

[1] Dept. of Signal Theory, Networking and Communications,
University of Granada, Spain
[2] Dept. of Architecture and Computer Technology,
University of Granada, Spain

Abstract. Alzheimer's Disease (AD) is a progressive neurologic disease of the brain that leads to the irreversible loss of neurons and dementia. The new brain imaging techniques PET (Positron Emission Tomography) and SPECT (Single Photon Emission Computed Tomography) provide functional information about the brain activity and have been widely used in the AD diagnosis process. However, the diagnosis currently relies on manual image reorientations, visual evaluation and other subjective, time consuming steps. In this work, a complete computer aided diagnosis (CAD) system is developed to assist the clinicians in the AD diagnosis process. It is based on bayesian classifiers, made up from features previously extracted. The small size sample problem, consisting of having a number of available samples much lower than the dimension of the feature space, is faced up by applying Principal Component Analysis (PCA) to the features. This approach provides higher accuracy values than other previous approaches do, yielding 91.21% and 98.33% accuracy values for SPECT and PET images, respectively.

Keywords: SPECT, PET, Alzheimer Type Dementia, Principal Component Analysis, Bayesian Classification.

1 Introduction

Alzheimer's Disease (AD) is a progressive, degenerative brain disorder that gradually destroys memory, reason, judgment, language, and eventually the ability to carry out even the simplest tasks. The current method to diagnose the AD is based on a whole study of the patient, including personal interviews with the patient's family members, physical and neurological exams and brain imaging.

[⋆] This work was partly supported by the Spanish Government under the PETRI DENCLASES (PET2006-0253), TEC2008-02113, NAPOLEON (TEC2007-68030-C02-01) projects and the Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía, Spain) under the Excellence Project (TIC-02566).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 949–956, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Recently, scientists have begun to do research on diagnosing AD with different kinds of brain imaging, trying to diagnose this dementia in its early stage, when the application of the treatment is more effective. Two types of non-invasive (i. e., no surgery is required) tests have been widely used in the AD diagnosis. Positron Emission Tomography (PET) scan is an imaging scan that measures the level of functional activity of the brain by measuring its use of glucose. Single Photon Emission Computed Tomography (SPECT) scan is a procedure that measures blood flow in different areas of the brain. For both tests PET and SPECT, a small amount of radioactive material is injected into the patient and emission detectors are placed on the brain, providing functional information about the brain activity. Nowadays, the evaluation of the brain images is carried out by an expert clinician who manually reorients and visually examines the images, giving a diagnostic about the patient's state.

Several approaches have been recently proposed in the literature aiming at providing an automatic tool that guides the clinician in the AD diagnosis process [1,2]. These approaches can be categorized into two types: univariate and multivariate approaches. The first family is statistical parametric mapping (SPM) [3] and its numerous variants. SPM consists of doing a voxelwise statistical test, comparing the values of the image under study to the mean values of the group of normal images. Subsequently the significant voxels are inferred by using random field theory. It was not developed specifically to study a single image, but for comparing groups of images. The second family is based on the analysis of the images, feature extraction and posterior classification in different classes, depending on the patient's state. The main problem to be faced up by these techniques is the well-known *curse of the dimensionality*, that is, the number of available samples is much lower than the number of features used in the training step. Among these techniques, Voxels-As-Features (VAF) for SPECT images [1] and others based on the study of regions of interest (ROIs) for SPECT and PET images [4,5] have been developed.

In this work, a fully computer aided diagnosis (CAD) tool is shown. This approach belongs to the multivariate family, and faces up the small size sample problem by applying Principal Component Analysis (PCA) to the feature vector, which allows us to reduce drastically the dimension and makes it comparable to the number of training samples. PCA was already applied in [3] in a qualitative way, but never used the coefficients as features for classification. In this approach, after the PCA transformation, the resultant feature vectors are used to made up a bayesian classifier, which uses the *a posteriori* information to classify new coming images.

2 Image Preprocessing

In order to make possible a direct comparison of the voxel intensities between SPECT or PET images, a previous normalization of the images is needed. For all the experiments, we normalize the images by applying an affine transformation to the intensities as suggested in [3]. All the images of the database are transformed

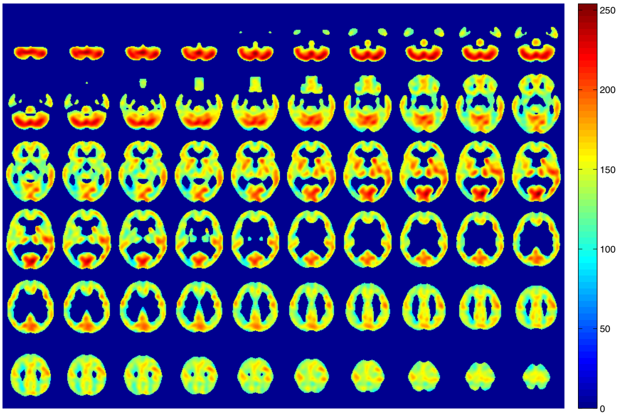


Fig. 1. SPECT average image of the dataset once the mask has been applied

using affine and non-linear spatial normalization, thus the basic assumptions are met.

After the normalization steps, a 79 × 95 × 69 brain volume representation is obtain for each subject. However, not all these voxels are significant in the AD diagnosis. Voxels in the black edges outside the brain zone or voxels with low gray level (both in normal and AD patients) can be rejected for the classification task. As proposed in [4], we first construct a binary mask which selects the voxels of interest and discards the rest. This is done by taking the voxels whose mean intensity value averaged over all images exceeds the half of the maximum mean intensity value, and this mask is applied to the original images. In the resulting masked images, the irrelevant information has been removed or reduced. Fig. 1 represents the masked SPECT average image along the transaxial axis. The mask application has rejected those voxels whose intensity values are lower than 128.

3 Principal Component Analysis and Eigenbrains

PCA generates a set of orthonormal basis vectors, known as principal components, which maximize the scatter of all the projected samples. After the preprocessing steps, the N remaining voxels are rearranged in a vector form. Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n]$ be the sample set of these vectors, where n is the number of patients, each of dimensionality N. After normalizing the vectors to unity norm and subtracting the grand mean a new vectors set $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_n]$ is obtained, where each $\mathbf{Y}_i$ represents a normalized vector with dimensionality N, $\mathbf{Y}_i = (y_{i1}, y_{i2}, ..., y_{iN})^t, i = 1, 2, ..., n$. The covariance matrix of the normalized vectors set is defined as

$$\mathbf{\Sigma}_Y = \frac{1}{n} \sum_{i=1}^{n} \mathbf{Y}_i \mathbf{Y}_i^t = \frac{1}{n} \mathbf{Y}\mathbf{Y}^t \tag{1}$$

and the eigenvector and eigenvalue matrices $\mathbf{\Phi}$, $\mathbf{\Lambda}$ are computed as

$$\mathbf{\Sigma}_Y \mathbf{\Phi} = \mathbf{\Phi}\mathbf{\Lambda} \tag{2}$$

Note that $\mathbf{Y}\mathbf{Y}^t$ is an $N \times N$ matrix while $\mathbf{Y}^t\mathbf{Y}$ is an $n \times n$ matrix. If the sample size n is much smaller than the dimensionality N, then diagonalizing $\mathbf{Y}^t\mathbf{Y}$ instead of $\mathbf{Y}\mathbf{Y}^t$ reduces the computational complexity [6]

$$(\mathbf{Y}^t\mathbf{Y})\mathbf{\Psi} = \mathbf{\Psi}\mathbf{\Lambda}_1 \tag{3}$$

$$\mathbf{T} = \mathbf{Y}\mathbf{\Psi} \tag{4}$$

where $\mathbf{\Lambda}_1 = diag\{\lambda_1, \lambda_2, ..., \lambda_n\}$ and $\mathbf{T} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, ..., \mathbf{\Phi}_n]$. Derived from the *eigenface* concept [6], the *eigenbrains* correspond to the dominant eigenvectors of the covariance matrix. In this approach, only m leading eigenvectors are used, which define the matrix $\mathbf{P}$

$$\mathbf{P} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, ..., \mathbf{\Phi}_m] \tag{5}$$

The criterion to choose the most discriminant eigenbrains is set by their separation ability, which is measured by the Fisher Discriminant Ratio (FDR), defined as

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{6}$$

where μ_i and σ_i denote the i-th class within class mean value and variance, respectively. For the whole database, a matrix of weights can be constructed, given by:

$$\mathbf{Z} = \mathbf{P}^t\mathbf{Y} \tag{7}$$

4 Bayes Classifier

For pattern recognition, the Bayes classifier is the best classifier in terms of minimum Bayes error, therefore the *a posteriori* probability functions will be evaluated [7]. Let ω_1 and ω_2 denote the object classes (AD and NORMAL), and $\mathbf{Z}$ a patient voxels vector in the reduced PCA subspace. The *a posteriori* probability function of ω_i given $\mathbf{Z}$ is defined as

$$P(\omega_i|\mathbf{Z}) = \frac{p(\mathbf{Z}|\omega_i)P(\omega_i)}{p(\mathbf{Z})}, \quad i = 1, 2. \tag{8}$$

where $P(\omega_i)$ is *a priori* probability, $p(\mathbf{Z}|\omega_i)$ the conditional probability density function of $\mathbf{Z}$ given ω_i, and $p(\mathbf{Z})$ is the mixture density. The maximum *a posteriori* (MAP) decision rule for the Bayes classifier is defined as

$$p(\mathbf{Z}|\omega_i)P(\omega_i) = \max_j\{p(\mathbf{Z}|\omega_j)P(\omega_j)\}, \quad \mathbf{Z} \in \omega_i \tag{9}$$

The brain projected data $\mathbf{Z}$ is classified to ω_i of whom the *A Posteriori* probability given $\mathbf{Z}$ is the largest between the classes. Usually there are not enough samples to estimate the conditional probability density function for each class (within class density). The within class densities are usually modeled as normal distributions

$$p(\mathbf{Z}|\omega_i) = \frac{1}{(2\pi)^{m/2}|\mathbf{\Sigma}_i|^{1/2}} \times exp\left\{-\frac{1}{2}(\mathbf{Z}-\mathbf{M}_i)^t\mathbf{\Sigma}_i^{-1}(\mathbf{Z}-\mathbf{M}_i)\right\} \qquad (10)$$

where $\mathbf{M}_i$ (see Eq. 11) and $\mathbf{\Sigma}_i$ are the mean and covariance matrix of class ω_i, respectively.

4.1 Probabilistic Reasoning Model (PRM)

Under the unified Bayesian framework, two new probabilistic reasoning models, PRM-1 and PRM-2 are derived in [8], which utilize the within class scatters to derive averaged estimations of within class covariance matrices.

In particular, let ω_1, ω_2 and N_1, N_2 denote the classes and number of patients within each class, respectively. Let $\mathbf{M}_1, \mathbf{M}_2$ be the means of the classes in the reduced PCA subspace span $[\mathbf{\Phi}_1, \mathbf{\Phi}_2, ..., \mathbf{\Phi}_m]$. We then have

$$\mathbf{M}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\mathbf{Z}_j^{(i)}, \quad i = 1, 2. \qquad (11)$$

where $\mathbf{Z}_j^{(i)}, j = 1, 2, ..., N_i$ represents the sample voxels vector for the ω_i class.

PRM-1. The PRM-1 model assumes the within class covariance matrices are identical and diagonal

$$\mathbf{\Sigma}_i = diag\left\{\sigma_1^2, \sigma_2^2, ..., \sigma_m^2\right\} \qquad (12)$$

Each component σ_i^2 can be estimated by sample variance in the one dimensional PCA subspace

$$\sigma_i^2 = \frac{1}{L}\sum_{k=1}^{L}\left\{\frac{1}{N_k-1}\sum_{j=1}^{N_k}\left(z_{ij}^{(k)} - m_{ki}\right)^2\right\} \qquad (13)$$

where $z_{ij}^{(k)}$ is the i-th element of the sample $\mathbf{Z}_j^{(k)}$, m_{ki} the i-th element of $\mathbf{M}_k$, and L the number of classes (two in our approach).

PRM-2. The PRM-2 model estimates the within class scatter matrix Σ_ω in the reduced PCA space as

$$\Sigma_\omega = \frac{1}{L}\sum_{k=1}^{L}\left\{\frac{1}{N_k}\sum_{j=1}^{N_k}\left(Z_j^{(k)} - M_k\right)\left(Z_j^{(k)} - M_k\right)^t\right\} \qquad (14)$$

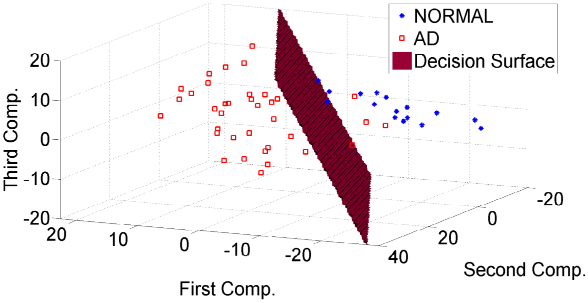


Fig. 2. For PET images, distributions of the three first principal coefficients for AD and NORMAL patients and the decision surface

The Σ_ω is diagonilized using singular value decomposition (SVD)

$$\Sigma_\omega = USV^t \tag{15}$$

where U and V are unitary matrices, S is a diagonal matrix

$$S = diag\{s_1, s_2, ..., s_m\} \tag{16}$$

with non-negative singular values as diagonal elements. The squared diagonal elements are ordered in decreasing value

$$\left(s^2_{(1)}, s^2_{(2)}, ..., s^2_{(m)}\right) = order\{s^2_1, s^2_2, ..., s^2_m\} \tag{17}$$

Finally, the within class covariance matrix is derived as

$$\Sigma_I = diag\left\{s^2_{(1)}, s^2_{(2)}, ..., s^2_{(m)}\right\} \tag{18}$$

5 Evaluation Results

SPECT and PET images used in this work were taken with a PRISM 3000 machine and a SIEMENS ECAT 47 respectively. Initially they were labeled by experienced clinicians of the “Virgen de las Nieves” Hospital (Granada, Spain) and “Clínica PET Cartuja” (Seville, Spain) respectively. The database consists of 91 SPECT patients (41 labeled as NORMAL and 50 labeled as AD) and 60 PET patients (18 NORMAL and 42 AD). Initially, the original brain image $79\times95\times69$ voxel sized is reduced by averaging over subsets of $4\times4\times4$ voxels. After applying the mask, the remaining voxels are rearranged into a vector form so that PCA can be applied to the training set and eigenbrains are obtained. The new patient to be classified is projected into the eigenbrain space, the *a posteriori* probabilities $P(\mathbf{Z}|\omega_1)$ and $P(\mathbf{Z}|\omega_2)$ are computed (where ω_1 = NORMAL and ω_2 = AD) and

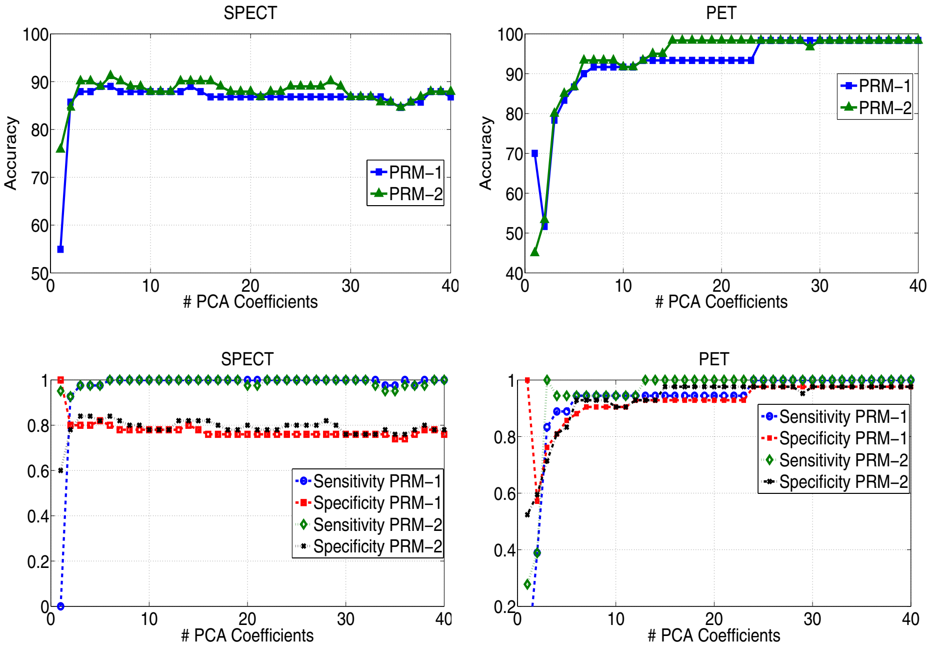


Fig. 3. Accuracy (up) and Sensitivity/Specificity (down) values for SPECT (left) and PET (right) images using bayesian classifiers when the number m of considered principal components increases

Table 1. Accuracy results obtained by the proposed methods and by the work taken as reference (VAF)

	SPECT	PET
VAF approach	85.71%	96.67%
Eigenbrains PRM-1	89.01%	98.33%
Eigenbrains PRM-2	**91.21%**	**98.33%**

the MAP rule (Eq. 9) is applied, both for PRM-1 and PRM-2 models. Fig. 2 represents the three main PCA coefficients as 3D points for NORMAL and AD patients separated by the classifier decision surface. Fig. 3 shows the accuracy and the sensitivity and specificity values obtained for SPECT and PET images when the number of considered principal component m increases.

As it is usually done in cases where the number of available samples is relatively small, the classifier was tested with the Leave-One-Out method, that is, the classifier is trained using all but one patient, which is used in the test phase. This procedure is repeated for each patient and an average accuracy value for all the experiments is obtained. The VAF approach was also implemented and tested by the same cross-validation strategy. Results are shown and compared in Table 1.

6 Conclusions

A computer aided diagnosis system for the early detection of the Alzheimer disease was shown in this paper. The system was developed by performing the principal components analysis to a subset of remaining voxels after some preprocessing steps, which allows us to face the small size problem and reduce drastically the feature space dimension. The most important components in terms of ability to separate are chosen to make up a bayesian classifier. The *a posteriori* information is used when a new patient is needed to be classified. With this approach, 91.21% and 98.33% accuracy values are obtained for SPECT and PET images respectively, which mean an improvement over the reference work.

References

1. Fung, G., Stoeckel, J.: SVM Feature Selection for Classification of SPECT Images of Alzheimer's Disease Using Spatial Information. Knowledge and Information Systems 11(2), 243–258 (2007)
2. Ramírez, J., Górriz, J.M., Salas-Gonzalez, D., Lassl, A., López, M., Puntonet, C. G., Gómez, M., Rodríguez, A.: Computer Aided Diagnosis of Alzheimer Type Dementia Combining Support Vector Machines and Discriminant Set of Features. Accepted in Information Sciences (2008)
3. Friston, K.J., Ashburner, J., Kiebel, S.J., Nichols, T.E., Penny, W.D.: Statistical Parametric Mapping: The Analysis of Functional Brain Images. Academic Press, London (2007)
4. Górriz, J.M., Ramírez, J., Lassl, A., Salas-Gonzalez, D., Lang, E.W., Puntonet, C.G., Álvarez, I., López, M., Gómez-Río, M.: Automatic Computer Aided Diagnosis Tool Using Component-based SVM. In: Medical Imaging Conference, Dresden (2008)
5. Álvarez, I., López, M., Górriz, J.M., Ramírez, J., Salas-Gonzalez, D., Puntonet, C.G., Segovia, F.: Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations. In: Proc. International Conference on Neural Information Processing (2008)
6. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 13(1), 71–86 (1991)
7. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, London (1991)
8. Chengjun, L., Wechsler, H.: A Unified Bayesian Framework for Face Recognition. In: Proceedings in International Conference on Image Processing (1998)

On the Use of Morphometry Based Features for Alzheimer's Disease Detection on MRI

Maite García-Sebastián[1], Alexandre Savio[1], Manuel Graña[1],
and Jorge Villanúa[2]

[1] Grupo de Inteligencia Computacional
www.ehu.es/ccwintco
[2] Osatek, Hospital Donostia Paseo Dr. Beguiristain 109, 20014 San Sebastián, Spain

Abstract. We have studied feature extraction processes for the detection of Alzheimer's disease on brain Magnetic Resonance Imaging (MRI) based on Voxel-based morphometry (VBM). The clusters of voxel locations detected by the VBM were applied to select the voxel intensity values upon which the classification features were computed. We have explored the use of the data from the original MRI volumes and the GM segmentation volumes. In this paper, we apply the Support Vector Machine (SVM) algorithm to perform classification of patients with mild Alzheimer's disease vs. control subjects. The study has been performed on MRI volumes of 98 females, after careful demographic selection from the Open Access Series of Imaging Studies (OASIS) database, which is a large number of subjects compared to current reported studies[1].

1 Introduction

Alzheimer's disease (AD) is a neurodegenerative disorder, which is one of the most common cause of dementia in old people. Currently, due to the socioeconomic importance of the disease in occidental countries it is one of the most studied. The diagnosis of AD is done after the exclusion of other forms of dementia but definitive diagnosis can only be made after a post-mortem study of the brain tissue. This is one of the reasons why Magnetic Resonance Imaging (MRI) based early diagnosis is a current research goal in the neurosciences.

Morphometry analysis has become a common tool for computational brain anatomy studies. It allows a comprehensive measurement of structural differences within or across groups, not only in specific structures but throughout the entire brain. Voxel-based morphometry (VBM) is a computational approach to neuroanatomy that measures differences in local concentrations of brain tissue, through a voxel-wise comparison of multiple brain images [2]. For instance, VBM has been applied to study volumetric atrophy of the grey matter (GM) in areas of neocortex of AD patients vs. control subjects [4,9,16]. The procedure involves the spatial normalization of subject images into a standard space, segmentation

---

[1] Research partially supported by Saiotek research projects BRAINER and S-PR07UN02, and the MEC research project DPI2006-15346-C03-03.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 957–964, 2009.
© Springer-Verlag Berlin Heidelberg 2009

of tissue classes using *a priori* probability maps, smoothing to correct noise and small variations, and voxel-wise statistical tests. Statistical analysis is based on the General Linear Model (GLM) to describe the data in terms of experimental and confounding effects, and residual variability. Classical statistical inference is used to test hypotheses that are expressed in terms of GLM estimated regression parameters. The computation of a given contrast provides a Statistical Parametric Map, which is thresholded according to the Random Field Theory.

Machine learning methods have become very popular to classify functional or structural brain images to discriminate them into two classes: normal or a specific neurodegenerative disorder. The Support Vector Machine (SVM) either with linear [10,14] or non-linear [7,11] kernels, have been extensively applied for this task. There are studies applying SVM to discriminate AD patients from controls based on Positron Emission Tomography (PET) or Single-Photon Emission Tomography (SPECT) functional volumes [14,15,1]. There are different ways to extract features from MRI for SVM classification: based on morphometric methods [6,7], based on ROIs (region of interest) [12,11] or GM voxels in automated segmentation images [10]. Work has also been reported on the selection of a small set of the most informative features for classification, such as the SVM-Recursive Feature Elimination [7], the selection based on statistical tests [12,14], the wavelet decomposition of the RAVENS maps [11], among others.

Many of the classification studies on the detection of AD were done over populations mixing men and women. However, it has been demonstrated that brains of women are different from men's to the extent that it is possible to discriminate the gender via MRI analysis [11]. Moreover, it has been shown that VBM is sensitive to the gender differences. For these reasons, we have been very cautious in this study. We have selected a set of 98 MRI women's brain volumes. It must be noted that this is a large number of subjects compared with the other studies referred above.

The approach taken in this paper is to use the clusters detected as result of VBM as a mask on the MRI and Grey Matter (GM) segmentation images to select the potentially most discriminating voxels. Features for classification are either the voxel values or some summary statistics of each cluster. We assume for classification the standard SVM, testing linear and non-linear (RBF) kernels. Section Materials and Methods gives a description of the subjects selected for the study, the image processing, feature extraction details and the classifier system. Section Results gives our classification performance results and section Conclusions gives some conclusions and further work suggestions.

2 Materials and Methods

2.1 Subjects

Ninety eight right-handed women (aged 65-96 yr) were selected from the Open Access Series of Imaging Studies (OASIS) database (http://www.oasis-brains.org) [13]. OASIS data set has a cross-sectional collection of 416 subjects covering the adult life span aged 18 to 96 including individuals with early-stage Alzheimer's

Table 1. Summary of subject demographics and dementia status. Education codes correspond to the following levels of education: 1 less than high school grad., 2: high school grad., 3: some college, 4: college grad., 5: beyond college. Categories of socioeconomic status: from 1 (biggest status) to 5 (lowest status). MMSE score ranges from 0 (worst) to 30 (best).

	Very mild to mild AD	Normal
No. of subjects	49	49
Age	78.08 (66-96)	77.77 (65-94)
Education	2.63 (1-5)	2.87 (1-5)
Socioeconomic status	2.94 (1-5)	2.88 (1-5)
CDR (0.5 / 1 / 2)	31 / 17 / 1	0
MMSE	24 (15-30)	28.96 (26-30)

Disease. We have ruled out a set of 200 subjects whose demographic, clinical or derived anatomic volumes information was incomplete. For the present study there are 49 subjects who have been diagnosed with very mild to mild AD and 49 nondemented. A summary of subject demographics and dementia status is shown in table 1.

2.2 Imaging Protocol

Multiple (three or four) high-resolution structural T1-weighted magnetization-prepared rapid gradient echo (MP-RAGE) images were acquired [8] on a 1.5-T Vision scanner (Siemens, Erlangen, Germany) in a single imaging session. Image parameters: TR= 9.7 msec., TE= 4.0 msec., Flip angle= 10, TI= 20 msec., TD= 200 msec., 128 sagittal 1.25 mm slices without gaps and pixels resolution of 256×256 (1×1mm).

2.3 Image Processing and VBM

We have used the average MRI volume for each subject, provided in the OASIS data set. These images are already registered and resampled into a 1-mm isotropic image in atlas space and the bias field has been already corrected [13]. The Statistical Parametric Mapping (SPM5) (http://www.fil.ion.ucl.ac.uk/spm/) was used to compute the VBM which gives us the spatial mask to obtain the classification features. Images were reoriented into a right-handed coordinate system to work with SPM5. The tissue segmentation step does not need to perform bias correction. We performed the modulation normalization for grey matter, because we are interested in this tissue for this study. We performed a spatial smoothing before computing the voxel-wise statistics, setting the Full-Width at Half-Maximum (FWHM) of the Gaussian kernel to 10mm isotropic. A GM mask was created from the average of the GM segmentation volumes of the subjects under study. Thresholding the average GM segmentation, we obtain a binary mask that includes all voxels with probability greater than 0.1 in the average GM segmentation volume. This interpretation is not completely true, since the data are modulated, but it is close enough for the mask to be reasonable. We design the statistical analysis as a Two-sample t-test in which the first group corresponds with AD subjects. We also have done some experiments with nWBV (normalized whole brain volume) as the covariate. The general linear model contrast has been set as [-1 1], a right-tailed (groupN > groupAD), correction FWE and p-value=0.05. The VBM detected clusters are used for the MRI feature extraction for the SVM classification.

2.4 Support Vector Machine Classification

The Support Vector Machine (SVM) [17] algorithm used for this study is included in the libSVM (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`) software package. The implementation is described in detail in [5]. Given training vectors $x_i \in R_n, i = 1, \ldots, l$ of the subject features of the two classes, and a vector $y \in R^l$ such that $y_i \in \{-1, 1\}$ labels each subject with its class, in our case, for example, patients were labeled as -1 and control subject as 1. To construct a classifier, the SVM algorithm solves the following optimization problem:

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

subject to $y_i(w^T \phi(x_i) + b) \geq (1 - \xi_i),\ \xi_i \geq 0, i = 1, 2, \ldots, n$. The dual optimization problem is

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

subject to $y^T \alpha = 0,\ 0 \leq \alpha_i \leq C,\ i = 1, \ldots, l$. Where e is the vector of all ones, $C > 0$ is the upper bound on the error, Q is an l by l positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the kernel function that describes the behaviour of the support vectors. Here, the training vectors x_i are mapped into a higher (maybe infinite) dimensional space by the function $\phi(x_i)$. The decision function is $sgn(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b)$.

The chosen kernel function results in different kinds of SVM with different performance levels, and the choice of the appropriate kernel for a specific application is a difficult task. In this study two different kernels were tested: the linear and the radial basis function (RBF) kernel. The linear kernel function is defined as $K(x_i, x_j) = 1 + x_i^T x_j$, this kernel shows good performance for linearly separable data. The RBF kernel is defined as $K(x_i, x_j) = exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$. This kernel is basically suited best to deal with data that have a class-conditional probability distribution function approaching the Gaussian distribution [3]. One of the advantages of the RBF kernel is that given a kernel, the number of support vectors and the support vectors are all automatically obtained as part of the training procedure, i.e., they don't need to be specified by the training mechanism.

2.5 Feature Extraction

We have tested three different feature extraction processes, based on the voxel location clusters obtained from the VBM analysis:

1. The first feature extraction process computes the ratio of GM voxels to the total number of voxels of each voxel location cluster
2. The second feature extraction process computes the mean and standard deviation of the GM voxel intensity values of each voxel location cluster
3. The third feature extraction process computes a very high dimensional vector with all the GM segmentation values for the voxel locations included in each VBM detected cluster. The GM segmentation voxel values were ordered in this feature vector according to the coordinate lexicographic order

2.6 Classifier Performance Indices

We evaluated the performance of the classifier using the 10-fold cross-validation test. To quantify the results we measured the accuracy, the ratio of the number of test volumes correctly classified to the total of tested volumes. We also quantified the specificity and sensitivity of each test defined as $Specificity = \frac{TP}{TP+FP}$ and $Sensitivity = \frac{TN}{TN+FN}$, where TP is the number of true positives: number of AD patient volumes correctly classified; TN is the number of true negatives: number of control volumes correctly classified; FP is the number of false positives: number of AD patient volumes classified as control; FN is the number of false negatives: number of control volumes classified as patient.

3 Results

In this section we present for each experiment the following data: the number of features, accuracy, specificity, which is related to AD patients and sensitivity, which is related to control subjects. We have performed the VBM twice, first without any covariate included in the GLM (Table 2) and second taking into account the normalized brain volume (nWBV) (Table 3). Each VBM process produces different sets of voxel location clusters, and, therefore, different sets of feature vectors. The covariate helps to focus the VBM, giving less and smaller clusters than the VBM without covariates. This implies that the feature vectors will be smaller. Each table entry contains the SVM results using the linear and RBF kernels upon the corresponding feature vector set. In both tables rows correspond to feature extraction processes as described in section 2.5.

The best accuracy result (Table 2) is 80.6% with the RBF kernel, but this result is not far away from the results of the linear SVM. The classification results of table 3, using the covariate nWBV in the design matrix of the GLM, confirm that the non-linear SVM is more accurate. However, as the size of the feature vectors is lower than in table 2, results in table 3 are systematically lower.

Overall the sensitivity results in tables 2 and 3 are much lower than the specificity. We hypothesize that the source of error is the confusion of mild demented AD patients with control subjects. Mild demented AD patients are subjects with CDR=0.5 (Clinical Dementia Ratio) and a high value for the MMSE (Minimental-State Examination), i.e. MMSE=30. Therefore we repeat the feature extraction and classification experiment taking out of the population

Table 2. Classification results with a linear kernel (lk) and a non-linear kernel (nlk). No covariates have been taken into account in the GLM used for the VBM. The values of $\gamma = \left(2\sigma^2\right)^{-1}$ for non linear kernel were 0.5, 0.031, 0.0078 for each feature extraction process, respectively.

Feature extracted	Features	Accuracy (lk/nlk)	Sensitivity (lk/nlk)	Specificity (lk/nlk)
GM proportion	12	69.39% / 68.36%	0.88 / 0.90	0.63 / 0.61
Mean & StDev	24	78.57% / 80.61%	0.88 / 0.89	0.72 / 0.75
Voxel intensities	3611	73.47% / 76.53%	0.75 / 0.76	0.72 / 0.77

Table 3. Classification results with a linear kernel (lk) and a non-linear kernel (nlk). The normalized brain volume (nWBV) covariate has been taken into account in the GLM for the VBM. The values of γ for nlk were 0.5, 2.7, 0.004 for GM proportion, Mean & StDev and voxel intensities respectively.

Feature extracted	Features	Accuracy (lk/nlk)	Sensitivity (lk/nlk)	Specificity (lk/nlk)
GM proportion	2	51% / 51%	1 / 1	0.50 / 0.50
Mean & StDev	4	69.38% / 72.45%	0.79 / 0.79	0.65 / 0.68
Voxel intensities	265	66.32% / 75.51%	0.67 / 0.80	0.65 / 0.72

Table 4. Classification results of 40 AD patients vs. 49 control subjects with the SVM and a RBF kernel, 9 possible outliers were taken out from the AD patients subset

Feature extracted	Features	γ	Accuracy	Sensitivity	Specificity
GM proportion	12	0.9	72.5%	0.84	0.66
Mean & StDev	24	0.6	87.5%	0.89	0.86
Voxel intensities	3611	1.5	86.25%	0.85	0.87

9 mild demented AD patients. The results for the RBF kernel SVM are given in table 4. The classification accuracy of the grows from 80.6% (in the best result of table 2) up to 87.5%. Also sensitivity and specificity improve if we compare table 2 and table 3 against table 4.

4 Conclusions

In this work we have studied feature extraction processes based on VBM analysis, to classify MRI volumes of AD patients and normal subjects. We have analyzed different designs for the SPM of the VBM and we have found that the basic GLM design without covariates can detect subtle changes between AD patients and controls that lead to the construction of SVM classifiers with a discriminative accuracy of 87.5%. In [6] they compare their results on a smaller population of controls and AD patients to the ones obtained with a standard VBM analysis using only one cluster and found a classification accuracy of 63.3% via cross-validation. Therefore, the results shown in this paper, along with the careful experimental methodology employed, can be of interest for the Neuroscience community researching on the AD. Further work may address the extraction of features based on other morphometric methods, such as Deformation-based Morphometry.

References

1. Alvarez, I., Lopez, M., Gorriz, J.M., Ramirez, J., Salas-Gonzalez, D., Segovia, F., Puntonet, C.G.: Automatic classification system for the diagnosis of alzheimer disease using Component-Based SVM aggregations. In: 15th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Assembly (ICONIP 2008) (2008)
2. Ashburner, J., Friston, K.J.: Voxel-based morphometry: The methods. Neuroimage 11(6), 805–821 (2000)
3. Burges, C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 167, 121 (1998)
4. Busatto, G.F., Garrido, G.E.J., Almeida, O.P., Castro, C.C., Camargo, C.H.P., Cid, C.G., Buchpiguel, C.A., Furuie, S., Bottino, C.M.: A voxel-based morphometry study of temporal lobe gray matter reductions in alzheimer's disease. Neurobiology of Aging 24(2), 221–231 (2003)
5. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
6. Davatzikos, C., Fan, Y., Wu, X., Shen, D., Resnick, S.M.: Detection of prodromal alzheimer's disease via pattern classification of magnetic resonance imaging. Neurobiology of Aging 29(4), 514–523 (2008)
7. Fan, Y., Shen, D., Davatzikos, C.: Classification of Structural Images via High-Dimensional Image Warping, Robust Feature Extraction, and SVM, pp. 1–8 (2005)
8. Fotenos, A.F., Snyder, A.Z., Girton, L.E., Morris, J.C., Buckner, R.L.: Normative estimates of cross-sectional and longitudinal brain volume decline in aging and AD. Neurology 64(6), 1032–1039 (2005)
9. Frisoni, G.B., Testa, C., Zorzan, A., Sabattoli, F., Beltramello, A., Soininen, H., Laakso, M.P.: Detection of grey matter loss in mild alzheimer's disease with voxel based morphometry. Journal of Neurology, Neurosurgery & Psychiatry 73(6), 657–664 (2002)
10. Kloppel, S., Stonnington, C.M., Chu, C., Draganski, B., Scahill, R.I., Rohrer, J.D., Fox, N.C., Jack Jr., C.R., Ashburner, J., Frackowiak, R.S.J.: Automatic classification of MR scans in alzheimer's disease. Brain 131(3), 681 (2008)
11. Lao, Z., Shen, D., Xue, Z., Karacali, B., Resnick, S.M., Davatzikos, C.: Morphological classification of brains via high-dimensional shape transformations and machine learning methods. Neuroimage 21(1), 46–57 (2004)
12. Liu, Y., Teverovskiy, L., Carmichael, O., Kikinis, R., Shenton, M., Carter, C.S., Stenger, V.A., Davis, S., Aizenstein, H., Becker, J.T.: Discriminative MR image feature analysis for automatic schizophrenia and alzheimer's disease classification. LNCS, pp. 393–401. Springer, Heidelberg (2004)
13. Marcus, D.S., Wang, T.H., Parker, J., Csernansky, J.G., Morris, J.C., Buckner, R.L.: Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. Journal of Cognitive Neuroscience 19(9), 1498–1507 (2007) PMID: 17714011

14. Ramirez, J., Gorriz, J.M., Lopez, M., Salas-Gonzalez, D., Alvarez, I., Segovia, F., Puntonet, C.G.: Early detection of the alzheimer disease combining feature selection and kernel machines. In: 15th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Assembly (ICONIP 2008) (2008)
15. Salas-Gonzalez, D., Gorriz, J.M., Ramirez, J., Lopez, M., Alvarez, I., Segovia, F., Puntonet, C.G.: Computer aided diagnosis of alzheimer disease using support vector machines and classification trees. In: 15th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Assembly (ICONIP 2008) (2008)
16. Scahill, R.I., Schott, J.M., Stevens, J.M., Rossor, M.N., Fox, N.C.: Mapping the evolution of regional atrophy in alzheimer's disease: Unbiased analysis of fluid-registered serial MRI. Proceedings of the National Academy of Sciences 99(7), 4703 (2002)
17. Vapnik, V.N.: Statistical Learning Theory. Wiley Interscience, Hoboken (1998)

Selecting Regions of Interest for the Diagnosis of Alzheimer's Disease in Brain SPECT Images Using Welch's t-Test

D. Salas-Gonzalez[1], J.M. Górriz[1], J. Ramírez[1],
M. López[1], I. Álvarez[1], F. Segovia[1], and C.G. Puntonet[2]

[1] Dept. of Signal Theory, Networking and Communications, University of Granada,
18071 Granada, Spain
`dsalas@ugr.es, gorriz@ugr.es, javierrp@ugr.es`
[2] Dept. of Computer Architecture and Computer Technology, University of Granada,
18071, Granada, Spain
`carlos@atc.ugr.es`

Abstract. This paper presents a computer-aided diagnosis technique for the diagnosis of Alzheimer type dementia. The proposed methodology is based on the selection of voxels which present a Welch's t-test between both classes, Normal and Alzheimer images, greater than a given value. The mean and standard deviation of intensity values are calculated for selected voxels. They are chosen as feature vectors for two different classifiers: support vector machines with linear kernel and classification trees. The proposed methodology reaches an accuracy greater than 98% in the classification task.

1 Introduction

Distinguishing Alzheimer's disease remains a diagnostic challenge specially during the early stage of the disease. Furthermore, in this early stage, the disease offers better opportunities to be treated.

Single photon emission computed tomography (SPECT) is a widely used technique to study the functional properties of the brain. SPECT brain imaging techniques employ radioisotopes which decay emitting predominantly a single gamma photon. When the nucleus of a radioisotope disintegrates, a gamma photon is emitted with a random direction which is uniformly distributed in the sphere surrounding the nucleus. If the photon is unimpeded by a collision with electrons or other particles within the body, its trajectory will be a straight line or ray. In order for a photon detector external to the patient to discriminate the direction that a ray is incident from, a physical collimation is required. Typically, lead collimator plates are placed prior to the the detectors crystal in such a manner that the photons incident from all but a single direction are blocked by the plates. This guarantees that only photons incident from the desired direction will strike the photon detector. SPECT has become an important diagnostic and research tool in nuclear medicine.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 965–972, 2009.
© Springer-Verlag Berlin Heidelberg 2009

In SPECT images, the differences between different brains make necessary the normalization of the images with respect to a reference template. The general affine model, with 12 parameters, is usually chosen as a first step in a normalization algorithm before to proceed with a more complex non-rigid spatial transformation model [1, 2].

This paper shows a computer aided diagnosis system for the early detection of Alzheimer Type Dementia (ATD) using Support Vector Machines (SVM) and classification trees (CT). SVM is a powerful tool which has been recently focused the attention for the classification of tomography brain images [3, 4, 5]. In this work, we compare the performance of both classifiers, SVM and CT, using the accuracy rate, sensitivity and specificity.

In the proposed methodology, Welch's t-test is performed in each voxel of the brain image for the mean Normal and mean Alzheimer's disease images. The set of voxels which presents greater values than different thresholds are selected. The mean and standard deviation of the selected voxels are used as features of the classifiers.

This work is organised as follows: in Section 2 we present an overview of the support vector machines and the classification trees; in Section 3, the SPECT image acquisition and image preprocessing steps are explained; we discuss the construction of the feature vector in Section 4; in Section 5, we summarize the classification performance obtained for linear support vector machine and classification trees; and, finally, the conclusions are drawn in Section 6.

2 Overview of the Classifiers

The images we work with belong to two different classes: normal and Alzheimer type dementia (ATD). The goal of the classification task is to separate a set of binary labelled training data consisting of, in the general case, N-dimensional patterns $\mathbf{v}_i$ and class labels y_i:

$$(\mathbf{v}_1, y_1), (\mathbf{v}_2, y_2), ..., (\mathbf{v}_l, y_l) \in (R^N \times \{\text{Normal}, \text{ATD}\}), \tag{1}$$

so that a classifier is produced which maps an object $\mathbf{v}_i$ to its classification label y_i. This classifier will correctly classify new examples $(\mathbf{v}, \mathbf{y})$.

2.1 Support Vector Machines with Linear Kernels

Linear discriminant functions define decision hypersurfaces or hyperplanes in a multidimensional feature space:

$$g(\mathbf{v}) = \mathbf{w}^T \mathbf{v} + w_0 = 0 \tag{2}$$

where $\mathbf{w}$ is the weight vector and w_0 is the threshold. $\mathbf{w}$ is orthogonal to the decision hyperplane. The goal is to find the unknown parameters $w_i, i = 1, ..., N$ which define the decision hyperplane [6].

Let $\mathbf{v}_i$, $i = 1, 2, ..., l$ be the feature vectors of the training set. These belong to two different classes, ω_1 or ω_2. If the classes are linearly separable, the objective is to design a hyperplane that classifies correctly all the training vectors. This hyperplane is not unique and it can be estimated maximizing the performance of the classifier, that is, the ability of the classifier to operate satisfactorily with new data. The maximal margin of separation between both classes is a useful design criterion. Since the distance from a point $\mathbf{v}$ to the hyperplane is given by $z = |g(\mathbf{x})|/ \parallel \mathbf{w} \parallel$, the optimization problem can be reduced to the maximization of the margin $2/ \parallel \mathbf{w} \parallel$ with constraints by scaling $\mathbf{w}$ and w_0 so that the value of $g(\mathbf{v})$ is $+1$ for the nearest point in w_1 and -1 for the nearest point in w_2. The constraints are the following:

$$\mathbf{w}^T\mathbf{v} + w_0 \geq 1, \forall \mathbf{v} \in w_1 \tag{3}$$

$$\mathbf{w}^T\mathbf{v} + w_0 \leq 1, \forall \mathbf{v} \in w_2, \tag{4}$$

or, equivalently, minimizing the cost function $J(\mathbf{w}) = 1/2\|\mathbf{w}\|^2$ subject to:

$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1, i = 1, 2, ..., l. \tag{5}$$

2.2 Classification Trees

Classification trees is a non-parametric technique that produces classification of categorical dependent variables [7]. Binary tree structured classifiers are constructed by repeated splits of subsets of X into two descendant subsets, beginning with X itself. This process is plotted in Figure 1 for a two classes tree. In the figure, X_2 and X_3 are disjoint with $X = X_2 \cup X_3$. Similarly, X_4 and X_5 are disjoint with $X_4 \cup X_5 = X_2$. Those subsets which are not split, in this case X_4, X_5, X_6 and X_7 are called terminal nodes. Each terminal node is designated by a class label. There may be more than one terminal subset with the same class label.

The first problem in tree construction is how to use the learning sample to determine the binary splits of X into smaller pieces. In order to build a classification tree, three questions need to be solved: how to select the splits, when to declare a node terminal or split and how to assign a class to each terminal node. Once a good split of X is found, then a search is made for good splits of each of the two descendant nodes.

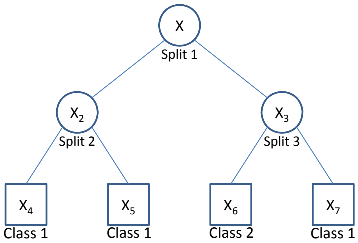


Fig. 1. Hypothetical two-class tree

3 SPECT Image Acquisition and Preprocessing

The patients were injected with a gamma emitting ^{99m}Tc-ECD radiopharmaceutical and the SPECT raw data was acquired by a three head gamma camera Picker Prism 3000. A total of 180 projections were taken for each patient with a 2-degree angular resolution. The images of the brain cross sections were reconstructed from the projection data using the filtered backprojection (FBP) algorithm in combination with a Butterworth noise removal filter [8, 9].

The complexity of brain structures and the differences between brains of different subjects make necessary the normalization of the images with respect to a common template. This ensures that the voxels in different images refer to the same anatomical positions in the brain. In this work, the images have been normalized using a general affine model, with 12 parameters [1, 10].

After the affine normalization, the resulting image is registered using a more complex non-rigid spatial transformation model. The deformations are parameterized by a linear combination of the lowest-frequency components of the three-dimensional cosine transform bases [11]. A small-deformation approach is used, and regularization is by the bending energy of the displacement field. Then, we normalize the intensities of the SPECT images with respect to the maximum intensity, which is computed for each image individually by averaging over the 3% of the highest voxel intensities, similar as in [12].

4 Feature Selection

We study the main differences between normal and Alzheimer type dementia images. Voxels which provide higher difference between both groups will be considered as training vectors of two classifiers: support vector machines with linear kernel and classification trees. Thus, this statistical study will allow us to select discriminant regions of the brain to establish whether a given SPECT image belongs to a Normal or an ATD patient.

4.1 Mean Image

Firstly, we study the mean intensity values of the Normals and ATD images. Let the brain image set be $I_1, I_2, ..., I_N$, where the number of images N is the sum of the images previously labelled as Normals (N_{NOR}) and Alzheimer type dementia (N_{ATD}) by expertises. Hence, the average Normal brain image of the dataset is defined as

$$\bar{I}_{NOR} = \frac{1}{N_{NOR}} \sum_{j \in NOR}^{N_{NOR}} I_j. \tag{6}$$

The average ATD can be calculated analogously:

$$\bar{I}_{ATD} = \frac{1}{N_{ATD}} \sum_{j \in ATD}^{N_{ATD}} I_j. \tag{7}$$

4.2 Standard Deviation Image

The root-mean-square deviation of the images from their mean for normal images is defined as

$$I^{\sigma}_{NOR} = \sqrt{\frac{1}{N_{NOR}} \sum_{j \in NOR}^{N_{NOR}} (I_j - \overline{I}_{NOR})^2}, \tag{8}$$

and for ATD:

$$I^{\sigma}_{ATD} = \sqrt{\frac{1}{N_{ATD}} \sum_{j \in ATD}^{N_{ATD}} (I_j - \overline{I}_{ATD})^2}. \tag{9}$$

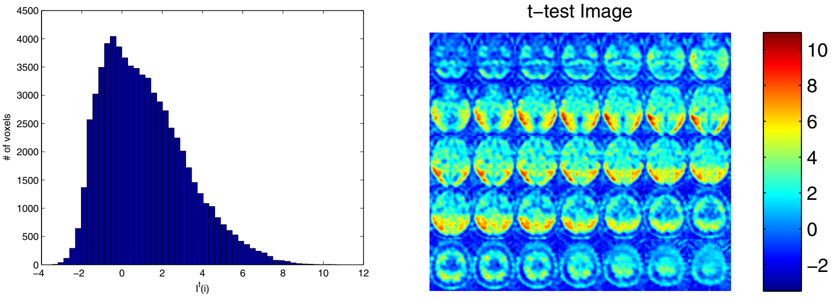
4.3 Welch's t-Test

Welch's t-test for independent samples is a modification of the t-test that it does not assume equal population variances. Using the information provided by the mean and standard deviation normal and ATD images, the t-test can be calculated for each voxel by:

$$I^t = \frac{\overline{I}_{NOR} - \overline{I}_{ATD}}{\sqrt{\frac{I^{\sigma}_{NOR}}{N_{NOR}} + \frac{I^{\sigma}_{ATD}}{N_{ATD}}}}, \tag{10}$$

where I^t denotes the resulting image with the value given by Welch's t-test calculated in each voxel. Figure 2 depicts this image I^t. The histogram with the distribution of I^t is plotted in Figure 2(b).

Welch's t-test gives us a value which allows us to measure the voxel intensity difference between the mean normal ($\overline{I}_{NOR}$) and mean ATD image ($\overline{I}_{ATD}$). Those voxels which present a t-test value greater than a given threshold will be selected for the classification task.



(a) (b)

Fig. 2. (a) Histogram representing the distribution of the t-values for the image $I^t(i)$. (b) Welch's t-test image I^t.

5 Results

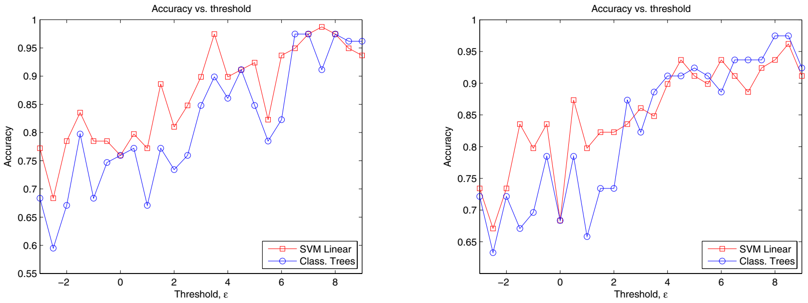
The performance of the classification is tested on a set of 79 real SPECT images (41 normals and 38 ATD) of a current study using the leave one-out method: the classifier is trained with all but one images of the database. The remaining image, which is not used to define the classifier, is then categorized. In that way, all SPECT images are classified and the success rate is computed from the number of correctly classified subjects.

First of all, we select those voxels i which fulfill the following condition:

$$i \quad / \quad \{I^t(i) > \varepsilon\}, \tag{11}$$

where ε is a threshold. We choose 25 threshold values equally spaced from -3 to 9. The number of selected voxels decreases as ε increases. Increasing ε allows us to select those voxels which present greater difference between the mean normal and ATD images according to Welch's t-test. Two different features will be used: on the one hand, mean of selected voxels and, on the other hand, a vector with two components: mean and standard deviation of selected voxels.

The classification performance obtained using support vector machines with linear kernel and classification trees versus the threshold value ε is plotted in Figure 3. As it was expected, the classification accuracy is lower for small threshold values ε and increases concomitantly with ε. Using both set of features, it is easily seen that support vector machines present, in general, a higher correct rate than classification trees. On the other hand, more stable results are obtained using mean and standard deviation as features. For instance, this fact is remarked in the case $\varepsilon = 0.55$ where using only the mean as feature, an accuracy rate of 82.3% and 78.5% was obtained using SVM and CT respectively. The best accuracy rate (98.7%) was obtained when the the mean of selected voxels was used as feature vector, the threshold was set to $\varepsilon = 7.5\%$ and classification was performed using support vector machines.



(a) (b)

Fig. 3. Accuracy rate versus threshold value ε. Feature vector: (a) Mean of selected voxels. (b) Mean and standard deviation of selected voxels.

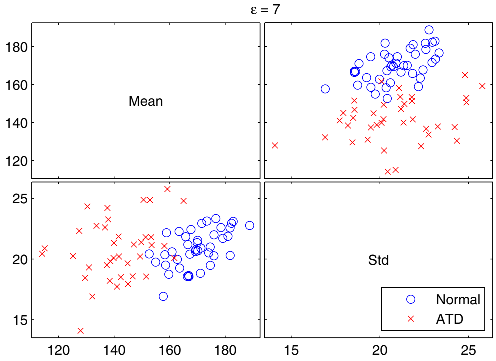


Fig. 4. Matrix plot $\varepsilon = 7$

As it was expected, higher accuracy rates are obtained for higher values of ε. This is due to the fact that feature vectors separate better both classes (normal and ATD) as the threshold value increases. In order to show the pairwise relationship between the mean and standard deviation of selected voxels for the case $\varepsilon = 7$, a matrix plots is drawn in Figure 4. This figure shows that the selected features are separated for Normals and ADT images when $\varepsilon = 7$.

6 Conclusion

In this work, a computer aided diagnosis for the diagnosis of Alzheimer type dementia (ATD) is presented. Initially, the images were labelled by experts as Normals or Alzheimer's dementia. After normalisation of the brain images, we calculate the mean normal and mean ATD image. Welch's t-test between both images is calculated in each voxel. The resulting statistic t give us a procedure to ranking voxels. Voxels with t greater than different threshold values are selected. The mean and standard deviation of selected voxels are used as feature vectors for two different classifiers: a support vector machine with linear kernel and a classification tree. The accuracy, sensitivity and specificity of both classifiers are analyzed using several threshold values. The proposed methodology reaches an accuracy of 98 % in the classification task. The method proposed in this work allows us to classify the brain images in normal and affected subjects in a parsimonious way, with no prior knowledge about the Alzheimer's disease.

Acknowledgment

This work was partly supported by the Spanish Government under the PETRI DENCLASES (PET2006-0253), TEC2008-02113, NAPOLEON (TEC2007-68030-C02-01) projects and the Consejera de Innovación, Ciencia y Empresa (Junta de Andaluca, Spain) under the Excellence Project (TIC-02566).

References

[1] Salas-Gonzalez, D., Górriz, J.M., Ramírez, J., Lassl, A., Puntonet, C.G.: Improved gauss-newton optimization methods in affine registration of spect brain images. IET Electronics Letters 44(22), 1291–1292 (2008)

[2] Friston, K., Ashburner, J., Kiebel, S., Nichols, T., Penny, W. (eds.): Statistical Parametric Mapping: The Analysis of Functional Brain Images. Academic Press, London (2007)

[3] Ramírez, J., Górriz, J.M., Romero, A., Lassl, A., Salas-Gonzalez, D., López, M., Gómez-Río, M., Rodríguez, A.: Computer aided diagnosis of alzheimer type dementia combining support vector machines and discriminant set of features. Accepted in Information Sciences (2008)

[4] Górriz, J.M., Ramírez, J., Lassl, A., Salas-Gonzalez, D., Lang, E.W., Puntonet, C.G., Álvarez, I., López, M., Gómez-Río, M.: Automatic computer aided diagnosis tool using component-based svm. In: Medical Imaging Conference. IEEE, Dresden, Germany (2008)

[5] Fung, G., Stoeckel, J.: SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information. Knowledge and Information Systems 11(2), 243–258 (2007)

[6] Vapnik, V.: Statistical learning theory. John Wiley and Sons, Chichester (1998)

[7] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Chapman and Hall, Boca Raton (1993)

[8] Bruyant, P.P.: Analytic and iterative reconstruction algorithms in spect. The Journal of Nuclear Medicine 43(10), 1343–1358 (2002)

[9] Ramírez, J., Górriz, J.M., Gómez-Río, M., Romero, A., Chaves, R., Lassl, A., Rodríguez, A., Puntonet, C.G., Theis, F., Lang, E.: Effective emission tomography image reconstruction algorithms for spect data. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 741–748. Springer, Heidelberg (2008)

[10] Woods, R.P., Grafton, S.T., Holmes, C.J., Cherry, S.R., Mazziotta, J.C.: Automated image registration: I. general methods and intrasubject, intramodality validation. Journal of Computer Assisted Tomography 22(1), 139–152 (1998)

[11] Ashburner, J., Friston, K.J.: Nonlinear spatial normalization using basis functions. Human Brain Mapping 7(4), 254–266 (1999)

[12] Saxena, P., Pavel, D.G., Quintana, J.C., Horwitz, B.: An automatic threshold-based scaling method for enhancing the usefulness of Tc-HMPAO SPECT in the diagnosis of Alzheimer's disease. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 623–630. Springer, Heidelberg (1998)

Alzheimer's Diagnosis Using Eigenbrains and Support Vector Machines

I. Álvarez[1], J.M. Górriz[1], J. Ramírez[1], D. Salas-Gonzalez[1], M. López[1],
F. Segovia[1], C.G. Puntonet[2], and B. Prieto[2]

[1] Dept. of Signal Theory, Networking and Communications
University of Granada, Spain
[2] Dept. of Computers Architecture and Technology
University of Granada, Spain

Abstract. An accurate and early diagnosis of the Alzheimer's Disease (AD) is of fundamental importance for the patients medical treatment. Single Photon Emission Computed Tomography (SPECT) images are commonly used by physicians to assist the diagnosis, rating them by visual evaluations. In this work we present a computer assisted diagnosis tool based on a Principal Component Analysis (PCA) dimensional reduction of the feature space approach and a Support Vector Machine (SVM) classification method for improving the AD diagnosis accuracy by means of SPECT images. The most relevant image features were selected under a PCA compression, which diagonalizes the covariance matrix, and the extracted information was used to train a SVM classifier which could classify new subjects in an unsupervised manner.

1 Introduction

Distinguishing AD from other causes of dementia still remains a diagnostic challenge, specially during the early stage of the disease, that offers better opportunities to treat its symptoms. Thus, an accurate and early diagnosis of the AD by means of non-invasive methods, is of fundamental importance for the patients medical treatment. Nuclear imaging as Single Photon Emission Computed Tomography (SPECT) or Positron Emission Tomography (PET) are examples of non-invasive, three-dimensional functional imaging modalities that provide clinical information regarding biochemical and physiologic processes in patients, and are frequently used as a diagnostic tool in addition to the clinical findings.

The examination of the predictive abilities of nuclear imaging with respect to AD and other dementia illnesses is usually done through visual assessments performed by experts[1, 2]. However, statistical classification methods have not been widely used to assist the diagnosis, being Statistical Parametric Mapping (SPM) the most extended tool in the neuro-imaging community[3, 4]. It consists of doing a voxel-wise statistical test, comparing the values of the image under study to the mean values of the group of normal images. Subsequently the significant voxels are inferred by using Random Field Theory [5]. This method suffers

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 973–980, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the inconvenient of local and mono-variate approaches and it was not developed specifically for the typical case of a single image study, but for comparing groups of images. One can use it for diagnostics by comparing the image under study to a group of normal images. This comparison has the disadvantages of assessing a population containing just one individual and being lacking in any information about the pathology, that the other population of AD affected images contain.

On the other hand, multivariate approaches that consider as one observation all the voxels in a single scan, suffer from the *curse of dimensionality* problem. This major problem, associated with pattern recognition systems, occurs when the number of available features for designing the classifier is very large compared with the number of available training examples. The importance of multivariate approaches is that the interactions among voxels and error effects are assessed statistically, while paying the price of losing capability of making statistical inferences about regionally specific changes.

Principal Component Analysis (PCA) is an example of multivariate technique that require more training examples than features and is not suitable for making any statistical inference about the characterizations that it obtains [3]. In the Alzheimer's disease aided diagnosis, it has been used as a mathematical device that simply identify prominent patterns of correlations or functional connectivity of brain regions, to be analyzed with other statistical tools as SPM, ANOVA or MANCOVA [6, 7, 8].

Our approach to the computer aided diagnosis (CAD) involves not statistical inference but machine learning techniques, which are appropriate for single image studies. In this work we used two combined techniques that, independently have been successful in solving several classification problems. Firstly, we made use of *eigenbrains* or eigenimages, that were obtained from the Principal Components [3], to reduce the dimension of the feature space to a set of projection coefficients (see Sect. 2), in a similar fashion as in face detection [9]. This process reduced the dimensionality of the feature space from $\sim 5 \cdot 10^5$ to $\sim 10^2$, thus facing the small sample size problem. Secondly, once a significant feature ensemble was selected, we built a SVM to manage the classification task.

Support Vector Machines (SVMs) have marked the beginning of a new era in the learning from examples paradigm [10]. Recently, SVMs have attracted attention from the pattern recognition community due to a number of theoretical and computational merits derived from [10]. These techniques have been successfully applied to many fields including voice activity detection (VAD) [11], content-based image retrieval [12], and medical imaging diagnosis [13, 14]. Somehow, the application of SVM to high dimensional and small sample size problems is still a challenge and improving the accuracy SVM based-approaches is still a field in development[15, 16].

The combination of these two methods grows a CAD system for the early detection of Alzheimer Type Dementia (ATD) tested over SPECT images, and developed with the aim of reducing the subjectivity in visual interpretation of these scans by clinicians, thus improving the accuracy of diagnosing Alzheimer's disease in its early stage.

2 PCA Application to SPECT Images: Eigenbrains

Principal Component Analysis is a standard technique for extracting the most significant features from a dataset. It is based on a linear transformation acting on a zero mean dataset, that diagonalizes its covariance matrix. In brain images, the dataset is an ensemble of 3D brain images $\mathbf{\Gamma}_i$, whose size M is typically $79 \times 95 \times 69 \sim 5 \cdot 10^5$ voxels. Let the full 3D brain image set be $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, ..., \mathbf{\Gamma}_N$, each understood as a vector of dimension M. The average brain image of the dataset is defined as $\mathbf{\Gamma} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{\Gamma}_n$. We need firstly to extract the average of the image set to each one, producing a new set $\mathbf{\Phi}_i = \mathbf{\Gamma}_i - \mathbf{\Gamma}$ with $n = 1, 2, ..., N$. On this set, a PCA transformation is composed by M orthogonal vectors $\mathbf{u}_i$, such that

$$\lambda_i = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{u}_i^T \mathbf{\Phi}_n)^2 \tag{1}$$

is maximum, subject to the constrain

$$\mathbf{u}_i^T \mathbf{u}_j \ = \ \delta_{ij} \tag{2}$$

where δ_{ij} is the Kronecker delta. The resulting $\mathbf{u}_i$ and λ_i are the eigenvectors and eigenvalues respectively of the covariance matrix:

$$\mathbf{C} = \frac{1}{N}\sum_{i=n}^{N} \mathbf{\Phi}_n \mathbf{\Phi}_n^T = \mathbf{A}\mathbf{A}^T \tag{3}$$

where $\mathbf{A} = [\mathbf{\Phi}_1, ..., \mathbf{\Phi}_N]$. We will refer to this orthogonal eigenvector basis $\{\mathbf{u}_i\}, i = 1, ..., M$ as eigenbrains, because of its brain like appearance (see Fig. 1). To obtain them, it is necessary to diagonalize the $M \times M$ covariance matrix, which for brain images would be approximately a $5 \cdot 10^5 \times 5 \cdot 10^5$ matrix. The computational complexity of the diagonalization process can be significantly reduced by diagonalizing the matrix $\hat{\mathbf{C}} = \mathbf{A}^T\mathbf{A}$, whose size is $N \times N$, with $N << M$ [9]. This allows to obtain N of the M eigenvectors $\mathbf{u}_n$ of $\mathbf{C}$, from the eigenvectors $\mathbf{v}_n$ of $\hat{\mathbf{C}}$ as $\mathbf{u}_n = \mathbf{A}\mathbf{v}_n$, $n = 1, .., N$. Usually, the first few eigenbrains explain the whole variance, so only a number $M' < M$ is necessary to appropriately describe the dataset (in this case $M' = N$). The obtained eigenbrains span a new subspace which we refer to as the "eigenbrain space". For the classification task we projected each image into the previously defined eigenbrain space. Each projected image produced a vector of weights so that a matrix of weights can be constructed with the whole database. This matrix $\mathbf{\Omega}$ is given by:

$$\mathbf{\Omega}_{in} = \mathbf{u}_n^T \mathbf{\Phi}_i, \qquad n = 1, 2, ..., M', \quad i = 1, 2, ..., N \tag{4}$$

and describes the contribution of each eigenbrain in representing the input brain image $\mathbf{\Phi}_i$, treating the eigenbrains as a basis set for brain images (see Fig. 1). The matrix $\mathbf{\Omega}$ contains the most significant information extracted from the principal component analysis, stacked in a $N \times M'$ data ensemble. We used this matrix $\mathbf{\Omega}$ for the following classification task, as N M'-dimensional patterns:

$$\mathrm{x}_i = [\mathbf{\Omega}_{i1}, \mathbf{\Omega}_{i2}, ..., \mathbf{\Omega}_{iM'}], \quad i = 1, 2, ..., N \tag{5}$$

each of them with its corresponding class label $y_i \in \{\pm 1\}$.

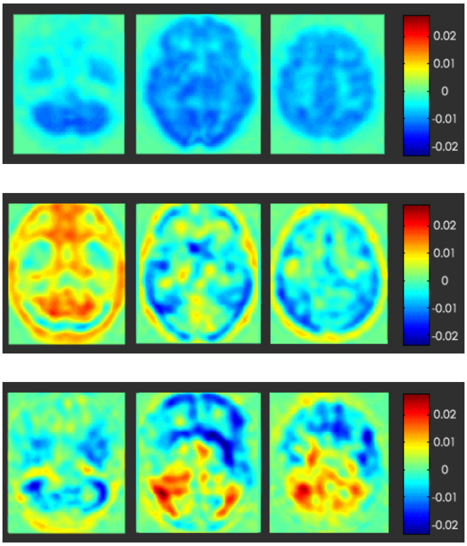


Fig. 1. Three representative transversal slices of the first three eigenbrains, ranked by its eigenvalue

3 Background on SVMs

The classification is achieved through a SVM, that separates a given set of binary labeled training data with a hyperplane that is maximally distant from the two classes (known as the maximal margin hyper-plane). The objective is to build a function $f : \mathbb{R}^{M'} \longrightarrow \{\pm 1\}$ using training data, consisting of M'-dimensional patterns x_i and class labels y_i:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N) \in \left(\mathbb{R}^{M'} \times \{\pm 1\}\right), \tag{6}$$

so that f will correctly classify new examples $(\mathbf{x}, y)$. When no linear separation of the training data is possible, SVM can work effectively in combination with kernel techniques using the *kernel trick*, so that the hyperplane defining the SVM corresponds to a non-linear decision boundary in the input space [10]. In this way the decision function f can be expressed in terms of the *support vectors* only [10]:

$$f(\mathbf{x}) = \mathrm{sign}\{\sum_{i=1}^{N_S} \alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + w_0\}, \tag{7}$$

where $K(.,.)$ is the kernel function, α_i is a weight constant derived form the SVM process and $\mathbf{s}_i$ are the support vectors [10]. Common kernels that are used by SVM practitioners for the nonlinear feature mapping are:

- Polynomial

$$K(\boldsymbol{x}, \boldsymbol{y}) = [\gamma(\boldsymbol{x} \cdot \boldsymbol{y}) + c]^d. \tag{8}$$

- Radial basis function (RBF)

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{y}||^2). \tag{9}$$

as well as the linear kernel, in which $K(.,.)$ is simply a scalar product.

4 Experiments

The database consists of a set of 3D SPECT brain images produced with an injected gamma emitting ^{99m}Tc-ECD radio-pharmaceutical and acquired by a three-head gamma camera Picker Prism 3000. Images of the brain cross sections are reconstructed from the projection data using the filtered back-projection (FBP) algorithm in combination with a Butter-worth noise removal filter. The SPECT images are spatially normalized using the SPM software [3] in order to ensure that the voxels in different images refer to the same anatomical positions in the brain, in a process described in detail in [13, 17]. The images were initially labeled by experienced clinicians of the "Virgen de las Nieves" hospital (Granada, Spain), within two classes: NORMAL and AD. In total, the database consists of 79 patients: 41 NOR and 38 AD.

The size of the images was reduced by a factor $1/n^3$, with n ranging from 2 to 12, in order to reduce the effect of possible defective acquisitions of some brain regions and to simplify the computation of the eigenbrains. In average, only 76 eigenbrains were necessary to explain near the 100% of the variance, with about the 60% retained by the first eigenbrain. These 76 eigenbrains were taken in groups of $M' = 1, 2, ...$ to 76 items to construct the matrix of (4), grouping them according to:

- their covariance eigenvalues.
- selecting those which made the Fisher Discriminant Ratio (FDR) between the $\boldsymbol{\Omega}$ values higher.

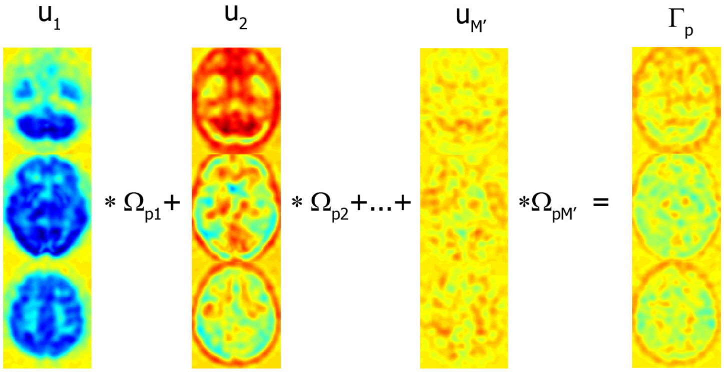


Fig. 2. Eigenbrain representation of three representative transversal slices of the p-th zero-mean brain image. The representation is encoded in the coefficients $(\Omega_{p1}, \Omega_{p2}, ..., \Omega_{pM'})$.

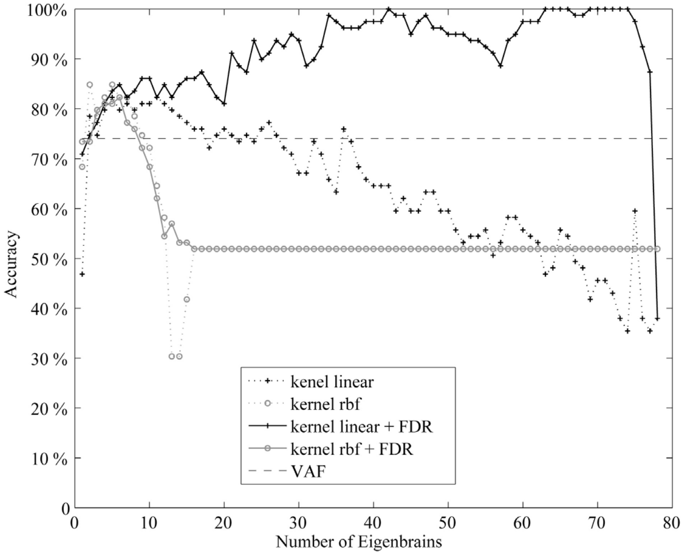


Fig. 3. Two different kernels accuracies versus the number of eigenbrains used to construct the matrix Ω for the two methods of selecting components

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{10}$$

where μ_i and σ_i^2 denote the variance and the mean of the i-th class respectively. Once the matrix $\mathbf{\Omega}$ was obtained, a SVM was trained using 4 different kernels: linear, quadratic, RBF and polynomial, and was tested using a leave-one-out cross-validation strategy.

Results: Linear kernels are expected to perform better than others when the dimension of the feature space increases. If this increase is made with a covariance eigenvalue criterion, the relevant information is contained in the few first eigenbrains, and the rest only contributes with noisy information, as seen from Fig. 3. But enhancing the dimension of the feature space with a FDR criterion shows a fundamental increase of the effectiveness, if compared to ranked covariance eigenvalues which allows to conclude that only a small fraction of the eigenbrains contains noisy information. When a linear kernel SVM was built in combination with a large dimensional feature space, independently of the reducing factor, the method reaches notably the 100% accuray. This outperforms previous results, as the 74% accuracy Voxel-as-Features approach [18] [19].

5 Conclusions

Using statistical classification methods in SPECT images for assisting the AD diagnosis looks promising, but still faces problems to become a useful tool to physicians. Two main problems are the high dimensionality of the feature space and the small sample size problem. In this work, we presented two solutions to each problem: we made use of the eigenbrain approach, inspired in the eigenface solution to the face detection problem, leading to the idea of collecting a small feature pattern which best describes the patient image characteristics. Secondly, we trained a SVM supervised learning classifier, which allowed us to automatically separate the patient database in normal and affected subjects. Furthermore, a classification of new subjects in a unsupervised manner was possible, without any concrete knowledge about the Alzheimer's disease. A significant advance is seen when estimating the performance of this learning process with a leave-one-out cross-validation test, reaching the 100% accuracy. PCA is then a simple and effective method to deal with the high dimensionality problem in SPECT images with a low computational cost, which shows its effectiveness in combination with a linear kernel SVM and high dimensional feature space, provided that the dimension is enhanced with a FDR criterion.

Acknowledgment

This work was partly supported by the MICINN under the PETRI DENCLASES (PET2006-0253), TEC2008-02113, NAPOLEON (TEC2007-68030-C02-01) and HD2008-0029 projects and the Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía, Spain) under the Excellence Project TIC-02566.

References

[1] Cummings, J.L., Vinters, H.V., Cole, G.M., Khachaturian, Z.S.: Alzheimer's disease: etiologies, pathophysiology, cognitive reserve, and treatment opportunities. Neurology 51(suppl. 1), S2–S17 (1998)

[2] Ng, S., Villemagne, V.L., Berlangieri, S., Lee1, S.-T., Cherk, M., Gong, S.J., Ackermann, U., Saunder, T., Tochon-Danguy, H., Jones, G., Smith, C., O'Keefe, G., Masters, C.L., Rowe, C.C.: Visual assessment versus quantitative assessment of 11c-pib pet and 18f-fdg pet for detection of alzheimer's disease. Journal of Nuclear Medicine 48, 547–552 (2007)

[3] Friston, K.J., Ashburner, J., Kiebel, S.J., Nichols, T.E., Penny, W.D.: Statistical Parametric Mapping: The Analysis of Functional Brain Images. Academic Press, London (2007)

[4] Yin, T.K., Chiu, N.T.: Discrimination between alzheimer's dementia and controls by automated analysis of statistical parametric maps of 99mTc-HMPAO-SPECT volumes. In: Proceedings of the Fourth IEEE Symposium on Bioinformatics and Bioengineering, pp. 183–190 (2004)

[5] Adler, R.J.: The Geometry of random fields. Wiley, New York (1981)

[6] Scarmeas, N., Habeck, C.G., Zarahn, E., Anderson, K.E., Park, A., Hilton, J., Pelton, G.H., Tabert, M.H., Honig, L.S., Moeller, J.R., Devanand, D.P., Stern, Y.: Covariance pet patterns in early alzheimer's disease and subjects with cognitive impairment but no dementia: utility in group discrimination and correlations with functional performance. NeuroImage 23(1), 35–45 (2004)

[7] Salmon, E., Kerrouche, N., Perani, D., Lekeu, F., Holthoff, V., Beuthien-Baumann, B., Sorbi, S., Lemaire, C., Collette, F., Herholz, K.: On the multivariate nature of brain metabolic impairment in alzheimer's disease. Neurobiology of Aging 30(2), 186–197 (2009)

[8] Nobili, F., Salmaso, D., Morbelli, S., Girtler, N., Piccardo, A., Brugnolo, A., Dessi, B., Larsson, S.A., Rodriguez, G., Pagani, M.: Principal component analysis of fdg pet in amnestic mci. Eur. J. Nucl. Med. Mol. Imaging 35(12), 2191–2202 (2008)

[9] Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of congnitive neuroscience 3(1), 71–86 (1991)

[10] Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, Inc., New York (1998)

[11] Ramírez, J., Yélamos, P., Górriz, J.M., Segura, J.C.: SVM-based speech endpoint detection using contextual speech features. Electronics Letters 42(7), 877–879 (2006)

[12] Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(7), 1088–1099 (2006)

[13] Ramírez, J., Górriz, J.M., Gómez-Río, M., Romero, A., Chaves, R., Lassl, A., Rodríguez, A., Puntonet, C.G., Theis, F., Lang, E.: Effective emission tomography image reconstruction algorithms for SPECT data. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 741–748. Springer, Heidelberg (2008)

[14] Ramírez, J., Górriz, J.M., Romero, A., Lassl, A., Salas-Gonzalez, D., López, M., Alvarez, I., Gómez-Río, M., Rodríguez, A.: Computer aided diagnosis of alzheimer type dementia combining support vector machines and discriminant set of features. In: Information Sciences (2008) (accepted)

[15] Lassl, A., Górriz, J.M., Ramírez, J., Salas-Gonzalez, D., Puntonet, C.G., Lang, E.W.: Clustering approach for the classification of spect images. In: Medical Imaging Conference, Dresden. IEEE, Los Alamitos (2008)

[16] Górriz, J.M., Ramírez, J., Lassl, A., Salas-Gonzalez, D., Lang, E.W., Puntonet, C.G., Álvarez, I., López, M., Gómez-Río, M.: Automatic computer aided diagnosis tool using component-based svm. In: Medical Imaging Conference, Dresden. IEEE, Los Alamitos (2008)

[17] Salas-Gonzalez, D., Górriz, J.M., Ramírez, J., Lassl, A., Puntonet, C.G.: Improved gauss-newton optimization methods in affine registration of spect brain images. IET Electronics Letters 44(22), 1291–1292 (2008)

[18] Fung, G., Stoeckel, J.: SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information. Knowledge and Information Systems 11(2), 243–258 (2007)

[19] Stoeckel, J., Malandain, G., Migneco, O., Koulibaly, P.M., Robert, P., Ayache, N., Darcourt, J.: Classification of SPECT images of normal subjects versus images of alzheimer's disease patients. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 666–674. Springer, Heidelberg (2001)

Artificial Intelligent Systems Based on Supervised HUMANN for Differential Diagnosis of Cognitive Impairment: Towards a 4P-HCDS

Patricio García Báez[1], Miguel Angel Pérez del Pino[2], Carlos Fernández Viadero[3], and Carmen Paz Suárez Araujo[2]

[1] Departamento de Estadística, Investigación Operativa y Computación,
Universidad de La Laguna. 38271 La Laguna, Canary Islands, Spain
[2] Instituto Universitario de Ciencias y Tecnologías Cibernéticas, Universidad de Las Palmas de Gran Canaria. 35017 Las Palmas de Gran Canaria, Canary Islands, Spain
[3] Unidad de Atención a la Dependencia de Santander, Gobierno de Cantabria,
39012 Santander, Cantabria, Spain
`pgarcia@ull.es, cpsuarez@dis.ulpgc.es`

Abstract. Differential and early diagnosis of cognitive impairment (CI) continues being one of the crucial points to which clinical medicine faces at every level of attention, and a significant public health concern. This work proposes new CI diagnostic tools based on a data fusion scheme, artificial neural networks and ensemble systems. Concretely we have designed a supervised HUMANN [1] with capacity of missing data processing (HUMANN-S) and a HUMANN-S ensemble system. These intelligent diagnostic systems are inside EDEVITALZH, a clinical virtual environment to assist the diagnosis and prognosis of CI, Alzheimer's disease and other dementias. Our proposal is a personalized, predictive, preventive, and participatory-healthcare delivery system (4P-HCDS) and is an optimal solution for an e-health framework. We explore their ability presenting preliminary results on differential diagnosis of CI using neuropsychological tests from 267 consultations on 30 patients by the Alzheimer's Patient Association of Gran Canaria.

Keywords: Cognitive Impairment; Dementia; Diagnosis; Alzheimer's Disease; Missing Data; *k*-fold Cross-validation, Neural Network Ensemble; HUMANN.

1 Introduction

Many sociological studies point out the world's population ages rapidly, primarily in developed countries. In Spain, the percentage of elder age population is 16,66%, with a progressive increase of over-aging of 4,52% and with a rising tendency. The neurological disorders that cause gradual loss of cognitive function, are the diseases with more interest in the aging.

This special interest is caused by its high prevalence, 5-10% in older than 65 years and more than 25%-30% in the very old ones, [2][3], when it is assessed CI in the elderly as an early stage of dementia, and because of the important repercussions that

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 981–988, 2009.
© Springer-Verlag Berlin Heidelberg 2009

originate in the scope of the patient, in the familiar, social and sanitary spheres [4]. Therefore, the differential diagnosis, as well as the early one, of CI are fundamental and continue to be one of the crucial points to which clinical medicine faces at every level of attention.

Three types of CI evaluation are possible. The first uses an open form, another is semi structured, and the third evaluation is carried out using the application of a series of cognitive evaluation scales. Several neuropsychological tests [5] have been developed and used to evaluate different cognitive sections of a patient throughout the years. It is not a simple task to define a clear relationship between test results and specific symptoms or different levels of CI. Other important problems in the use of these scales are the absence of universal cut points, trans-cultural difficulties and the level of precision that seems to be similar with the use of short or long scales [6].

All these handicaps, the high uncertainty diagnosis [7], and the degree of the underdiagnosis which is so crucial and can reach 95% of the cases in some settings [8], highlight the need to develop new and alternative methods and instruments of diagnosis, making special emphasis on early and differential diagnosis, and causing its use in all the sanitary scopes, in the specialized and the primary attention care.

Especially computer-intensive algorithms based on “ensemble learning”-methods that generate many classifiers and aggregate their results, are being developed in the last years in regard of CI and Alzheimer´s disease (AD) classification using as input essentially EEG abnormalities associated with the analyzed pathology [9][2]. We have previously developed Counterpropagation based multi-net systems to diagnose neuro decline processes related to aging [10].

We propose new diagnostic tools based on a data fusion scheme using artificial neural networks and ensemble systems. Concretely we have designed two HUMANN-S based systems, where HUMANN-S is the supervised version of HUMANN architecture, with capacity of missing data processing. These systems are, a HUMANN-S diagnosis system, and a HUMANN-S ensemble system. HUMANN is a hierarchical, feed-forward modular neural network, with three operational modules, SOM, Tolerance, and Labelling [1]. The modular organization allows to develop a dynamic learning framework and provides the possibility to introduce specific behavior in each module. We used this capacity for designing HUMANN-S introducing a supervised labelling module, applying a perceptron-type neural network.

In this paper we explore the ability of a simple HUMANN-S and a HUMANN-S ensemble system, which was combined with simple and weight majority vote strategies. We present our preliminary results on differential diagnosis of CI using neuropsychological tests along with these systems. We also perform a comparative study between them. The inherent complexities of the information environment used and the scope of our work, are very appropriate scenes for using this approach.

Our proposed artificial intelligent systems are inside EDEVITALZH, which is a clinical virtual environment, focused on diagnosis and prognosis of CI, AD and other dementias. It is an environment based on Telemedicine which propitiates on line interactivity between physicians and their colleagues. EDEVITALZH will also allow that any patient with dementia, or any potential patient could have an appropriate medical attention, diagnosis and treatment, independently of existing specialized dementia centres where the patient is. Our proposal is a 4P-HCDS and is an optimal solution for an e-health framework.

The proposed systems in this paper has important advantages referring to other computational solutions based on artificial neural networks: HUMANN-S can handle in an efficient way the noise, it has a strong adaptive character, it has capacity for missing data processing and for the data fusion scheme. The importance of this proposal is in the fact that the specific application we investigate is itself a significant public health concern, and that it has widespread impact on long term geriatric care.

2 EDEVITALZH

EDEVITALZH [11] is a clinical virtual environment focused on diagnosis and prognosis of CI, AD and other dementias. This environment implements the Global Clinical Protocol for Dementias (GCPD), which reflects the specific data of interest channelled towards the diagnosis of theses neuropathologies. It will allow computerized medical records to be set up, creating a data base of the patients' medical records.

EDEVITALZH allows performing analysis of the CI evolution as well as predictive analysis with the possibility of providing pre-diagnosis. This makes possible to design proper and customized therapeutic strategies for each patient as well as the validation of these strategies. EDEVITALZH belongs to what we can call the 4P-HCDS. With the use of EDEVITALZH, the diagnostic work of the physician will be more accurate, agile, and patient dedicated, allowing time and resources optimization, while it is, in a parallel way, capable to work with different patients, in different geographical locations; patients which are able to be studied by other physicians.

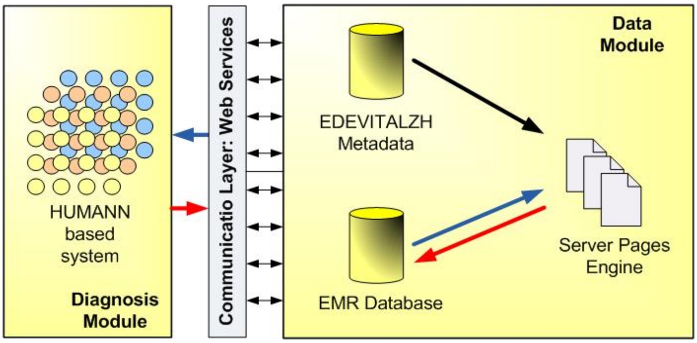


Fig. 1. EDEVITALZH Modular Architecture

EDEVITALZH has been implemented using a web application model. This allows that any computer can access the application using a web browser; and the information is centralized being always available for consultation from anywhere.

Based on a modular system concept, EDEVITALZH is made up of two first level blocks: 1) Data Module (Electronic Medical Records Database; Metadata; Server Pages Engine), and 2) Diagnosis Module (HUMANN-based system) [11]. Both modules communicate one another thanks to networking. This logical structural separation also allows a physical separation; different parts of the system can be located in different places. Furthermore, thanks to its modular character, its complexity and the cost of the system hardware and software are reduced –i.e., the same data module can operate with various diagnosis modules at the same time.

EDEVITALZH GUI (graphic user interface) also gives access and control of the HUMANN neural system, core of the intelligent system to aid the diagnosis. Both architecture modules will be connected one another using a communication layer based on Web Services, (Fig. 1) and data will be exchanged using standard XML. The area of development at a Research Level will allow the analysis and study of GCPD in depth, as well as the implementation of different configurations of HUMANN with the aim of producing precise and reliable diagnosis and obtain an optimum GCPD.

2.1 Diagnosis Systems

Two neural computing systems employing a simple-net and an ensemble approach [12] were used to approach the problem of CI differential diagnosis. The systems are based on modules that implement a supervised variant of HUMANN architecture (HUMANN-S) [1]. The first system is a HUMANN-S-based system an the second one is a HUMANN-S ensemble system, which was combined with simple and weighted majority vote strategies (SMVE/WMVE) [12].

2.1.1 Supervised HUMANN

The main part of both proposed systems is a supervised version of neural architecture HUMANN [1]. This neural network can implement the general approach of the classification process, which has three stages: a) feature extraction, b) template generation, c) discrimination (labelling), in a transparent and efficient way. Normally, and specifically in this application, step a), must be implemented by pre-processing modules that will be application-dependent. HUMANN is composed by three neural modules, SOM, Tolerance and Labelling, with a multilayer structure having different neurodynamics, connection topologies and learning laws.

The first neural module of HUMANN-S is a Kohonen´s Self-Organizing Map (SOM). This module implements a non-linear "projection" from an input space onto a two-dimensional array. This kind of neural structure is used because its main feature is the formation of topology-preserving feature maps and approximation of the input probability distribution, by means a self-organizing process which can produce feature´s detectors. For this application HUMANN-S has the capacity of missing data processing, which is implemented by means of a variant of the SOM architecture [13]. This variant prevents missing values from contributing when coming out or modifying weights. Even so, this way of approaching missing values is insufficient by itself, essentially when the proportion of missing values is excessive.

The second module is the Tolerance layer. It is the main module responsible for the robustness of HUMANN-S against noise. Its topology is a two-dimensional array which has the same dimension as the Kohonen layer and an interconnection scheme one to one with that previous layer. Its main objective is to compare the fitting between the input patterns and the Kohonen detectors. If the goodness of the fit is not sufficient the pattern is regarded as an outlier and is discarded. The weights of this layer are responsible for storing the mean of the fits between the inputs and the Kohonen detectors when these neurons are the winner ones. We are introducing a new concept called the “Tolerance margin”. The goodness of the representation of a pattern by a detector will be a function of the ratio of the euclidean distance between both of them and the Tolerance margin of the detector. This justifies its neurodynamic and it learning law [1].

The last module embodiment the supervised character of this neural network, is implemented by a perceptron type net [14] and perform the last stage of a classification process, the discrimination task. It receives the input values from the previous stage and has an output for each class to recognize. One of the most important advantages in this network is its tremendous speed. Possible trainings take place between 10 and 100 times faster than the conventional back-propagation networks, producing similar results. Increased speed is attributed to the simplification which occurs in the self-organizing stage. Simplification also allows this stage to use a simple classifier which produces proven convergence in linear problems, better generalization skills and a reduction in the consumption of computing resources [15].

2.1.2 HUMANN-S Ensemble System

Two strategies are needed to build an ensemble system, diversity and combination strategies. In our development we have used as diversity strategy, several HUMANN-S with data fusion scheme and lowest validation errors. The combination strategies used were the simple and the weighted majority voting, working with two schemes of HUMANN-S ensemble (SMVE/WMVE).

SMVE system uses the modules with the lowest validation errors and creates a simple majority voting process based on their outputs. Each module emits a vote, indicating whether it considers that said input belongs to a class or is unknown. A later module is responsible for the overall count, of considering whether the input belongs to one class or another, depending on whether most of the HUMANN-S modules consider it. Then the class with the maximum number of votes is selected as the ensemble output.

The WMVE system assigns different weights to the HUMANN-S modules, according to their performance, in order to avoid the possibility of certain classifiers to be better than the other ones. We assign heavy weights to the decisions of the more expert classifiers [12]. It is assigned a weight to each classifier in proportion to its estimated performance. We have used the single HUMANN-S validation error, as an estimate of that classifier’s future performance. The combination of the outputs of several classifiers does not guarantee a superior performance to the best ensemble module, but it clearly reduces the risk of making a particularly poor selection [12].

3 Results and Discussion

The systems have been evaluated using a dataset that includes results from 267 clinical consultations on 30 patients during 2005 at the Alzheimer's Patient Association of Gran Canaria [16]. The data structure includes a patient identifier, results from 5 neuropsychological tests: Mini Mental Status Examination (MMSE), Functional Assessment Staging scale (FAST), Katz's index (Katz), Barthel's index (Bar) and the Lawton-Brody's index (L-B), and a diagnosis of CI as well as differential dementia. An important advantage in this data set is its homogeneity, each patient has scores from monthly tests, except the Mini Mental test, which is carried out twice a year.

Nevertheless, even though the majority of the patients were tested 12 times, there are some patients with missing data in their consultations. In effect, we are working with a dataset where a missing data feature is present, since not all of the patients are subject to the complete set of tests.

Table 1. Comparison between results for the validation sets of best module and ensembles

		HUMANN-S MSE+FAST	*SMVE / WMVE*
Sensitivity	*Mild CI*	88%	78%
	Moderate CI	85%	98%
	Severe CI	97%	100%
Specificity	*Mild CI*	82%	100%
	Moderate CI	90%	91%
	Severe CI	96%	99%
	Error	8.61%	4.12%

We obtain the error for the calculated training and validation sets by means of *k*-fold cross-validation (30 folds by leaving consultations of one-patient out) method for all of the HUMANN-S modules with different input subgroups. The best module using all validation data is the HUMANN-S with the MMSE and FAST tests inputs (HUMANN-S MMSE+FAST). The error reached in the validation set is 8.61%. Table 1 includes all the statistics for the validation sets. These results improve successfully, in a 0.38% the validation error, to the best Counterpropagation network obtained in our previous work [10], the same happened for the rest of the modules.

The 7 best modules stored by error using only the data with some available value to the input were used in the SMVE. They correspond to the following input combinations and errors: MMSE(5.73%), MMSE+FAST(8.61%), MMSE+FAST+Katz (10.11%), MMSE+FAST+Bar(10.11%), MMSE+L-B(10.49%), MMSE+FAST+L-B(11.99%) and MMSE+Bar(12.36%). The obtained system with this approach presents an error in the validation sets of 4.12% (see Table 1). These results successfully improve, by 4.04% the validation error when compared to the best one from the modules (HUMANN-S MMSE+FAST) applied on the total of all consultations. The drawback of this system is its computational requirements. Output computation of more than one module is required and combines the exit of theirs.

With WMVE we used also the 7 best HUMANN-S modules. The errors produced in the validation set are the same as in SMVE, and it rate error is also 4.12% (see Table 1). 4 of the 11 errors produced by both ensembles were misclassified by all of 7 modules, another 5 of them were misclassified by 6 modules and the remaining 2 cases were misclassified by 5 modules. HUMANN-S MMSE+FAST, the best module also for this 11 cases, failed to classify 7 of them; HUMANN-S MMSE failed on 9; HUMANN-S MMSE+FAST+Katz, HUMANN-S MMSE+FAST+Bar and HUMANN-S MMSE+L-B failed to classify 10 of them and the remaining modules failed on all of 11 cases. The Principal Component Analysis (PCA) of our data set [16] can help us to determine the difficulties of it classification. In Fig. 2. we show the errors of SMVE and WMVE inside PCA graphical projection.

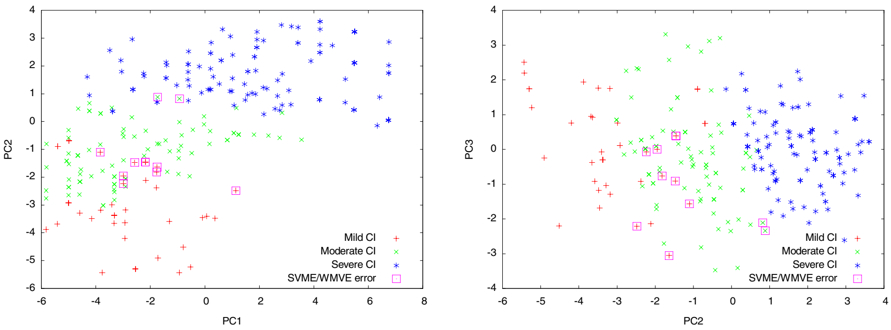


Fig. 2. Scores of first (PC1), second (PC2) and third (PC3) factors from PCA. Show patients with Mild CI (+), Moderate CI (x), Severe CI (*) and the SVME and WMVE errors (□).

4 Conclusions

Our work represents important advances on intelligent decision systems, medical informatics and on healthcare. We have presented intelligent diagnostic systems embodied in a clinical virtual environment to assist the diagnosis and prognosis of CI, AD and other dementias, EDEVITALZH, Our proposal is a personalized, predictive, preventive, and participatory-healthcare delivery system (4P-HCDS) and is an optimal solution for an e-health framework.

We propose new CI diagnostic tools based on a data fusion scheme, artificial neural networks and ensemble systems. We have designed a supervised HUMANN with capacity of processing missing data which provides an important performance for this difficult application.

The use of this new instrument can help alleviate the degree of existing underdiagnosis, since it would be used in primary care centres. It also allows early detection of dementia which can be used to design appropriate therapeutic strategies which are practically effective in the early phase of the disease. In addition to this, the analysis conducted on the different modules which the system up, allows the best cognitive test for a correct diagnosis to be selected. Therefore the study of other possible tests could be extended to elaborate refined diagnosis protocols.

References

[1] García-Báez, P., Fernández López, P., Suárez-Araujo, C.P.: A Parametric Study of HUMANN in Relation to the Noise. Application to the Identification of Compounds of Environmental Interest. Systems Analisis Modelling and Simulation 43, 1213–1228 (2003)

[2] Polikar, R., Topalis, A., Green, D., Kounios, J., Clark, C.M.: Comparative multiresolution analysis and ensemble of classifiers approach for early diagnosis of Alzheimers disease. Computers in Biology and Medicine 37, 542–558 (2007)

[3] López-Pousa, S.: La demencia: concepto y epidemiología. In: Enfermedad de Alzheimer y otras demencias, pp. 33–42. Médica Panamericana, Madrid (2006)

[4] Andersen, C.K., Wittrup-Jensen, K.U., Lolk, A., Andersen, K., Kragh-Sorensen, P.: Ability to perform activities of daily living is the main factor affecting quality of life in patients with dementia. Health Qual Life Outcomes 2, 52 (2004)

[5] Nestor, P.J., Scheltens, P., Hodges, J.R.: Advances in the early detection of Alzheimer's desease. Nature Reviews Neuroscience 5, S34–S31 (2004)

[6] Borson, S., Scanlan, J.M., Watanabe, J., Tu, S.P., Lessing, M.: Simplifying detection of cognitive impairment: comparison of the Mini-Cog and Mini-Mental State Examination in a multiethnic sample. Journal of the American Geriatrics Society 53, 871–874 (2005)

[7] Neuropathology Group of the Medical Research Council Cognitive Function and Ageing Study: Pathological correlates of late-onset dementia in a multicentre, community-based population in England and Wales. The Lancet 357, 169–175 (2001)

[8] Solomon, P., Murphy, C.: Should we screen for Alzheimer's disease? Geriatrics 60, 26–31 (2005)

[9] Lehmann, C., Koenig, T., Jelic, V., Prichep, L., John, R.E., Wahlund, L., Dodge, Y., Dierks, T.: Application and comparison of classification algorithms for recognition of Alzheimer's disease in electrical brain activity (EEG). Journal of Neuroscience Methods 161, 342–350 (2007)

[10] García Báez, P., Fernández Viadero, C., Regidor García, J., Suárez Araujo, C.P.: An Ensemble Approach for the Diagnosis of Cognitive Decline with Missing Data. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS, vol. 5271, pp. 353–360. Springer, Heidelberg (2008)

[11] Suárez Araujo, C.P., Pérez del Pino, M.A., García Baez, P., Fernández López, P.: Clinical Web Environment to Assist the Diagnosis of Alzheimer's Disease and other Dementias. WSEAS Transactions on Computers 6, 2083–2088 (2004)

[12] Polikar, R.: Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine 6, 21–45 (2006)

[13] Samad, T., Harp, S.A.: Self-organization with partial data. Network 3, 205–212 (1992)

[14] Rosenblatt, F.: Principles of Neurodynamics. Spartan Books, Washington (1961)

[15] Hrycej, T.: Modular Learning in Neural Networks. John Wiley and Sons, New York (1992)

[16] García Báez, P., Suárez Araujo, C.P., Fernández Viadero, C., Regidor García, J.: Automatic Prognostic Determination and Evolution of Cognitive Decline Using Artificial Neural Networks. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 898–907. Springer, Heidelberg (2007)

Stratification Methodologies for Neural Networks Models of Survival

Ana S. Fernandes[2], Ian H. Jarman[1], Terence A. Etchells[1]
José M. Fonseca[2], Elia Biganzoli[3], Chris Bajdik[4], and Paulo J.G. Lisboa[1]

[1] School of Computing and Mathematical Sciences, Liverpool John Moores University,
Byrom Street, Liverpool L3 3AF, UK
`{T.A.Etchells,I.H.Jarman,P.J.Lisboa}@ljmu.ac.uk`
[2] Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa
`{asff,jmf}@uninova.pt`
[3] Universita degli Studi di Milano, 20133 Milano, Italy
`elia.biganzoli@unimi.it`
[4] British Columbia Cancer Agency, Vancouver BC V5Z 1L3, Canada
`cbajdik@bccrc.ca`

Abstract. Clinical management often relies on stratification of patients by outcome. The application of flexible non-linear time-to-event models to stratification of patient populations into different and clinically meaningful risk groups is currently an important area of research. This paper proposes a definition of prognostic index for neural network models of survival. This index underpins different stratification strategies including k-means clustering, regression trees and recursive application of the log-rank test. It was obtained with multiple imputation applied to a neural network model of survival fitted to a substantial data set for breast cancer (n=931) and was evaluated with a large out of sample data set (n=4,083). It was found that the constraint imposed by regression trees on the form of the permitted rules makes it less specific than stratifying directly from the prognostic index and deriving unconstrained low-order rules with Orthogonal Search Rule Extraction.

Keywords: Prognostic risk index, risk stratification.

1 Introduction

In a clinical environment stratification of patients, by risk, based on survival models is frequently used in the evaluation of treatments or to assess the impact of prognostic factors on survival. The most widely used statistic test to stratify prognostic indices in the presence of censored data is the log-rank test. However, there are some concerns about using this statistic as a means of identifying patient cohorts, since thresholding a prognostic index (PI) does not necessarily separate patients into groups with different clinical characteristics. Therefore, this paper presents a comparison between three stratification methodologies. The first uses the log-rank statistic to separate a PI into significantly different survival groups. For this method two different survival models were considered: PLANN-ARD, a neural network for time-to-event data and Cox proportional hazards. The methodology relies on thresholding the PI to stratify

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 989–996, 2009.
© Springer-Verlag Berlin Heidelberg 2009

patients and thus risks mixing patients with different clinical characteristics into groups within a given range of prognostic scores. The second method is a regression tree, based on CART algorithm (Classification and Regression Trees) [7], where the PI is now the target for regression using a rule-tree. The grouping of variables into rules ensures uniformity of patient characteristics within each rule. The third methodology is a robust unsupervised clustering methodology that uses k-means where only patient covariates without any knowledge of outcome are considered.

It is important to note that survival models must take account of censorship, which occurs when a patient drops out of the study before the event of interest is observed or if the event of interest does not take place until the end of the study. All patients in this study were censored after 5 years of follow-up.

Real-world clinical data, especially when routinely acquired, are likely to have missing data. Previous research [3] of this dataset showed the missing information as missing at random (MAR) and can therefore be successfully imputed. The application of multiple imputation in combination with neural network models for time-to-event modelling takes account of missing data and censorship within principled frameworks [3]. Section 2 gives a description of the two datasets used for training and external validation. Sections 3 and 4 present the prognostic models employed and the methodologies used for patient stratification into different survival groups. Next an external validation of the results is presented, followed finally by the conclusions of the analysis.

2 Data Description and Model Selection

The training data set consists of 931 routinely acquired clinical records for patients recruited by Christie Hospital in Wilmslow, Manchester, UK, during 1990-94 and the validation data set, which comprises 4083 records, was acquired by the British Columbia Cancer Agency (BCCA), British Columbia, during 1989-93. The latter has also been used to validate another prognostic model for breast cancer survival [10]. The cohort under study consist of patients with early or operable breast cancer and were selected using the standard TNM (Tumour, Nodes, Metastasis) staging system as tumour size less than 5 cm, node stage less than 2 and without clinical symptoms of metastatic spread. Two of the 931 records in the training data were identified as outliers and removed. Time-to-event was measured in months from surgery for Christie and from date of diagnosis for BCCA with a follow up of 5 years and the event of interest death to any cause. There is an assumption that surgery took place soon after diagnosis, therefore survival is not significantly affected evidenced by the survival profiles. Both datasets contain 9 identical variables, presented in table 1.

After analysis of the data, missing information was considered Missing at Random (MAR). Given this, a new attribute may be created to denote missing or the missing values can be imputed. The latter has been shown to be effective [1,2] and does not make assumptions relating to the distribution of “missingness” in the training data, which is essential on future patients inference. Therefore the missing covariates were imputed following the method indicated in [1,2] and repeated 10 times. The choice of this number is a conservative choice, as several studies have shown that the required number of repeated imputations can be as low as three for data with up to 20% missing.

Table 1. Variable description, existing values, coding for each category and missing prevalences for Christie Hospital data set and BCCA data set

Variable description	Values	Categories
Age category	1;2;3	20-39; 40-59; 60+
Histological type	1;2;3	Invasive ductal ; Invasive lobular/lobular in situ; In situ/ mixed/ medullary/ ucoid/ papillary/ tubular/ other mixed in situ
Menopausal status	1;2;3	Pre-Menopausal; Peri-Menopausal; Post-menopausal
	9	Missing (Christie- 0% ; BCCA-2,8%)
Histological Grade	1;2;3	Well differentiated; Moderately differentiated; Poorly differentiated
	9	Missing(Christie- 33,4% ; BCCA-10%)
Ivn	1;2	Lymphatic, vascular or neural invasion present; Ivn absent
	9	Missing(Christie- 0% ; BCCA-5,3%)
Nodes involved	1;2;3;4	0; 1-3; 4+; 98 (too many to count)
	9	Missing(Christie- 19,7% ; BCCA-0%)
Nodes removed	1;2;3;4	0-9 ; 10-19; 20+; 98(too many to count)
	9	Missing(Christie- 0% ; BCCA-2,8%)
Nodes Ratio	1;2;3;4	0-20% ; 21-40% ; 41-60% ; 61+%
	9	Missing(Christie- 19,7% ; BCCA-2,8%)
Pathological size	1;2	<2 cm ; 2-5 cm
Oestrogen receptors	1;2	0-10; 10+
	9	Missing(Christie- 51% ; BCCA-19%)
Metastasis stage	1	M0 (no distant metastasis)

Model selection was carried out through Cox regression (proportional hazards) [3], where six predictive variables were identified: *age at diagnosis*, *node stage, histological type*, *ratio of axillary nodes affected to axilar nodes removed*, *pathological size* (i.e. tumour size in cm) and *oestrogen receptor count.* All of these variables are binary coded as 1-from-N.

3 Prognostic Modeling for Breast Cancer Patients

Two analytical models were used to fit the training data set: a piecewise linear model Cox regression, also known as proportional hazards and a flexible model consisting of the Partial Logistic Artificial Neural Networks regularised with Automatic Relevance Determination (PLANN-ARD). Cox regression factorises dependence on time and the covariates, where the hazard rate is modelled for each patient with covariates x_p at time t_k, as follows:

$$\frac{h(x_p,t_k)}{1-h(x_p,t_k)}=\frac{h_0(t_k)}{1-h_0(t_k)}.\exp(\sum_{i=1}^{N_i} bx_i) \qquad (1)$$

where h_0 represents the empirical hazard for a reference population and x_i are the patient variables. Here the PI is defined by the traditional linear index βx, for this study it was used for 10 imputed data sets, which means that the final patient PI was determined as the mean of the 10 prognostic indices.

PLANN-ARD has the structure of a multi-layer perceptron with a single hidden layer and sigmoidal activations in the hidden and output layer nodes. Covariates and discrete monthly time increments are introduced into the network as inputs, where the output is the hazard for each patient and for each time. The objective function is the log-likelihood summed over the observed status of the patient with a binary indicator when the patient status is observed alive or has died. Automatic Relevance Determination (ARD) [4] has the effect of suppressing covariates that have little predictive influence on outcome. This has the important advantage of automatically adjusting the effect of each covariate according to its relevance to the model, protecting against over-fitting of the data without requiring hard model selection. The papers cited in [4] make a strict theoretical correspondence between this neural network model and classical statistical time-to-event models for censored data. It is also important to define a time independent PI as a natural extension of the *b,* appropriate for non-linear models, the following expression is proposed:

$$\mathrm{PI}(x_p) = (-\ln(1-\mathrm{CCI}(t))) = \ln(-\ln(S(t))) . \tag{2}$$

where the CCI is the crude cumulative incidence, identified as the probability of the occurrence of a specific event of interest [5] and *S(t)* is the estimated survival at the end of follow up. As a consequence of imputation, PLANN-ARD was run 10 times for each imputed data set, which resulted in 10 different trained networks. The mentioned time independent PI was also computed for the 10 networks and averaged for each patient, producing a final PI. Once the PI is defined, a stratification methodology needs to be defined in order to separate the different patients in statistically significant risk groups by overall mortality.

4 Stratification Methodologies

Prognostic indices can be split in order to obtain different survival groups. However the definition of cohorts with significantly different survival may not be determined directly from a PI. Consequently three different stratification methodologies are compared: a robust bootstrap log-rank aggregation, a regression decision tree and a clustering method.

The bootstrap log-rank aggregation [6] is based on the log-rank test. This statistic optimizes the cut-points resulting in an overestimation of the relative risk between the two prognostic groups. Moreover, it should be kept in mind that the cut-point is highly data dependent and could vary markedly between different data sets. Bootstrapping re-sampling can be applied to the prognostic indices, diminishing the cut-points overestimation and dependence on the data. Regression tree decision methodology was performed using CART algorithm [7] with 6 predictor categorical variables (using the mode of the imputed data sets obtained previously) and one continuous target variable, which was the PI previously obtained. In order for populations to be compared the minimum number of records allowed for each node tree was 20 and 10 for each leaf. Two trees were obtained for the Cox regression and PLANN-ARD prognostic indices. After growing the tree a “pruning method” was developed in order to combine pairs of leaves which do not have significantly different survival, calculating a new average for the PI for that new leaf. This is repeated until all leaves have 95% significantly different pairwise survival. The final leaves are ordered by their average PI and each pairwise log-rank statistic computed. For both trees there were 4 survival groups, which means the regression trees after the application of the "pruning method" can be considered classification trees.

The clustering method is an orthogonal, unsupervised, approach where the clinical data is clustered without reference to the PI, then organised in order of mean group survival. It is an iterative k-means algorithm, which uses Monte Carlo methods to overcome initialization problems. Several cluster solutions were considered with 3, 4, 5 and 6 cluster centres. The first two solutions with 3 and 4 centres were analysed further, any more than this was rejected due to sparse membership for some cluster centres. For all methods, risk groups displayed distinct observed survival, measured by Kaplan-Meier actuarial estimates. However group separation is much better for CART and log-rank.

5 External Validation of Stratification Methodologies

The PI was calculated for each patient in the validation dataset using the appropriate parameters found in the training data for the Cox regression and PLANN-ARD respectively. Missing attributes were imputed using the same methodology as with the training data set. Each β vector, obtained previously using Cox regression being multiplied by the imputed variables and each patient's imputed variables entered into the 10 imputed networks. Therefore there will be 100 prognostic indices for each patient for both models. These 100 PI are then averaged in order to obtain a single time

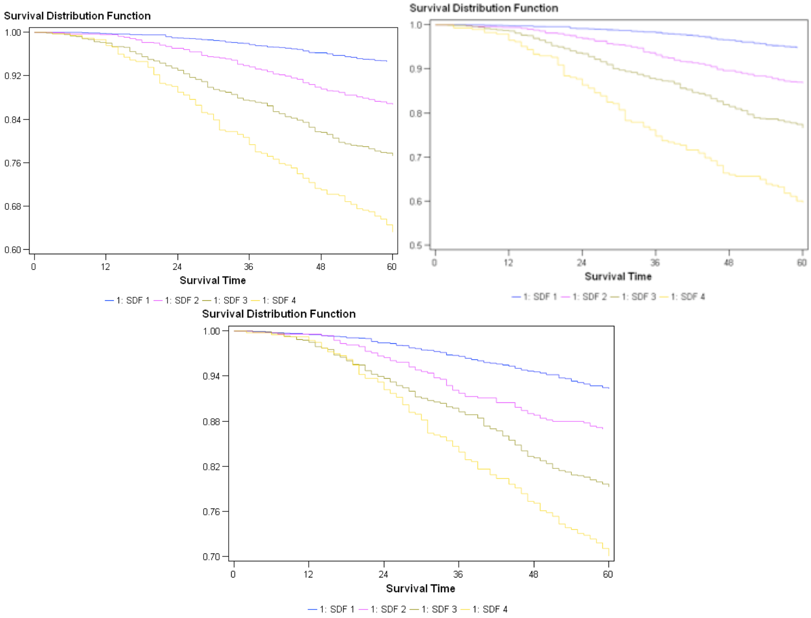


Fig. 1. Actuarial estimates of survival obtained with the Kaplan-Meier method, stratified over a 60 month period using the validation data set (BCCA). The more left picture represents the regression tree methods; the more right graph uses the bootstrap log-rank aggregation, both for Cox PI. The bottom picture uses the clustering algorithm (4 clusters).

Table 2. BCCA Patients' cross tabulation for Log-rank Bootstrapping and CART methodology using the PI obtained with Cox

		Log-rank Bootstrapping Cox				
		Group1	Group2	Group3	Group4	**Total**
CART Cox	Group1	1612	108	3	0	1723
	Group2	233	833	150	11	1227
	Group3	0	67	661	33	761
	Group4	0	0	130	242	372
	Total	1845	1008	944	286	4083

Table 3. Rules obtained for the 4th risk group, for CART and Log-rank bootstrapping methodology, where the PI was obtained with Cox

Rules	**CART rules**	**OSRE rules**
Rule 1	Histype=1 and Pathsize=2 and Noderatio=2 and Histgrade in {1, 2} and Agecategory in {1, 3}	Histgrade in {1, 2} and Noderatio=1 and Oestrog=2 and Pathsize =1
Rule 2	Histype=1 and Pathsize=2 and Noderatio in {2, 3} and Histgrade=3	Histtype=3 and Noderatio in {1,2, 3}
Rule 3	Histype=1 and Pathsize=2 and Noderatio=4	Histtype in {2, 3} and Noderatio=1
Rule 4	-	Histgrade in {1, 2} and Noderatio=1 and Oestrog =2 and Agecategory=2
Rule 5	-	Histtype in {2, 3} and Noderatio in {1,2,3} and Oestrog=2 and Pathsize= 1

independent PI. The mode of the prognostic indices were used from the 10 imputed datasets were used to validate the three stratification methodologies. To define each group for each method the parameters from the training data were used, so for: regression trees the rules, bootstrap log-rank aggregation the PI cut-points identified and clustering the rules acquired for each cluster. For the last set of rules, 38 patients were not identified by any rule and 6 patients were classified as belonging to two different clusters, therefore these can be considered as outliers. Inspecting figure 1, the first two figures present better patient stratification than the figure in the second row, as there is a higher separation between the different survival curves for the first two figures. Although survival for both methods is similar, group membership is not the same, as can be observed in Table 2. Therefore, survival curves were obtained for each cell in Table 2 and are presented in Figure 3. Inspecting figure 3, the Log-rank bootstrapping methodology is more consistent in terms of survival than the CART methodology evidenced by more homogeneous survival in the columns than the rows and confirmed using the log-rank statistic. For comparison, explanatory rules can be obtained for risk groups for both CART and Log-rank Bootstrapping methodologies. CART rules were obtained from the tree, while the latter's rules were obtained using a rule extraction algorithm, OSRE [9], for reasons of space only the rules for group 4 are presented in table 3.

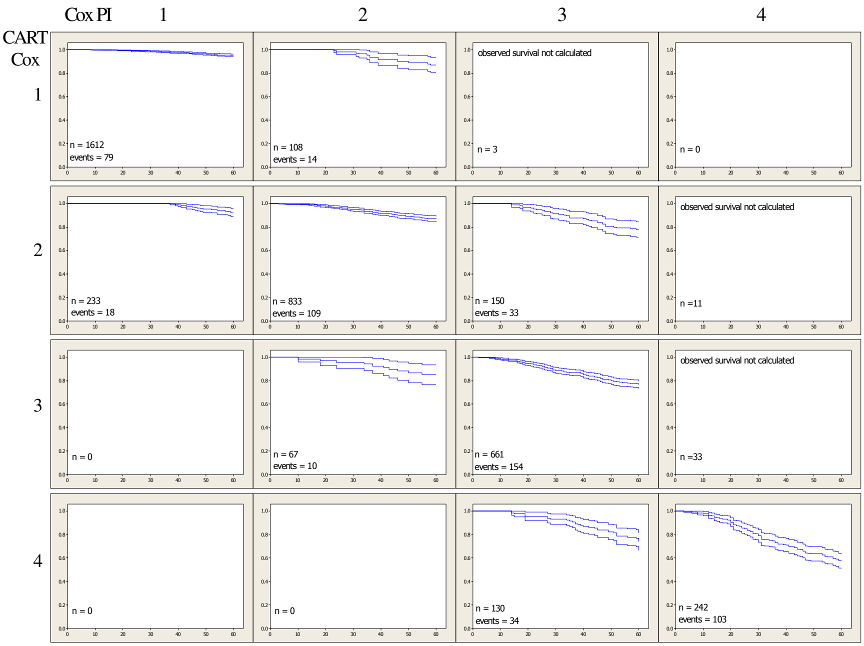


Fig. 3. Survival curves obtained for the patients' cross-tabulation obtained with the CART and Bootstrapping methodology, where it was used the PI obtained with Cox

6 Conclusions

This paper defines a PI that may be used to stratify patients into risk groups with statistically significant grouped outcomes, utilizing flexible non-linear survival models which take into account censorship and values MAR. The proposed use of the Crude Cumulative Incidence rate as the basis of the PI calculation means that this approach will extend from single to competing risks modelling. Three stratification strategies were applied. Pure clustering of the population of covariates without reference to the prognostic indicator results in coherent patient groups but with relatively poor specificity for outcome, as measured by the separation between the group means of the overall mortality rates.

Regressing the distribution of prognostic scores with rule-based trees (CART) succeeds in separating patient groups with statistically different mean survival but coherent membership in each group.

The third approach stratified the PI directly. This can result in mixed populations within single risk groups, which the application of an automatic rule extraction method (OSRE) identified and characterized. This showed the composition of the risk groups to be no more complex than for the groups identified by CART and the mean grouped survival show similar, statistically significant, separation.

Importantly, however, the cross-tabulation of the memberships of the risk groups identified by the two rule extraction methods, CART and OSRE, showed the latter to be much more consistent, by mean overall mortality measured in the out-of-sample data. Therefore, while all methods generalize well to the validation data set, as all risk groups are significantly different evidenced by the Kaplan Meier observed survival curves, thresholding of the PI followed by rule-based characterization of the patient cohorts in identified prognostic risk group is the only methodology to offer both specificity to outcome and transparency of group composition. This method generalizes in external validation as required by a staged methodology for the development of decision support systems in medicine [8] and offers a possible route to a clinically useful patient stratification index.

Acknowledgments. The authors gratefully acknowledge Mr. R. Swindell from Christie Hospital for making the anonymised modelling data set available for this study and members of the BC Cancer Agency's Breast Cancer Outcomes Unit for assembling the British Columbia data. This study was partially funded by the European Network of Excellence Biopattern (FP6/2002/IST/1-508803) and by Fundação para a Ciência e Tecnologia, through the POS_C(SFRH/BD/30260/2006). Chris Bajdik is the recipient of a Scholar Grant from the Michael Smith Foundation for Health Research.

References

1. Clark, T.G., Altman, D.G.: Developing a prognostic model in the presence of missing data an ovarian cancer case study. Journal of clinical epidemiology 56, 28–37 (2003)
2. Van Buuren, S., Boshuizen, H.C., Knook, D.L.: Multiple imputation of missing blood pressure covariates in survival analysis. Statistics in Medicine 18(6), 681–694 (1999)
3. Fernandes, A.S., Jarman, I.H., Etchells, T.A., Fonseca, J.M., Biganzoli, E., Bajdik, C., Lisboa, P.J.G.: Missing data imputation in longitudinal cohort studies – application of PLANN-ARD in breast cancer survival. In: Machine Learning and Applications, ICMLA 2008, pp. 644–649 (2008)
4. Lisboa, P.J.G., Wong, H., Harris, P., Swindell, R.: A Bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer. Artificial Intelligence in Medicine 28(1), 1–25 (2003)
5. Ambrogi, F., Biganzoli, E., Boracchi, P.: Estimates of clinically useful measures in competing risks survival analysis. Statistics in Medicine 27(30), 6407–6425 (2008)
6. Etchells, T.A., Fernandes, A.S., Jarman, I.H., Fonseca, J.M., Lisboa, P.J.G.: Stratification of severity of illness indices: a case study for breast cancer prognosis. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS, vol. 5178, pp. 214–221. Springer, Heidelberg (2008)
7. Breiman, L., Friedman, J.H., Olsen, A.R., Stone, C.J.: Classification and Regression Trees. The Wadsworth & Brooks (1984)
8. Lisboa, P.J.G.: A review of evidence of health benefit from artificial neural networks in medical intervention. Neural Networks 15(1), 9–37 (2002) (invited paper)
9. Etchells, T.A., Lisboa, P.J.G.: Orthogonal search-based rule extraction (OSRE) for trained neural networks: a practical and efficient approach. IEEE Tran. Neural Netw. 17(2), 374–384 (2006)
10. Olivotto, B., Ravdin, S., Coldman, N., Davis, C., Gelmon, K.: Population-based validation of the prognostic model ADJUVANT for early breast cancer. Journal of Clinical Oncology 23(12), 2716–2725 (2005)

Model Comparison for the Detection of EEG Arousals in Sleep Apnea Patients

D. Álvarez-Estévez and V. Moret-Bonillo

Laboratory for Research and Development of Artificial Intelligence (LIDIA)
Department of Computer Science, Faculty of Informatics, University of A Coruña, Campus
Elviña 15071 A Coruña, Spain
`{dalvareze,civmoret}@udc.es`

Abstract. Sleep Apnea/Hypopnea Syndrome (SAHS) is a very common sleep disorder characterized by the repeated occurrence of involuntary breathing pauses during sleep. Cessation in breathing often causes Electroencephalographic (EEG) arousals as a response, and therefore detection of arousals is important since they provide important evidence for localization of apneic events and their number is directly related with SAHS severity. Arousals result in fragmented sleep and so they are one of the most important causes of daytime sleepiness. In this paper we present an approach to detect these arousals over polysomnographic recordings based on the machine learning paradigm. First a signal processing technique is proposed for the construction of learning patterns. Subsequently classifiers based on Fisher's linear and quadratic discriminates, Support Vector Machines (SVM) and Artificial Neural Networks (ANN) are compared for the learning process. The more suitable model was chosen finally showing an accuracy of 0.92.

Keywords: Sleep Apnea/Hypopnea Syndrome, Arousal, Intelligent algorithms.

1 Introduction

SAHS is one of the most common sleep disorders characterized by the involuntary occurrence of interruptions in breathing during sleep [1]. Depending on their intensity, these pauses are classified in hypopneas, when amplitude reduction in airflow is around 50% of normal breathing, or apneas when cessation in airflow is total or nearly total. Among the most frequent symptoms of SAHS are excessive daytime somnolence, snoring, and respiratory pauses. As consequence of the hypoventilation produced by the apneic event, a drop in blood oxygen levels is produced and commonly an alertness response is triggered in the form of an arousal event.

According to the American Academy of Sleep Medicine (AASM), an EEG-arousal during sleep is defined as an abrupt shift in EEG frequency including alpha, theta and/or frequencies greater than 16 Hz. (but not spindles) that lasts at least 3 seconds, with at least 10 seconds of stable sleep preceding the change [2]. Normal sleep architecture is altered by the presence of these events, especially in patients suffering from sleep disorders. The sleep fragmentation they cause is one of the main reasons for daytime sleepiness associated with a number of these disorders.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 997–1004, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Thus localization of these events is important for diagnosis of SAHS but also for sleep studies in general. Sleep studies are performed by means of an overnight polysomnographic recording (PSG). For scoring arousals at least one central derivation (C4/A1 or C3/A2) of EEG need to be performed in the recording. Also scoring of arousal during Rapid Eye Movement (REM) phase requires a concurrent increase in submental electromyography (EMG) lasting at least 1 second [2].

Automation of PSG analysis for detection of arousal events is desirable since this an extremely time-consuming task for the physician. As it is a growing area of interest, a number of papers have been published in recent years [3-6]. For example: F. De Carli et al. [3] suggest a method for the automatic detection of arousals based in wavelet analysis followed by a threshold selection. In S.P. Cho et al. [4], a SVM-based classifier over C3/A2 EEG derivation is proposed. Although it achieves good performance, working only with one EEG derivation limits its applicability as it is not capable of dealing with neither arousals in REM stage nor in awake. Background experience from the authors of this paper on the application of artificial intelligence techniques for the analysis of sleep studies can be found in [5-6].

In the present work, a computer technique for the detection of arousal events in SAHS patients is proposed. The approach combines a first stage of signal analysis, both time and frequency based, where a set of features are extracted along the signals of the patient. After that, in a second phase, artificial intelligence techniques -more precisely from the machine learning field-, are applied over the extracted features in the first stage to classify the corresponding PSG interval as having presence, or not, of an arousal event. A number of models, including Fisher's linear and quadratic discriminants analysis, SVMs and ANNs –with different configurations- are tested and compared as classifiers. We are not aware of a study comparing among different machine learning models applied to arousal detection using the same source of data. The best model, from the data provided by the model comparison, is chosen as the classifier to implement the second phase of the proposed method. A different data set from that used in models comparison will be used in order to validate this approximation.

2 Materials and Methods

The method proposed works with three signals: EEG both C2/A2 and C4/A1 central derivations and submental EMG, all sampled at 125 Hz. As previously mentioned, the method combines a first phase of signal processing whose objective is the detection of possible events in the three signals. Event detection is accomplished independently among the signals. A feature extraction process will be accomplished over these events in order to be used in the second phase to form the learning vectors for both the training and the validation of the different classification approaches analysed.

2.1 First Phase: Signal Processing

In the first phase the signal processing occurs in the three signals in a similar manner. For each second of the recording two temporal windows are associated with the corresponding sample. The two temporal windows are used to compare the consecutive temporal segments representing the past and the future related to the current instance of time under analysis (Fig. 1).

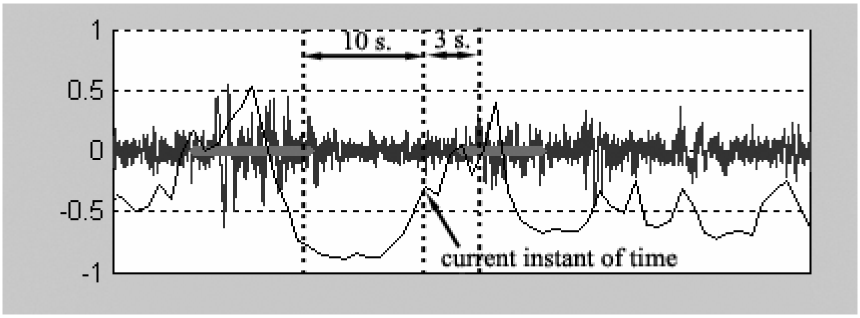


Fig. 1. Processed EEG signal. Displayed is one minute of EEG signal with amplitude normalized in [1,-1]. Signal $e_{\alpha,\beta}(n)$ is superposed and events are marked where this signal is above zero.

In the case of EEGs, for every second and going back in time, a 10-second window ending in the corresponding sample is used to represent the prior information, and immediately following it, a new 3-second window is used to compare the evolution of the latter signal against the former. As the definition of EEG arousal is based on the localization of an abrupt shift in EEG frequency, we perform a frequency-based comparison between these two windows. The chosen duration of each window is based on the idea that the EEG frequency shift must be 3 seconds or greater to be scored as an arousal, and that a minimum of 10 seconds of intervening sleep is necessary to score a second arousal once the former is detected [2].

The frequency comparison is made as follows: (i) each of the two segments is transformed into frequency domain making use of the Fourier Transform and then, (ii) a band-pass filter is applied in the range 8-30 Hz., so including the energy in the alpha (8-12 Hz.) and beta (13-30 Hz.) bands. Finally we measure the energy in the filtered spectra with the following formula:

$$energy = \frac{1}{n^2}\sum_{i=1}^{n}\left|X(n)\right|^2 \tag{1}$$

In equation (1) n is the number of samples, $X(n)$ is the band-passed signal in the frequency domain, and $1/n^2$ is a weight factor in order to make the energy between two signal segments of different duration in time comparable (we are comparing a segment of 10 seconds with a another of 3 seconds). We obtain thus, a scalar representing the energy estimation in each window in the range of 8-30 Hz., so comparing them we obtain a measure of the energy change. Performing this procedure along each of the EEG derivations, we obtain a new signal $e_{\alpha,\beta}(n)$ (1 Hz. resolution) that grows wherever evidence of a shift in the EEG frequency in the alpha and beta range of the signal occurs, decreasing as the evidence disappears (Fig. 1).

Once we have this $e_{\alpha,\beta}(n)$ signal and by means of using a threshold value, we will consider as possible arousal events, those intervals where this signal is above the threshold (Fig. 1).

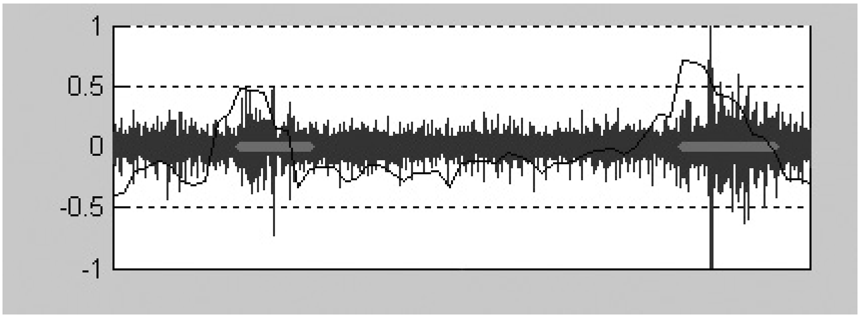


Fig. 2. Processed EMG signal. Displayed is one minute of EMG signal with amplitude normalized in [1,-1]. Signal *a(n)* is superposed and events are marked where this signal is above zero.

Although the possible events will be marked based on shifts in the band of 8-30 Hz., a similar process is accomplished individually for delta (0.5-3 Hz.), theta (4-7 Hz.), alpha (8-12 Hz.), beta (13-30 Hz) and sigma (12-14 Hz.) ranges. These frequency bands are the most relevant in the case of sleep studies [2]. In this way an evolution of the different relevant frequencies along the time is available, and once a possible event is detected with the marker (8-30 Hz.), features from these ranges are available to be extracted to form the vectors for the training.

For the EMG signal searching for possible events is performed in a similar way (Fig. 2). As was the case for EEG we process the signal second by second taking into consideration two windows for each sample obtaining the signal *a(n)*. In this case the former has a duration of 30 seconds, while the latter remains in 3 second its duration. This time the values were chosen empirically. In the case of EMG the objective is the detection of changes in the amplitude of the signal. To achieve this, a comparison between the amplitudes of the two windows is performed this time rather than one based upon frequency.

Finally over all the possible events detected in the signals, a set of features are extracted. These features will be used later to build the learning examples to train different models. A total of 40 features were extracted including for each possible event marked in both, the two derivations of EEG, the maximum, the minimum and the area enclosed during the marked event in each frequency range (delta, theta, alpha, beta and sigma) and in $e_{\alpha,\beta}(n)$. For the EMG, the maximum, the minimum and the area enclosed during the marked event in *a(n)* were used.

2.2 Second Phase: Training Classifiers

To construct the learning patterns, an epoch-by-epoch approach was followed. Establishing 30 seconds as the epoch duration, a 30s window is displaced along the PSG, not overlapping, and features from events localized within the window in the three signals were used to form the pattern. Desired output for each pattern is established using expert annotations provided with each PSG indicating the presence of arousals.

After the construction of the patterns several models were compared using a number of approaches from the machine learning paradigm. A total of 18 models were compared using (i) the classic Fisher's linear discriminant, (ii) a quadratic discriminant, (iii) 10 configurations of Support Vector Machines (SVM) and (iv) 6 configurations of an Artificial Neural Networks (ANN).

Literature on both Fisher's linear and quadratic discriminants can be found in [7]. For the SVMs a radial basis kernel function was chosen, and various configurations were tested for adequate smoothing parameter S for the kernel function, and C, a compromising term between the margin size and the classification error [8]. In total 10 models of SVM were tested for the possible combinations of C = 0.1,1,10,100,1000 and S = 0.1,0.5. Also 6 different configurations of a feed-forward neural network with one hidden layer, trained with a scaled conjungate gradient back-propagation algorithm [9] were tested. A different number of neurons were used in the hidden layer for each configuration, ranging from 10 neurons to 60 on 10-neurons increments.

To select the best model and having a good estimation of the real error, a K-fold cross-validation was employed. We chose K=10 – the number of folds - which will provide us with a good generalization for the error estimation. All the patterns were normalized resulting in each variable having a mean of zero and standard deviation equal to one. To choose the best model, this process was repeated 30 times, and again the overall error is calculated for each model. In all 300 trainings were performed over each model – searching for statistical relevance -, and a Kruskal-Wallis test [10], a nonparametric version of the classical one-way analysis of variance (ANOVA), was performed to identify if there was a significant difference among the means of the models.

If variance test concludes that there is a difference among the means, a Multiple Comparison Method (MCM) [11] is applied to find those different models. The Tukey's method was chosen to compare differences between each pair of means, and select the simplest model over those being significantly the best ones.

3 Results

The set of polysomnographic recordings, as well as the corresponding arousal annotations from the physicians come from the *Sleep Heart Health Study* (SHHS). This database, granted by the Case Western Reserve University, contains more than 2000 PSG recordings obtained from a multi-center cohort study implemented by the National Heart Lung & Blood Institute to determine the cardiovascular and other consequences of sleep-disordered breathing. The resulting database was then placed online to be used as resource for subsequent studies related whit sleep disorders. Details about the design of the SHHS study can be found in [12].

A set of ten records was randomly taken from this database in order to perform the validation. Each record is accompanied with an annotation file of the results of the analysis made by the SHHS scorers. In total the ten records contains 1434 arousal events along 6127 minutes of PSG recording. From the whole of 12255 patterns, 2/3 of them (8170) were used for the comparison of the machine learning models, while the remaining 1/3 (4085) were used for testing the final model. Results for the model comparison are summarized in Table 1. Total averaged error of the 300 iterations of cross-validation and the standard deviation is presented. Error is presented in terms of accuracy, i.e. as the proportion of misclassified patterns.

Table 1. Mean training error and standard deviation for each model with their different configurations

Model	Configuration	Error
Linear discriminator	---	0.1956 ± 0.0152
Quadratic discriminator	---	0.1948 ± 0.0152
SVM	C=0.1, S=0.1	0.1403 ± 0.0119
	C=0.1, S=0.5	0.1403 ± 0.0119
	C=1, S=0.1	0.1403 ± 0.0119
	C=1, S=0.5	0.1403 ± 0.0119
	C=10, S=0.1	0.1403 ± 0.0117
	C=10, S=0.5	0.1404 ± 0.0117
	C=100, S=0.1	0.1404 ± 0.0117
	C=100, S=0.5	0.1403 ± 0.0117
	C=1000, S=0.1	0.1404 ± 0.0120
	C=1000, S=0.5	0.1404 ± 0.0120
ANN	10 hidden neurons	0.0935 ± 0.0116
	20 hidden neurons	0.0925 ± 0.0104
	30 hidden neurons	0.0926 ± 0.0102
	40 hidden neurons	0.0921 ± 0.0098
	50 hidden neurons	0.0925 ± 0.0104
	60 hidden neurons	0.0924 ± 0.0097

It's possible to see from the results that both, linear and quadratic discriminants, are those achieving the worse precision showing an error around 0.19. SVM models are quite similar between them, looking like the values for the parameters C and S for this problem have no particular influence. The mean error among the SVM models is around 0.14. It can be seen from Table 1 that ANN models achieve the best precision. Again without large differences between them, the mean error is about 0.09. To decide which is the more suitable model, a Kruskal-Wallis test was performed comparing the means of errors for each of the 18 classifiers, with the null hypothesis that all means are drawn from the same population –i.e. all the models are statistically equivalents-. A value of 0.05 was chosen as the statistically significant critical value to consider the null hypothesis and a p-value of 0 was obtained, so the null hypothesis was rejected. Tukey's multiple comparison method was then applied in order to find out how models are different from each other. Results are shown in Fig. 3.

In the graph each mean is represented by a symbol and an interval around it. Two means are significantly different if their intervals are disjointed, but are not if their intervals overlap. As can be seen in Fig. 3, although ANN 4 is the one with the best accuracy, there is nothing statistically different among other ANN models. When this happens, normally the simplest model should be chosen. ANN 1 is the simplest model because it is that with least number of neurons –ten- in its hidden layer. A mean error of 0.09 is achieved for ANN 1.

Selecting an ANN with 10 hidden neurons as the more suitable classifier, a new test was performed using then remaining 1/3 (4085) patterns from the whole dataset for testing the final model. Error over this dataset was 0.0846.

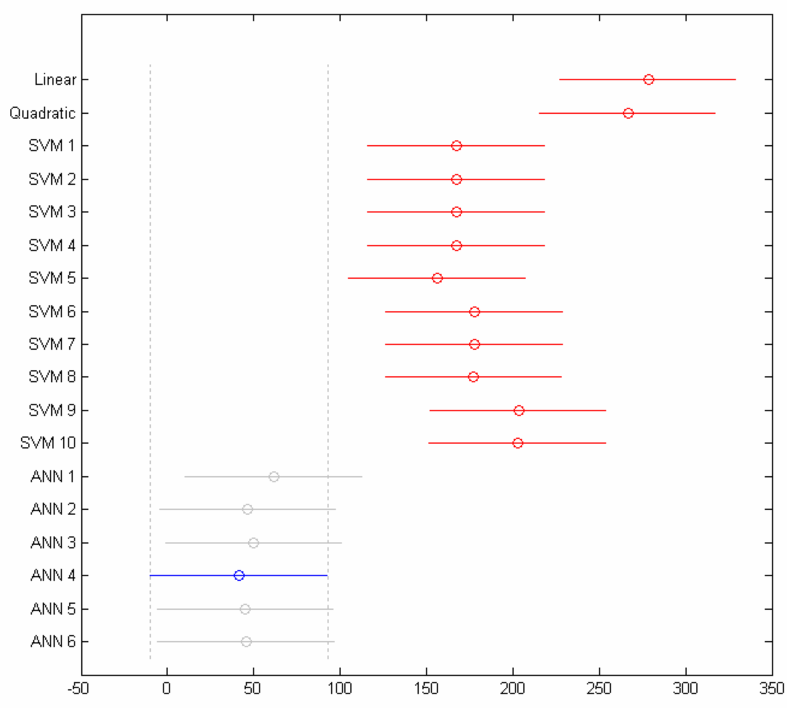


Fig. 3. Models mean multi-comparison. Model with best precision –ANN model with 40 hidden neurons- is selected. Those models whose media overlaps with the interval of the selected method are not considered to be statistically different.

4 Discussion and Conclusions

This paper describes a method for detection of EEG-related arousal events in sleep apnea patients. Detection of these kinds of events is very important for analysis of PSG recordings and diagnosis of sleep diseases like SAHS. Specifically, in case of SAHS, high correlation between apneic events and arousals exists [13].

The method presented is structured in two different phases. Firstly a stage of signal processing for extraction of features occurs. Two derivations of EEG where used, a frequency-based analysis is performed breaking them down into their main frequencies. A marker based on changes into alpha-beta range is used for localization of possible arousal events in these signals. The EMG signal is analysed searching for amplitude changes. Subsequently, in a second stage, an ANN is made responsible for the final classification, deciding if the possible event is a true one or not.

A comparison of various models for the classification was performed, showing that an ANN model with 10 hidden neurons is the more suitable. Two separate sets, for the model comparison and for the final validation of the best model were used. Maintaining similar precision over the final validation set demonstrated generalization capabilities of the proposed method.

Analysis of PSGs is a very time consuming task for physicians. This is why tools that facilitate this task should be provided. The final objective is to achieve a comprehensive supporting tool to assist physicians in SAHS diagnosis. Future work will be focused on incorporating this information on arousal detection to reasoning-capable tools for the diagnosis of SAHS.

Acknowledgments. The authors and the LIDIA research group would like to acknowledge for the support received by program 2007/000134-0 of the Xunta de Galicia, partially supported by FEDER funds.

References

1. Guilleminault, C., Hoed, J., Milter, M.M.: Sleep apneas syndromes. Liss, New York (1978)
2. Iber, C., Ancoli-Israel, S., Chesson, A., Quan, S.F.: The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications, 1st edn. American Academy of Sleep Mecidine, Westchester (2007)
3. De Carli, F., Nobili, L., Gelcich, P., Ferrillo, F.: A Method for the Automatic Detection of Arousals During Sleep. Sleep 22(5), 561–572 (1999)
4. Cho, S.P., Lee, J., Park, H.D., Lee, K.J.: Detection of arousal in patients with respiratory sleep disorders using a single channel EEG. In: 27th IEEE Engineering in Medicine and Biology Annual Conference, pp. 2733–2735 (2005)
5. Cabrero-Canosa, M., Hernandez-Pereira, E., Moret-Bonillo, V.: Intelligent Diagnosis of Sleep Apnea Syndrome. IEEE Eng. Med. Biol. Mag. 23(2), 72–81 (2004)
6. Alvarez-Estevez, D., Moret-Bonillo, V.: Fuzzy Reasoning Used to Detect Apneic Events in the Sleep Apnea-Hypopnea Syndrome. Expert Systems with Applications 36, 7778–7785 (2009)
7. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: Machine Learning: Neural and Statistical Classification. Ellis Horwood, Chichester (1994)
8. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
9. Moller, M.F.: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks 6, 525–533 (1993)
10. Hollander, M., Wolfe, D.A.: Nonparametric Statistical Methods. Wiley, Chichester (1973)
11. Toothaker, L.E.: Multiple Comparison Procedures. Sage Publications Inc., Thousand Oaks (1992)
12. Quan, S.F., Howard, B.V., Iber, C., Kiley, J.P., Nieto, F.J., O'Connor, G.T., Rapoport, D.M., Redline, S., Robbins, J., Samet, J.M., Wahl, P.W.: The Sleep Heart Health Sutdy: design, rationale and methods. Sleep 20(12), 1077–1085 (1997)
13. Pitson, D., Stradling, J.: Autonomic Markers of Arousal during Sleep in Patients Undergoing Investigation for Obstructive Sleep Apnoea, Their Relationship to EEG Arousals, Respiratory Events and Subjetive Sleepiness. J. Sleep Res. 7(1), 53–59 (2002)

Ranking of Brain Tumour Classifiers Using a Bayesian Approach

Javier Vicente[1,⋆], Juan Miguel García-Gómez[1], Salvador Tortajada[1],
Alfredo T. Navarro[1], Franklyn A. Howe[2], Andrew C. Peet[3], Margarida Julià-Sapé[4,5],
Bernardo Celda[6,5], Pieter Wesseling[7], Magí Lluch-Ariet[8], and Montserrat Robles[1]

[1] IBIME-Itaca, Universidad Politécnica de Valencia, Spain
[2] St George's University of London, UK
[3] University of Birmingham and Birmingham Children's Hospital NHS Foundation Trust, UK
[4] Universitat Autònoma de Barcelona, Spain
[5] Departamento Química Física, Universidad de Valencia, Spain
[6] CIBER Bioengineering, Biomaterials and Nanomedicine, ISC-III, Spain
[7] Radboud University Nijmegen Medical Centre, Netherlands
[8] MicroArt, Barcelona, Spain

Abstract. This study presents a ranking for classifers using a Bayesian perspective. This ranking framework is able to evaluate the performance of the models to be compared when they are inferred from different sets of data. It also takes into account the performance obtained on samples not used during the training of the classifiers. Besides, this ranking assigns a prior to each model based on a measure of similarity of the training data to a test case. An evaluation consisting of ranking brain tumour classifiers is presented. These multilayer perceptron classifiers are trained with ^{1}H magnetic resonance spectroscopy (MRS) signals following a multiproject multicenter evaluation approach. We demonstrate that such a framework can be effectively applied to the real problem of selecting classifiers for brain tumour classification.

1 Introduction

Several decision support systems (DSS) aiming at the assessment of the diagnosis and prognosis of brain tumours are conceived as dynamic systems where the classification tools based on Machine Learning (ML) can be created and updated throughout time with the incorporation of new cases. This approach is followed in the distributed DSS developed in the HEALTHAGENTS project (2005-2008) [1] and in the DSS for clinical environments CURIAM [2]. Such systems can offer diagnostic support from a large

---

[⋆] This work was partially funded by the European Commission (FP6-2002-LIFESCIHEALTH 503094) and (IST-2004-27214). The authors acknowledge to Programa Torres Quevedo from Ministerio de Educación y Ciencia, co-founded by the European Social Fund (PTQ05-02-03386). We thank eTUMOUR and INTERPRET partners for providing data, in particular J. Capellades (IDI-Badalona), C. Majós (IDI-Bellvitge), A. Moreno (Centre Diagnòstic Pedralbes), J. Griffiths (SGHMS), W. Gajewicz (MUL), Jorge Calvar (FLENI), Antoni Capdevila (H Sant Joan de Déu), A. Heerschap (RUNMC) W. Semmier (DKFZ-Heidelberg) and Carles Arús (UAB).

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1005–1012, 2009.
© Springer-Verlag Berlin Heidelberg 2009

number of potential classifiers and a procedure to sort the classifiers according to their effectiveness for the required differential diagnosis is needed.

This paper introduces a novel ranking model based on a Bayesian approach. A demonstration of the versatility of this model is presented for dynamically ranking the most suitable classifiers from a brain tumour case classification request.

2 Bayesian Approach for Model Sorting

The Bayes' rule can be applied to compare different models given the data $\mathbf{Z} = \{z_j\}_1^N = \{\mathbf{x}_j, t_j\}_1^N$, where $\mathbf{x}_j$ is a data vector describing the j^{th} sample and t_j is its associated label. The posterior probability of a model $\mathcal{M}_i$ can be expressed as:

$$P(\mathcal{M}_i|\mathbf{Z}) = \frac{P(\mathcal{M}_i)P(\mathbf{Z}|\mathcal{M}_i)}{P(\mathbf{Z})}, \tag{1}$$

where $P(\mathbf{Z}|\mathcal{M}_i)$ is the model likelihood or evidence for $\mathbf{Z}$. The term $P(\mathcal{M}_i)$ is a 'subjective' prior over the model space. This subjective part of the inference is typically overwhelmed by the objective term, the evidence [3]. $P(\mathbf{Z})$ is usually ignored since it is assumed that models are compared for the same $\mathbf{Z}$.

To compare two models $\mathcal{M}_m$ and $\mathcal{M}_l$, we form the posterior odds

$$\frac{P(\mathcal{M}_m|\mathbf{Z})}{P(\mathcal{M}_l|\mathbf{Z})} = \frac{P(\mathcal{M}_m)}{P(\mathcal{M}_l)} \frac{P(\mathbf{Z}|\mathcal{M}_m)}{P(\mathbf{Z}|\mathcal{M}_l)} \tag{2}$$

If the odds are greater than one we choose model m, otherwise we choose model l. $\frac{P(\mathbf{Z}|\mathcal{M}_m)}{P(\mathbf{Z}|\mathcal{M}_l)}$ is called the Bayes factor (BF), the contribution of the data toward the posterior odds [4]. Notice that BF is a ratio of the evidence of the alternative models.

Several techniques are available for computing Bayes factors. An exhaustive review can be found in [5]. The Bayesian Information Criterion (BIC) gives a rough approximation to the logarithm of BF, which does not require evaluation of prior distributions of the parameters of the models [6].

In our application scenario, classifiers offering orientation to brain tumour diagnosis can be trained with different sets of data. Thus, each model $\mathcal{M}_i$ can be trained from a set of samples $\mathbf{Z}_i$ of the same population $\mathbf{Z}$.

In a classification problem, comparing the predictive distributions of the models (how well they predict new data) gives more information than comparing their evidences (how well they explain the data). In other words, we are interested in finding the predictive distribution for $\mathbf{Z}$ of each $\mathcal{M}_i$ trained with $\mathbf{Z}_i$. Thus, we could re-arrange (2):

$$\frac{P(\mathcal{M}_m|\mathbf{Z})}{P(\mathcal{M}_l|\mathbf{Z})} = \frac{P(\mathcal{M}_m)}{P(\mathcal{M}_l)} \frac{P(\mathbf{Z}_{\neg m}|\mathbf{Z}_m, \mathcal{M}_m)}{P(\mathbf{Z}_{\neg l}|\mathbf{Z}_l, \mathcal{M}_l)} \tag{3}$$

The righmost expression is a ratio of the predictive distributions of two models. Notice that $\mathbf{Z}$ has been splitted into the training samples of each model ($\mathbf{Z}_m$ and $\mathbf{Z}_l$, respectively) and the samples not used for tuning their parameters ($\mathbf{Z}_{\neg m}$ and $\mathbf{Z}_{\neg l}$).

Two alternative methods are proposed to perform the model comparison when different datasets are used for training each model: **Method A**, consisting on creating a $\mathbf{Z}$

from the union of samples with labels common to every $\mathbf{Z}_i$; and **Method B**, which joins all the samples, $\mathbf{Z} = \bigcup \mathbf{Z}_i$, and applies Mean Imputation (MI) when a model is unable to deal with some of the labels. Although MI can lead to biased estimations when used for missing data, in this context it is applied as an estimator of the mean likelihood obtained by the set of classifiers able to discriminate a specific label.

The first method offers a way of obtaining a common set of samples for comparing models addressing different questions but with some common answers (namely labels). It also ensures that every model is able to discriminate any sample in $\mathbf{Z}$. The latter, however, is less restrictive and offers a compromise solution that balances the weight assigned to a model able to discriminate classes with low prevalences in $\mathbf{Z}$ with models capable to solve more general, yet less challenging questions.

2.1 Assigning Proper Priors to Models

Up to now, we have assumed that we have no reason to assign strongly differing priors $P(\mathcal{M}_i)$ to the alternative models. However, it could be useful to calculate the prior according to the similarity of the current test sample z^κ with respect to the data each model has been trained with. For doing this, let us assume that for each set of samples $\mathbf{Z}_i = \{z_j\}_1^N$ there exists a set $\mathcal{K}_i = \{\mathbf{k}_j\}_1^N$, with direct correspondence between the j^{th} sample z_j and $\mathbf{k}_j$. Each $\mathbf{k}_j$ is a set of values describing the contextual information of the j^{th} sample. Contextual information is defined as extra information about the samples not used for training purposes. The prior probability of $\mathcal{M}_i$ can be expressed as:

$$P(\mathcal{M}_i) = P(k_\kappa|\mathcal{K}_i) = \int P(k_\kappa|\omega, \mathcal{K}_i)P(\omega|\mathcal{K}_i)d\omega, \tag{4}$$

where k_κ is the contextual information associated to the current sample z_κ to be tested by the different candidate models and ω is the set of parameters modeling the contextual information. $P(k_\kappa|\mathcal{K}_i)$ is a way of objectively determine the similarity of a test sample to the training set of each classification model to be compared.

2.2 Measuring Train and Test Performances

Let us suppose that there exists $\mathbf{Z}_{test}$, composed by samples different to any sample in $\mathbf{Z} = \bigcup \mathbf{Z}_i$.

If we assume that samples in $\mathbf{Z}$ and $\mathbf{Z}_{test}$ are independent and identically-distributed (i.i.d.), we can calculate the next posterior odds for comparing $\mathcal{M}_m$ and $\mathcal{M}_l$:

$$\begin{aligned}\frac{P(\mathcal{M}_m|\mathbf{Z}, \mathbf{Z}_{test})}{P(\mathcal{M}_l|\mathbf{Z}, \mathbf{Z}_{test})} &= \frac{P(\mathcal{M}_m)}{P(\mathcal{M}_l)} \frac{P(\mathbf{Z}_{\neg m}, \mathbf{Z}_{test}|\mathbf{Z}_m, \mathcal{M}_m)}{P(\mathbf{Z}_{\neg l}, \mathbf{Z}_{test}|\mathbf{Z}_l, \mathcal{M}_l)} \\ &= \frac{P(\mathcal{M}_m)}{P(\mathcal{M}_l)} \frac{P(\mathbf{Z}_{\neg m}|\mathbf{Z}_m, \mathcal{M}_m)}{P(\mathbf{Z}_{\neg l}|\mathbf{Z}_l, \mathcal{M}_l)} \frac{P(\mathbf{Z}_{test}|\mathbf{Z}_m, \mathcal{M}_m)}{P(\mathbf{Z}_{test}|\mathbf{Z}_l, \mathcal{M}_l)}\end{aligned} \tag{5}$$

We call $\frac{P(\mathbf{Z}_{\neg m}|\mathbf{Z}_m, \mathcal{M}_m)}{P(\mathbf{Z}_{\neg l}|\mathbf{Z}_l, \mathcal{M}_l)}$ the Training Factor because evaluates how well $\mathbf{Z}$ is predicted.

Analogously, we call the Testing Factor to $\frac{P(\mathbf{Z}_{test}|\mathbf{Z}_m, \mathcal{M}_m)}{P(\mathbf{Z}_{test}|\mathbf{Z}_l, \mathcal{M}_l)}$ because it measures the predictive ratio of the two models with respect to a new set of samples $\mathbf{Z}_{test}$.

Now, using (4) in (5) we can obtain:

$$\frac{P(\mathcal{M}_m|\mathbf{Z},\mathbf{Z}_{test})}{P(\mathcal{M}_l|\mathbf{Z},\mathbf{Z}_{test})} = \frac{P(k^{\kappa}|\mathcal{K}_m)}{P(k^{\kappa}|\mathcal{K}_l)} \frac{P(\mathbf{Z}_{\neg m}|\mathbf{Z}_m,\mathcal{M}_m)}{P(\mathbf{Z}_{\neg l}|\mathbf{Z}_l,\mathcal{M}_l)} \frac{P(\mathbf{Z}_{test}|\mathbf{Z}_m,\mathcal{M}_m)}{P(\mathbf{Z}_{test}|\mathbf{Z}_l,\mathcal{M}_l)} \qquad (6)$$

Finishing, if we need to compare M models, (6) can be generalized to:

$$\frac{P(\mathcal{M}_m|\mathbf{Z},\mathbf{Z}_{test})}{\sum_{l=1}^{M} P(\mathcal{M}_l|\mathbf{Z},\mathbf{Z}_{test})} = \frac{P(k^{\kappa}|\mathcal{K}_m)P(\mathbf{Z}_{\neg m}|\mathbf{Z}_m,\mathcal{M}_m)P(\mathbf{Z}_{test}|\mathbf{Z}_m,\mathcal{M}_m)}{\sum_{l=1}^{M} P(k^{\kappa}|\mathcal{K}_l)P(\mathbf{Z}_{\neg l}|\mathbf{Z}_l,\mathcal{M}_l)P(\mathbf{Z}_{test}|\mathbf{Z}_l,\mathcal{M}_l)} \qquad (7)$$

3 Material and Methods

3.1 Database and Corpora

An evaluation of the Bayesian approach consisting of ranking brain tumour classifiers is shown in this study. A multicenter multiproject evaluation is followed [7]. It is based on training classifiers with the data of a multicenter project, INTERPRET (2000-2002) [8,9], and evaluating with the data acquired in a subsequent multicenter project, eTUMOUR (2004-2009) [10]. Both projects followed an acquisition protocol to guarantee the compatibility of the signals accrued at the different participating centers [11].

Table 1. Description of the labels and the number of samples of the INTERPRET (2000-2002) and eTUMOUR (2004-2009) projects for each corpus. Labels have been defined according to WHO. $C7_{kids}$ contains only paediatric cases ($\leq$ 18 y.o.).

CORPUS	LABELS	INTERPRET	eTUMOUR
$C1$	common aggressive common grade II glial low grade meningioma	217	170
$C2$	aggressive grade II glial low grade meningioma	243	195
$C3$	grade I-II glial grade III-IV glial	140	158
$C4$	grade I-II grade III-IV	262	222
$C5$	affected healthy tissue	304	222
$C6$	grade I-II meningioma other brain tumours	282	222
$C7_{kids}$	aggressive non-aggressive	31	62
$C7_{adults}$	aggressive non-aggressive	240	151

304 samples from the INTERPRET database acquired from six international centers were used for training purposes. These samples were Single voxel (SV) ^{1}H magnetic resonance spectra (MRS) at 1.5T at Short echo time (TE, 20 or 30-32ms). Histopathological diagnoses and spectral quality were validated respectively by the INTERPRET Clinical Data Validation Committee and expert spectroscopists [8].

Ten international institutions in the framework of eTUMOUR provided a total amount of 222 cases with SV Short TE (20 or 30-32ms) MRS at 1.5T. Histopathological diagnoses were validated in the context of clinical setting and radiological images by the eTUMOUR Clinical Validation Committee.

The classes considered were based on the histological classification of the Central Nervous System tumours set up by the World Health Organization (WHO) [12].

16 corpora (8 for training, 8 for testing) were defined according to different criteria like the tumour type and grade or the age of the patients. Table 1 shows the labels associated to each corpus, always according to WHO, and the number of samples available from each project.

In order to reduce the dimensionality of the data managed, a peak integration (PI) to mean metabolites of the region of [0.5, 4.1] ppm interval was carried out. PI is a method successfully applied in [7] which allows a reduction from the whole spectra to 15 parameters, representing the concentration of the mean metabolites in each spectra.

3.2 Classifiers: Multilayer Perceptron Networks

This ranking framework can be applied to any classification technique able to express its answers as a probability. Classifiers based on Gaussian mixtures, linear discriminant analysis, support vector machines or k nearest neighbors can be applied because these techniques allow the calculation of the likelihood of a test case. Nevertheless, we have choose Multilayer perceptron networks (MPN) for the evaluation of the ranking because allow easy comparison of different model complexities (namely, network topologies).

MPN can be used to define probabilistic models for regression and classification tasks by using the outputs to define the conditional distribution for one or more targets, y_k as a function of several possible values of an input vector, x [13,14].

Fully connected MPN with different topologies have been selected for the evaluation of the ranking framework: $N1$, with one hidden layer of 8 units; $N2$, with one hidden layer of 15 units; $N3$, counting two hidden layers of 8 and 4 units respectively; and $N4$ with two hidden layers of 12 and 6 units. Backpropagation algorithm with Levenberg-Marquardt optimization was used to train the networks. The theorem of Ripley [15] indicates that a great amount of training samples (around 6.000) is needed if we want guarantee a minimum error of 0.25 for the most complex topology ($N4$). Nevertheless, this theorem is a pessimistic estimator and, usually, MPN are able to obtain better training results with much less samples.

4 Results and Discussion

We have trained 4 classification networks ($N1, \ldots, N4$) with each INTERPRET corpus (Table 1), counting a total of 32 candidates to be compared.

Table 2. Likelihood of the eTUMOUR data for the classifiers trained with INTERPRET data

	Method A				Method B			
	N1	N2	N3	N4	N1	N2	N3	N4
$C1$	0.716	0.754	0.706	0.665	0.724	0.727	0.704	0.681
$C2$	0.716	0.714	0.531	0.512	0.662	0.665	0.531	0.504
$C3$	0.764	0.743	0.758	0.764	0.700	0.670	0.650	0.688
$C4$	0.716	0.763	0.827	0.763	0.713	0.706	0.799	0.731
$C5$	0.971	0.952	0.980	0.980	0.930	0.953	0.973	0.968
$C6$	0.915	0.917	0.909	0.880	0.840	0.831	0.837	0.820
$C7_{kids}$	0.471	0.567	0.573	0.434	0.583	0.629	0.619	0.562
$C7_{adults}$	0.734	0.694	0.735	0.705	0.712	0.652	0.715	0.612

The mean likelihood obtained by each candidate is summarized in Table 2. In Method A, likelihood was calculated for a set of samples with labels common to every corpus. This set of labels can be seen in Table 3. Although Oligodendroglioma is uncommon in both adults and children, it was selected by Method A because it collects the labels included in all corpora regardless their prevalence. In contrast, likelihoods obtained with Method B were evaluated for the whole set of eTUMOUR; When a network was unable to classify a sample because it could not be categorized into the labels the network discriminated among, mean imputation was applied.

Networks trained with $C7_{kids}$ corpus showed a low likelihood compared to the others. This was because these networks were trained with a low number of paediatric samples and the likelihood had been calculated for a set of samples where adult cases predominated. The tumour types commonly seen in children with brain tumours differ from those in adults and this may explain the performance differences [16].

Although similar results were observed for both methods, $C7_{kids}$ results were better when Method B was applied. This is due to the mean imputation and the larger amount of paediatric cases collected when using the whole eTUMOUR data set as $\mathbf{Z}_{test}$. Method B slightly favours the performance of networks trained with samples of low prevalence.

Table 4 shows the top 10 ranking according to the evidence calculated for each network. The evidence is a combination of the likelihood and a penalty (often known as Occam's penalization) due to the complexity of the model measured in terms of number of parameters to be tuned. Since likelihoods obtained were very similar for each corpus, the Occam's penalization was determinant for establishing the sorting of the candidates.

Table 3. Number of samples obtained for $\mathbf{Z}_{test}$ (eTUMOUR) applying Method A

WHO Label	eTUMOUR	
	Children	Adults
DIFFUSE ASTROCYTOMA 9400/3	7	19
GLIOBLASTOMA 9440/3	4	65
OLIGODENDROGLIOMA 9450/3	2	5
TOTAL	13	89

Table 4. Top 10 ranking of the networks. Left ranking is obtained applying equal priors to each candidate. Center and right rankings are the result of assigning weights with relation to the similarity of the age of a test case (5 and 56 y.o. respectively) to the mean age of the training data of each candidate. Candidates are expressed in pairs of (corpus; network). Ranking coincidences are reflected by showing the candidate centered between the two methods.

<table>
<tr><th></th><th colspan="2">EQUAL PRIORS</th><th colspan="2">PROPER PRIORS (5 y.o. test)</th><th colspan="2">PROPER PRIORS (56 y.o. test)</th></tr>
<tr><th>RANKING</th><th>METHOD A</th><th>METHOD B</th><th>METHOD A</th><th>METHOD B</th><th>METHOD A</th><th>METHOD B</th></tr>
<tr><td>1^{st}</td><td colspan="2">$C5; N1$</td><td colspan="2">$C7_{kids}; N1$</td><td colspan="2">$C5; N1$</td></tr>
<tr><td>2^{nd}</td><td colspan="2">$C6; N1$</td><td colspan="2">$C5; N1$</td><td>$C6; N1$</td><td>$C7_{adults}; N1$</td></tr>
<tr><td>3^{rd}</td><td>$C3; N1$</td><td>$C4; N1$</td><td colspan="2">$C6; N1$</td><td>$C7_{adults}; N1$</td><td>$C6; N1$</td></tr>
<tr><td>4^{th}</td><td>$C7_{adults}; N1$</td><td>$C3; N1$</td><td colspan="2">$C3; N1$</td><td>$C3; N1$</td><td>$C4; N1$</td></tr>
<tr><td>5^{th}</td><td>$C4; N1$</td><td>$C7_{adults}; N1$</td><td colspan="2">$C4; N1$</td><td>$C4; N1$</td><td>$C3; N1$</td></tr>
<tr><td>6^{th}</td><td colspan="2">$C7_{kids}; N1$</td><td colspan="2">$C7_{adults}; N1$</td><td colspan="2">$C1; N1$</td></tr>
<tr><td>7^{th}</td><td colspan="2">$C1; N1$</td><td colspan="2">$C2; N1$</td><td colspan="2">$C2; N1$</td></tr>
<tr><td>8^{th}</td><td colspan="2">$C2; N1$</td><td colspan="2">$C1; N1$</td><td colspan="2">$C7_{kids}; N1$</td></tr>
<tr><td>9^{th}</td><td colspan="2">$C5; N3$</td><td colspan="2">$C7_{kids}; N3$</td><td colspan="2">$C5; N3$</td></tr>
<tr><td>10^{th}</td><td colspan="2">$C6; N3$</td><td colspan="2">$C5; N3$</td><td colspan="2">$C6; N3$</td></tr>
</table>

The ranking obtained when equal priors were assigned to each of the 32 candidates can be seen in Table 4, left. Observe also that $C7_{kids}; N1$, though presenting a low likelihood compared with others, was preferred due to its simplicity (it is ranked as 6^{th}).

Center and right rankings were the result of applying weights to each candidate according to the similarity of the age of a test patient with respect to the mean age of the samples used for training each network. This weightening can be appreciated in the results: When ranking the candidates for a patient of 5 y.o. (center ranking) both methods assigned as a first option a network trained with children cases ($C7_{kids}$). Similarly, when applied for a test case of 56 y.o. (right ranking), the ranking model assigned as first option networks where the mean age of their training corpora was greater.

Althought minor differences are observed in the ranking obtained with each method, both behave as expected, giving a ranking balanced between the similarity of the training data to the test case and the candidate performances.

This ranking framework can be used in any DSS offering support from a large set of classifiers similarly to the way a Web search engine works. The user can have access to the first top 10 classifier results or totally rely on the ranking and consult only the result of the first candidate.

5 Concluding Remarks

This study has examined a ranking model for classifiers for brain tumour classification from a Bayesian perspective. This ranking model evaluates the performance of the classifiers to be compared even if they are inferred from different sets of data. Additionally, we assign a prior to each model based on a measure of similarity of the training data to a test case.

The main conclusion from this test is that the ranking model based on a Bayesian approach can be effectively applied to the real problem of selecting classifiers for brain tumour classification.

Although evaluation with larger datasets is necessary, our results are very promising. Moreover, the ranking model we present is general-purpose and can easily be applied to real data in diagnostically challenging areas.

References

1. González-Vélez, H., et al.: HealthAgents: Distributed multi-agent brain tumour diagnosis and prognosis. Applied Intelligence (in press, 2007), doi:10.1007/s10489-007-0085-8
2. Saez, C., et al.: A generic decision support system featuring an assembled view of predictive models for magnetic resonance and clinical data. In: ESMRMB 2008: 25th Annual Scientific Meeting, Springer, Heidelberg (2008)
3. MacKay, D.J.C.: Bayesian interpolation. Neural Computation 4, 415–447 (1992)
4. Kass, R.E., Raftery, A.E.: Bayes factors. Journal of the American Statistical Association 90, 773–795 (1995)
5. Mackay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge (2003)
6. Liddle, A.R.: Information criteria for astrophysical model selection. MON.NOT.ROY.ASTRON.SOC.LETT. 377, L74 (2007)
7. García-Gómez, J.M., et al.: Multiproject-multicenter evaluation of automatic brain tumor classification by magnetic resonance spectroscopy. Magnetic Resonance Materials in Physics, Biology and Medicine (2008) (Epub ahead of print)
8. Tate, A.R., et al.: Development of a decision support system for diagnosis and grading of brain tumours using in vivo magnetic resonance single voxel spectra. NMR Biomed. 19, 411–434 (2006)
9. INTERPRET Consortium: INTERPRET web site. Technical report, IST-1999-10310, EC (accessed March 23, 2009), `http://azizu.uab.es/INTERPRET`
10. eTumour Consortium: eTumour: Web accessible MR Decision support system for brain tumour diagnosis and prognosis, incorporating in vivo and ex vivo genomic and metabolomic data. Technical report, FP6-2002-LIFESCIHEALTH 503094, VI framework programme, EC (accessed March 23, 2009), `http://www.etumour.net`
11. van der Graaf, M., et al.: MRS quality assessment in a multicentre study on MRS-based classification of brain tumours. NMR Biomed. 21, 148–158 (2008)
12. Louis, D.N., Ohgaki, H., Wiestler, O.D., Cavenee, W.K.: WHO classification of tumours of the central nervous system. IARC Press, Lyon, France (2007)
13. Rumelhart, D., Hinton, G., Williams, R.: Learning representation by back-propagation errors. Nature 323, 533–536 (1986)
14. Neal, R.M.: Bayesian Learning for Neural Networks, 1st edn. Lecture Notes in Statistics. Springer, Heidelberg (1996)
15. Ripley, B.D.: Statistical aspects of neural networks. In: Networks and Chaos: Statistical and Probabilistic Aspects, Chapman and Hall, Boca Raton (1993)
16. CBTRUS (2007): 2007-2008 statistical report: Primary brain tumors in the united states, 1998-2002. Technical report, Central Brain Tumor Registry of the United States (2007)

Feature Selection with Single-Layer Perceptrons for a Multicentre ^{1}H-MRS Brain Tumour Database

Enrique Romero[1], Alfredo Vellido[1], and Josep María Sopena[2]

[1] Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya
{eromero,avellido}@lsi.upc.edu
[2] Laboratori de Neurocomputació, Universitat de Barcelona
jsopena@ub.edu

Abstract. A Feature Selection process with Single-Layer Perceptrons is shown to provide optimum discrimination of an international, multi-centre ^{1}H-MRS database of brain tumors at reasonable computational cost. Results are both intuitively interpretable and very accurate. The method remains simple enough as to allow its easy integration in existing medical decision support systems.

1 Introduction

Diagnostic decision making in brain oncology is, for rather obvious reasons, an extremely sensitive matter. Much of the responsibility of brain tumour diagnostic decisions ultimately rest on the expert clinician's shoulders. Taking into account that most diagnostic techniques have to be non-invasive in this domain, clinicians might benefit from the use of an at least partially automated computer-based medical decision support system (MDSS) that embedded data mining processes. The reluctance of a clinician to seek the support of a computer-based MDSS should not be underestimated, though, as exemplified by the few products of this kind reaching mainstream medical practice [1]. This makes simplicity and robustness compulsory requirements for the employed data analysis methods.

In this study, we analyze Magnetic Resonance Spectroscopy (MRS) brain tumor data from the INTERPRET European research project [2]. The original database contains records corresponding to many brain tumour pathologies (many of which are represented by a very small sample of cases) and even to healthy brain tissue. This makes their computer-based automated classification a non-trivial undertaking that must be carefully designed. Most importantly, the high dimensionality of the data also precludes the straightforward interpretation of the obtained results, limiting their usability in a practical medical context, in which interpretability is paramount.

In the case of MRS data mining, one way to comply simultaneously with the aforementioned simplicity and interpretability requirements is through dimensionality reduction and, more specifically, through feature selection (FS). In this paper a FS procedure with Single-Layer Perceptron (SLP) classifiers that yields very accurate results with a parsimonious subset of interpretable spectral MRS frequencies is described. The method is based on the hypothesis that irrelevant features produce smaller variations in the output values than relevant ones. Hence, a natural way of comparing the

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1013–1020, 2009.

© Springer-Verlag Berlin Heidelberg 2009

relevance of two features is by comparing the absolute values of the derivatives of the output function with respect to the corresponding input units in the trained model. For SLP, the variation (in absolute value) of the output function is smaller for input features with smaller weights (in absolute value). Therefore, the magnitude of the weight can be considered as an indicator of its importance.

A backward selection technique was used as search procedure: starting from the complete set of available features, several features are removed at every step. The number of features removed at every step is a parameter of the system that controls the granularity of the selection and its computational cost. Under the hypothesis that many of the features are not necessary to obtain a good classification performance (which is a reasonable hypothesis for the analyzed MRS data set), an aggressive strategy structured in three phases was designed to save computational time: in a first phase, 50% of the features were removed at every step of the backward selection procedure. In a second phase, a 20% of the remaining features were removed. Finally, in a third phase, features were removed one by one.

The experiments with ^{1}H-MRS data reported in the following sections validate the usefulness of the described method, since the results presented are, to the best of our knowledge, the best reported up to date using this database. In addition, most of the selected features have a direct interpretation in terms of metabolites and molecules often mentioned in the MRS literature as descriptors of brain tumor pathologies.

2 Description of the INTERPRET ^{1}H-MRS Brain Tumour Data

The echo time is a determinant parameter in ^{1}H-MRS data acquisition. In short-echo time (SET) spectra some metabolites are better resolved (e.g. lipids, myo-inositol, glutamine and glutamate). However, there may be numerous overlapping resonances making the spectra difficult to interpret [3]. The use of a long-echo time (LET) in the acquisition of spectra yields less clearly resolved metabolites but also less baseline distortion, resulting in a more readable spectrum. LET data also allow a more reliable analysis and testing of classification methods [4].

In this study we consider three different data sets of single voxel ^{1}H-MR spectra, acquired *in vivo* from brain tumor patients. The first was acquired at SET, the second at LET and, finally, the third is a fusion of the two previous ones, where the spectra acquired at two echo times obtained for the same patient (when both echo times were available) are combined through straight concatenation. The clinically-relevant regions of both the SET and LET spectra were sampled to obtain 195 frequency intensity values.

Class (tumour pathology type) labelling was performed according to the World Health Organization (WHO) system for diagnosing brain tumors by histopathological analysis of a biopsy sample. For the reported experiments, spectra were bundled into three groups, namely: *G1*: low grade gliomas (astrocytomas grade II, oligoastrocytomas grade II and oligodendrogliomas grade II); *G2*: high grade malignant tumors (metastases and glioblastomas); and *G3*: meningiomas. In summary, three data sets were analyzed: SET (217 cases and 195 features), LET (195 cases and 195 features) and SET+LET (195 cases and 390 features).

3 Some Precedents of Feature Selection and Classification with INTERPRET ^{1}H-MRS Brain Tumour Database

Previous published work analyzing similar ^{1}H-MRS INTERPRET data used feature extraction (PCA, specifically) followed by LDA to distinguish between *high-grade malignant tumours* and *meningiomas*, obtaining a mean AUC (area under the ROC curve) of 0.94, using 6 principal components [5]. The same method was used to distinguish between *high-grade malignant tumours* and astrocytomas Grade II (part of the *low-grade gliomas* group), obtaining a mean AUC of 0.92, also using 6 principal components. Independent Component Analysis (ICA), an alternative feature extraction method, has also been applied to analyze an earlier version of the INTERPRET data in [6].

This type of binary classification, in different combinations, has been carried out with different versions of INTERPRET ^{1}H-MRS data and with varying degrees of success (see, for instance, the combination of spectrometric and imaging data in [7]). This is a somehow easier problem than the multi-class one that we deal with in this paper. In [8], this time using exactly the same three groups of tumours that we have analyzed in this study, a basic linear model (LDA) with 6 spectral frequencies (3.72, 3.04, 2.31, 2.14, 1.51 and 1.20 ppm) achieved a 83% of correct classification on an independent test set for LET, and a 89% correct classification for SET, using 5 frequencies (namely 3.76, 3.57, 3.02, 2.35 and 1.28 ppm). Similar results were found for LET data in [4] for a combination of PCA and LDA and for different versions of support vector machines.

Only a few recent works have addressed the problem of multi-class classification of ^{1}H-MRS data by combination of echo times. In [3], SET and LET data were combined in a classification problem that involved four groups of tumours: *high grade malignant tumours* and *meningiomas*, as in the current study, and astrocytomas grade II (part of our *low grade gliomas*) and anaplastic astrocytomas. Using LDA, the correct diagnosis was suggested by at least one of the echo times in 90% of all cases. A far more similar approach to the one followed in the current study (combining the SET and LET data through concatenation) was used in [9], also to classify *G1*, *G2* and *G3*: feature selection followed by LDA achieved 88.71% test accuracy, while PCA followed by LDA achieved a maximum of just over 90% test accuracy for 8 principal components.

4 Feature Selection Process

The problem of FS can be defined as follows: given a set of d features, select a subset that performs the best under certain evaluation measure. From a computational point of view, the definition of FS usually leads to a search problem in a space of 2^d elements. In this case, two components must be specified: the feature subsets evaluation measure and the search procedure through the space of feature subsets. If any of these two componentes depends on an external model, it must also be specified.

In the rest of the section, the constituent elements of the FS process for the ^{1}H-MRS brain tumour database are described.

4.1 Model

SLP Artificial Neural Networks (ANN) with sigmoidal output units were used both for the feature subsets evaluation measure (within the FS process) and to obtain the test accuracy (within the learning process). The number of output units was set to the number of classes of the problem. Therefore, the activation y_j of the output unit j for a d-dimensional input vector x is computed as

$$y_j = g\left(\sum_{k=1}^{d} x_k \cdot \omega_{jk} + b_j\right), \tag{1}$$

where ω_{jk} is the weight that connects the input unit k with the output unit j, b_j is the bias of the output unit j, and $g(z)$ is a sigmoidal function. The SLP were trained in this study so as to minimize the sum-of-squares error.

There are several reasons for using SLP instead of more complex ANN alternative models in this particular case. FS with Multi-Layer Perceptrons (MLP) would be computationally too expensive for the number of features of the ^{1}H-MRS brain tumour data set [10]. In addition, MLP parameters are more difficult to adjust. Alternatively, FS with linear Support Vector Machines (SVM) [11] usually computes the saliency of the features as a function of the weights, as in our model (see below). However, the weights of a SLP are not necessarily a linear combination of the data, as for linear SVM. Therefore, the saliency of every feature is likely be more independent for SLP than for linear SVM. In addition, linear models had shown quite good performance with these data in previous studies [8].

4.2 Feature Subsets Evaluation Measure

The evaluation measure (the relevance) of a feature subset was computed as the sum of the individual saliencies of its features. The saliency s_i of a feature i over O outputs was computed as: $s_i = \sum_{j=1}^{O} |\hat{\omega}_{ji}|$, where $\hat{\omega}_{ji}$ are the weights of the trained SLP.

This method is based on the hypothesis that irrelevant features produce smaller variations in the output values than relevant ones. Hence, a natural way to compare the relevance of two features is to compare the absolutes values of the derivatives of the output function with respect to their respective input units in the trained model.

Formally, the derivative in the trained model of the output function y_j in (1) with respect to an input feature x_i is

$$\frac{\partial y_j}{\partial x_i} = g'\left(\sum_{k=1}^{d} x_k \cdot \hat{\omega}_{jk} + b_j\right) \cdot \hat{\omega}_{ji},$$

and, for every j,

$$\frac{|\partial y_j / \partial x_{i_1}|}{|\partial y_j / \partial x_{i_2}|} = \frac{|\hat{\omega}_{ji_1}|}{|\hat{\omega}_{ji_2}|}.$$

Therefore, the variation (in absolute value) of the output function is smaller for input features with smaller weights (in absolute value), and they are the main candidates to be eliminated in a FS process. In summary, for linear discriminant functions such as SLP, the magnitude of the weights corresponding to a feature is considered as an indicator of its importance. Similar ideas can be found elsewhere (see, for example, [11] or [12]).

4.3 Search Procedure

A backward selection procedure was used as an iterative selection process guided by the previously defined saliency measure. Starting from the complete set of available features, a subset of them was removed at every step of the algorithm according to the evaluation measure. Since the evaluation measure of a feature subset is computed as the sum of the saliencies of its features, the features to be removed at every step are those with the smallest saliency. The number of features removed at every step is a parameter of the system, that controls the granularity of the selection and the computational cost. The main reason for choosing a backward procedure instead of a forward one is twofold: First, a backward procedure allows, at the onset, to take into account all the interactions among variables. Second, the parameters of the SLP are easier to adjust.

4.4 Algorithm

The FS algorithm finally applied in this study consisted of three general phases:

1. Perform a backward selection procedure (see section 4.3), starting with the whole set of features, until only one feature remain. At every step:
   (a) Train a SLP with the remaining features (see section 4.1).
   (b) Compute the saliency of every feature (see section 4.2).
   (c) Remove the 50% of the remaining features with the lowest saliency.
   For every feature subset obtained, estimate its generalization performance. Out of all the results, keep the previous to the best one for the next phase (to avoid missing a possible generalization maximum in intermediate, not analyzed, subsets).
2. The second phase is similar to the first one, except for:
   (a) The initial feature subset is the one obtained in the first phase.
   (b) At every step, 20% of the remaining features are removed.
3. The third phase is similar to the second one, except for:
   (a) The initial feature subset is the one obtained in the second phase.
   (b) At every step, one feature is removed.

5 Experiments

5.1 Experimental Setting

No preprocessing of the data was done, since all the features were in the same range of values. The target values were generated with a 1-of-C coding scheme (a value of 1 for the correct class, an 0 for all the others).

The SLP models were trained with the Delta rule in on-line mode for $10,000$ epochs. The logistic function $g(z) = \frac{1}{1+e^{-z}}$ was used for the output units. Initial weights and momentum were set to 0. Learning rates were heuristically adjusted.

The saliencies were calculated using the whole data set, and five runs of a 5-fold stratified Cross-Validation (CV) were performed to estimate the generalization performance. Prior to every CV, the data were randomly shuffled. The complete experiment took around 7 hours in an Intel Xeon CPU at 2,000 MHz.

Table 1. Classification and FS results

Data Set	Test	NF	Features Selected (ppm)
SET	95.72%	29	3.13 1.51 4.15 1.65 3.74 3.60 3.51 1.30 1.82 3.45 2.31 2.22 2.27 3.32 3.77 1.87 0.94 3.81 3.24 2.29 2.03 1.97 1.47 1.63 3.56 3.93 2.23 1.59 3.34
SET	90.51%	18	1.51 3.13 2.22 2.27 3.56 3.60 3.45 3.51 1.87 2.29 2.03 1.47 1.59 1.97 3.74 4.15 3.77 1.30

Data Set	Test	NF	Features Selected (ppm)
LET	95.79%	50	1.23 2.16 2.27 2.33 3.36 0.87 3.72 1.42 2.99 2.39 3.09 3.05 1.76 0.54 2.88 1.53 1.44 2.52 2.54 2.56 3.70 3.64 3.32 1.32 3.91 2.94 1.06 3.20 3.55 3.85 1.59 3.53 3.18 3.79 1.04 0.64 3.26 2.48 0.73 1.27 1.51 3.37 3.45 3.03 1.84 4.19 1.91 3.39 3.94 0.89
LET	90.26%	8	2.27 2.16 1.23 2.99 2.88 0.87 3.72 1.76

Data Set	Test	NF	Features Selected (ppm)
SET+LET	98.46%	18	L2.88 L2.27 S0.89 S3.58 L2.03 L2.54 L3.64 L1.59 S1.32 L3.79 S3.77 L1.84 S3.45 L3.70 S1.25 S2.12 L0.70 S3.18
SET+LET	91.08%	9	L2.27 L2.88 S2.12 L1.59 L2.54 L3.79 S3.58 S1.25 L2.03

5.2 Results

Classification and FS results of the experiments for the three data sets studied are summarized in Table 1 (SET: top, LET: middle, and SET+LET: bottom). For the combination of echo times, SET+LET, up to a 98.46% average test accuracy (an average of only 3 misclassifications out of 195 spectra) was achieved using a parsimonious subset of only 18 spectral frequencies, 8 of them belonging to SET and 10 to LET. The results reported in the second rows of Table 1 illustrate how the test accuracy deteriorates as we detract features from the selected subset. Nevertheless, for the SET+LET data set, a selection of 9 spectral frequencies (a mere 2.3% of the original 390) was still able to retain over 91% of the average test classification accuracy in the multi-class problem involving *low grade gliomas*, *high grade malignant tumors*, and *meningiomas*.

For illustration, the 18 selected spectral frequencies are displayed, together with the mean spectra of the three tumour classes, in Figure 1, both for SET (top) and LET (bottom) data separately. Many of them have a clear interpretation as resonances of metabolites and molecules often reported in the MRS literature as descriptors of brain tumor pathologies. They include, at SET: Alanine (^{2}CH-group) at 3.77 ppm, Glycine (^{2}CH$_2$-group) or Myo-Inositol at 3.58, possibly Taurine at 3.45, Choline and other trimethylamine-containing compounds at 3.18, Glutamate and Glutamine (^{3}CH$_2$- and ^{4}CH$_2$- groups) at 2.12, Lactate and Lipids at 1.32 and 1.25, and also lipids at 0.89. They also include, at LET: Glutamate/Glutamine-containing compounds (^{2}CH-groups) at 3.79 and, possibly, 3.70, Glutamate and Glutamine metabolites (this time $^4CH_2$-groups) at 2.54 and 2.27, and N-acetylaspartate and other N-acetyl-containing compounds at 2.03. All this metabolic information should provide medical experts with intuitive insights on the diagnoses of the analyzed brain tumour pathologies.

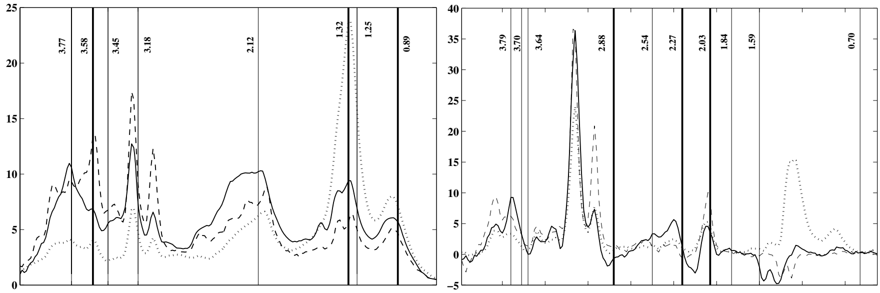


Fig. 1. Representation of the 18 selected spectral frequencies from the SET+LET data set as vertical lines, with their value in ppm tagged by their side. Left plot: SET results; right plot: LET results. The 3 most relevant frequencies of each echo time are represented with thicker lines. Mean spectra of each class are represented as dashed lines (low grade gliomas), dotted lines (high grade malignant tumors), and solid lines (meningiomas).

The use of a single echo time yields a 95.72% average test accuracy (an average of between 8 and 9 misclassifications out of 195 spectra) for SET using 29 spectral frequencies and a very similar 95.79% for LET, this time using a less compact subset of 50 selected frequencies. According to these results, a classification based on SET only might be preferred to one based on LET only (on the grounds of easier interpretation), although LET reaches a rather good compromise between accuracy and interpretability with a 90.26% result using only 8 frequencies. In any case, according to the results reported in Table 1, models using SET+LET instead of a single echo time should always be the choice in order to achieve optimal generalization.

Furthermore, our results using SET+LET to discriminate between *low grade gliomas*, *high grade malignant tumors*, and *meningiomas* with a combination of FS and SLP improve on those reported in [9] for the same problem (using stepwise FS with LDA), by almost a 10% average test accuracy. Our results using only SET and LET also improve on those previously reported for the same problem in [8,4,3].

6 Conclusions and Future Work

A simple SLP with FS has achieved a near perfect classification (measured by test accuracy) of an international, multi-centre ^{1}H-MRS data set by combining data acquired at two different echo times. This performance is better than that achieved using data obtained at any of the echo times separately, reinforcing previous results [9]. In future research, out-of-sample data should be acquired in order to fully ensure the generalization capabilities of the model. Importantly, these good results are obtained while retaining model simplicity and interpretability, as they only require less than 5% of the original frequencies (most of them identifiable as known descriptors of brain tumor pathologies).

The FS procedure was designed with an aggressive strategy that allowed to save computational time, making it feasible for data sets with a large number of variables. However, for highly correlated features the values of the weights (and therefore the saliency) may be distributed in the trained SLP, leading to unsuitable FS. In this case, the FS process would benefit from a previous filter procedure where highly correlated features were considered as a single one.

Acknowledgements

Authors gratefully acknowledge the former INTERPRET (EU-IST-1999-10310) European project partners. Data providers: Dr. C. Majós (IDI), Dr. À.Moreno-Torres (CDP), Dr. F.A. Howe and Prof. J. Griffiths (SGUL), Prof. A. Heerschap (RU), Dr. W. Gajewicz (MUL) and Dr. J. Calvar (FLENI); data curators: Dr. A.P. Candiota, Dr. M. Julià-Sapé, Ms. T. Delgado, Ms. J. Martín, Mr. I. Olier and Mr. A. Pérez (all from GABRMN-UAB). GABRMN-UAB coordinator: Prof. C. Arús. The authors acknowledge funding from M.E.C. research project TIN2006-08114. We also acknowledge the use of the UPC MA1 Department computing cluster system (http://www.ma1.upc.edu/eixam/index.html).

References

1. Vellido, A., Lisboa, P.J.G.: Neural networks and other machine learning methods in cancer research. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 964–971. Springer, Heidelberg (2007)
2. INTERPRET: International Network for Pattern Recognition of Tumours Using Magnetic Resonance project, `http://azizu.uab.es/INTERPRET`
3. Majós, C., Julià-Sapé, M., Alonso, J., Serrallonga, M., Aguilera, C., Acebes, J.J., Arús, C., Gili, J.: Brain tumor classification by proton MR spectroscopy: Comparison of diagnostic accuracy at short and long TE. American Journal of Neuroradiology 25, 1696–1704 (2004)
4. Lukas, L., et al.: Brain tumor classification based on long echo proton MRS signals. Artificial Intelligence in Medicine 31, 73–89 (2004)
5. Devos, A.: Quantification and Classification of MRS Data and Applications to Brain Tumour Recognition, PhD thesis, Katholieke Univ., Leuven, Belgium (2005)
6. Huang, Y., Lisboa, P.J.G., El-Deredy, W.: Tumour grading from Magnetic Resonance Spectroscopy: A comparison of feature extraction with variable selection. Statistics in Medicine 22, 147–164 (2003)
7. Luts, J., Heerschap, A., Suykens, J.A.K., Van Huffel, S.: A combined MRI and MRSI based multiclass system for brain tumour recognition using LS-SVMs with class probabilities and feature selection. Artificial Intelligence in Medicine 40, 87–102 (2007)
8. Tate, A., et al.: Development of a decision support system for diagnosis and grading of brain tumours using in vivo magnetic resonance single voxel spectra. NMR in Biomedicine 19, 411–434 (2006)
9. García-Gómez, J.M., Tortajada, S., Vidal, C., Julià-Sapé, M., Luts, J., Moreno-Torres, A., Van Huffel, S., Arús, C., Robles, M.: The influence of combining two echo times in automatic brain tumor classification by Magnetic Resonance Spectroscopy. NMR in Biomedicine (2008) (accepted for publication)
10. Romero, E., Sopena, J.M.: Performing feature selection with Multi-Layer Perceptrons. IEEE Transactions on Neural Networks 19, 431–441 (2008)
11. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.N.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
12. Steppe, J.M., Bauer, K.W.: Feature saliency measures. Computer & Mathematics with Applications 33, 109–126 (1997)

Weakly-Supervised Classification with Mixture Models for Cervical Cancer Detection

Charles Bouveyron

SAMOS-MATISSE, CES, UMR CNRS 8174
Université Paris 1 (Panthéon-Sorbonne), Paris, France
`charles.bouveyron@univ-paris1.fr`

Abstract. The human supervision is required nowadays in many scientific applications but, due to the increasing data complexity, this kind of supervision has became too difficult or expensive and is no longer tenable. This paper therefore focuses on weakly-supervised classification which uses contextual informations to label the learning observations and to build a supervised classifier. This new kind of classification is treated in this work with a mixture model approach. For this, the problem of weakly-supervised classification is recasted in a problem of supervised classification with uncertain labels. The proposed approach is applied to cervical cancer detection for which the human supervision is very difficult and promising results are observed.

1 Introduction

The supervised classification is used nowadays in many applications requiring a decision: optical character recognition, image classification or cancer detection. All existing approaches for supervised classification require a human supervision of all learning observations in order to build the classifier which is then used to classify new observations (see [7]). However, in many applications, this kind of supervision is either difficult or expensive. For instance, in bio-medical applications (cervical cancer detection, MRI images, DNA micro-array, ...), domain experts are asked to manually label a sample of learning data which are then used for building the supervised classifier. In such applications, the cost of the supervision phase is usually high due to the difficulty of labeling complex data. In addition, the performance of human experts is not constant over the time whereas automatic labeling is slightly less efficient but remains stable. Figure 1 illustrates this phenomenon.

To face with the increasing difficulty of labeling learning data, a new kind of classification is emerging: the weakly-supervised classification. The idea of such a strategy is to used informations which are usually easy to obtain instead of asking an expert to label a very large set of learning data. Weakly-supervised classification aims to build a supervised classifier able to discriminate several classes using a set of learning observations and weak informations about their belongings to the classes. The weak informations are usually provided by the context in which the observations have been collected. For instance, each pixel of

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1021–1028, 2009.
© Springer-Verlag Berlin Heidelberg 2009

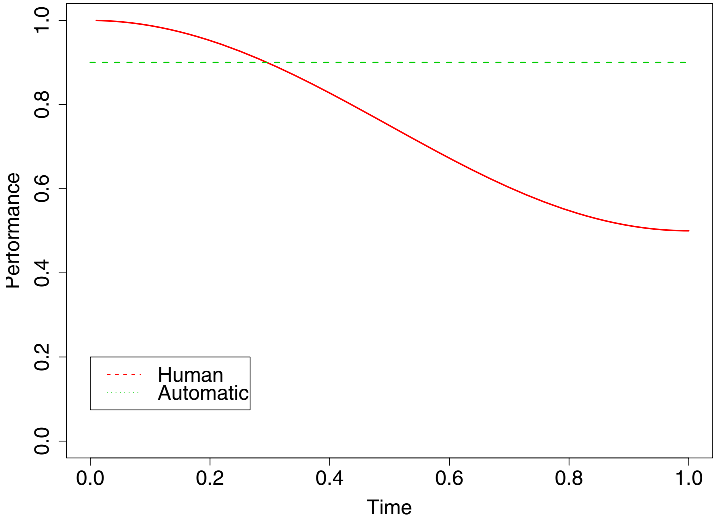


Fig. 1. Performance of a human expert over the time for a supervision task

an image containing a bike can be associated with the label "bike" even so several pixels of this image can belong to the background. Therefore, the additional difficulty compared to the traditional supervised classification is finding a way to extract a label for each learning observation from its associated contextual informations.

This paper is organized as follows. Section 2 first introduces the problem of weakly-supervised classification and then presents the mixture model which will be used in this work to learn a classifier under a weak supervision. Section 3 is devoted to the application of the proposed approach to cervical cancer detection and Section 4 proposes some concluding remarks.

2 Weakly-Supervised Classification with Mixture Models

This section introduces the problem of weakly-supervised classification and presents a weakly-supervised classification approach based on the mixture model of Robust Mixture Discriminant Analysis.

2.1 Our Approach of Weakly-Supervised Classification

In this work, we propose to closer the task of weakly-supervised classification to the supervised classification with label noise in which a supervised classifier is learned from data with uncertain labels. Recently, Bouveyron *et al.* proposed in [2] a method, called Robust Mixture Discriminant Analysis (RMDA), able to build a supervised classifier from data with uncertain labels. In order to be able to use this approach for weakly-supervised classification, we have to recast the problem of weakly-supervised classification in a problem of supervised classification with uncertain labels.

For the sake of clarity, we describe our approach in the binary classification framework (2 classes: positive and negative) but it can obviously be extented to the general multi-class classification case. We propose to roughly label the learning observations with their associated contextual informations: all the observations associated with the information "positive" are labeled as positive even so many of these observations do not belong to this class. In the same manner, all the observations associated with the information "negative" are labeled as negative. By doing that, we consciously introduce a label noise between both classes. However, it should be possible using RMDA to identify the actual label of each learning observation and, finally, build a supervised classifier able to classify correctly new observations.

2.2 Robust Mixture Discriminant Analysis

The idea of Robust Mixture Discriminant Analysis is to compare the supervised information carried by the labels with an unsupervised modelling of the data in order to detect inconsistent labels. For this, let us consider a multivariate mixture model in which two different structures coexist: an unsupervised structure of K clusters (represented by the random discrete variable S) and a supervised structure, provided by the learning data, of L classes (represented by the random discrete variable C). As in the standard mixture model, we assume that the data $(x_1, ..., x_n)$ are independent realizations of a random vector $X \in \mathbb{R}^p$ with density function:

$$p(x) = \sum_{k=1}^{K} P(S = k) p(x|S = k), \tag{1}$$

where $P(S = k)$ is the prior probability of the kth cluster and $p(x|S = k)$ is the conditional density of the kth cluster. Let us now introduce the supervised information carried by the learning data. Since $\sum_{\ell=1}^{L} P(C = \ell|S = k) = 1$ for all $k = 1, ..., K$, we can plug this quantity in (1) in order to obtain:

$$p(x) = \sum_{\ell=1}^{L} \sum_{k=1}^{K} P(C = \ell|S = k) \pi_k p(x; \alpha_k), \tag{2}$$

where $P(C = \ell|S = k)$ can be interpreted as the probability that the kth cluster belongs to the ℓth class and thus measures the consistency between classes and clusters. Using the classical notations of parametric mixture models and introducing the notation $r_{\ell k} = P(C = \ell|S = k)$, we can reformulate (2) as follows:

$$p(x) = \sum_{\ell=1}^{L} \sum_{k=1}^{K} r_{\ell k} \pi_k p(x; \alpha_k). \tag{3}$$

Therefore, (3) exhibits both the modelling part of our approach, based on a mixture model, and the supervision part through the parameters $r_{\ell k}$.

Estimation Procedure. Due to the nature of the model presented in the previous paragraph, the estimation procedure is made of two steps corresponding respectively to the unsupervised and to the supervised part of the comparison. In this first step of the estimation procedure, we do not use the labels of the data in order to form K homogeneous groups. Therefore, this step consists in estimating the parameters of the mixture and depends on the chosen mixture model. The most used mixture model is the Gaussian model [1] but one can prefer to use a HD Gaussian model [3] if the data are in high dimension. We therefore refer to [1,3,8] for inference aspects in the mixture model framework. The usual way to estimate mixture model parameters is to maximize the likelihood using the iterative Expectation-Maximization algorithm [4]. In this second step of the procedure, we introduce the labels of the learning data to estimate the parameters $r_{\ell k}$ and we use the parameters learned in the previous step for computing the posterior probabilities $P(S=k|X=x)$ through the Bayes' rule. From (3), the log-likelihood associated to our model can be expressed as:

$$\log(\mathcal{L}) = \sum_{\ell=1}^{L} \sum_{x \in \mathcal{C}_i} \log\left(\sum_{k=1}^{K} r_{\ell k} P(S=k|X=x) \right) + C^{ste}, \tag{4}$$

where C^{ste} does not depend on the parameters $r_{\ell k}$. Consequently, we end up with the following constrained optimization problem to solve:

$$\begin{cases} \text{maximize} & \log(\mathcal{L}), \\ \text{with respect to} & r_{\ell k} \in [0,1],\ \forall \ell = 1,...,L,\ \forall k = 1,...,K, \\ \text{and} & \sum_{\ell=1}^{L} r_{\ell k} = 1,\ \forall k = 1,...,K. \end{cases}$$

Since it is not possible to find an explicit solution to this optimization problem, an iterative optimization procedure has to be used to compute the maximum likelihood estimators of the parameters $r_{\ell k}$.

Classification Step. In model-based discriminant analysis, new observations are usually assigned to a class using the maximum a posteriori (MAP) rule which assigns a new observation x to the class for which x has the highest posterior probability. Therefore, the classification step mainly consists in calculating the posterior probability $P(C=\ell|X=x)$ for each class $\ell = 1,...,L$ which can be expressed as follows:

$$P(C=\ell|X=x) = \sum_{k=1}^{K} r_{\ell k} P(S=k|X=x). \tag{5}$$

As we can see, the probabilities $r_{\ell k}$, which quantify the consistency between the groups and the classes, balance the importance of the groups in the final classification rule. Consequently, the classifier associated with this decision rule will be mainly based on the groups which are very likely to be made of points from a unique class.

3 Application to Cervical Cancer Detection

This section presents an application of the weakly-supervised classification with mixture models to an important public health field: cervical cancer detection. Pap smear screening aims to identify potentially pre-cancerous changes in order to prevent the development of cancer. In developed countries, the widespread use of cervical screening programs has reduced the incidence of invasive cervical cancer by 50% or more. The pap smear screening is usually done by an human expert who screens each cell contained in a smear. This task is obviously extremely tiring due to the very large number of cells by smear (up to 20 000) and the very small proportion of cancer cells (less than 1%). We therefore investigate here the possibility of substituting the human supervision by a weak supervision. Figure 2 presents some cytological samples (thin preparation) with normal and abnormal cells.

3.1 The Data

The dataset comes from a study [5] on the performance of the DNA ploidy measurements on cervical nuclei for the detection of abnormal cells and has been provided by Dr. Guillaud from the British Columbia Cancer Agency. For our study, 20 samples of smears are used and comprise between 2,000 and 4,000 cells. Each nucleus is described by several features which may be divided into 3 categories: morphological, photometric or texture features. These nuclei can be divided in 2 classes: normal nuclei and abnormal (diseased) nuclei. The diseased population is really scarce in our dataset and represents only 0.5% of the studied population. The diagnosis of each cell has been provided by a gynecological pathologist. The learning dataset is made of 10 smears from healthy subjects and 7 smears from infected subjects for a total of 56 448 cells: 56 317 normal cells and 131 abnormal cells. The test dataset used for evaluating the performance of the classification methods is made of 3 smears from infected subjects for a total of 12 473 cells: 12 431 normal cells and 42 abnormal cells.

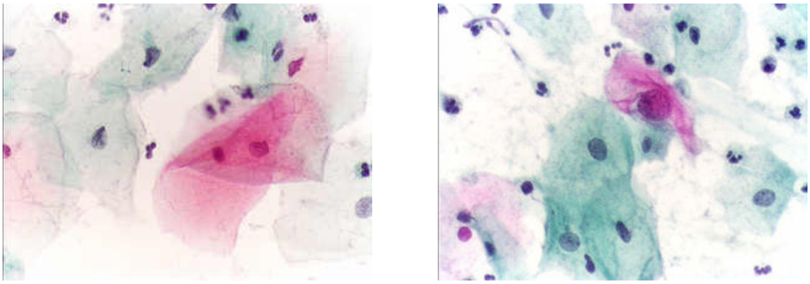


Fig. 2. Cytological specimen (thin preparation): smear with normal cells (left) and smear containing abnormal cells (right)

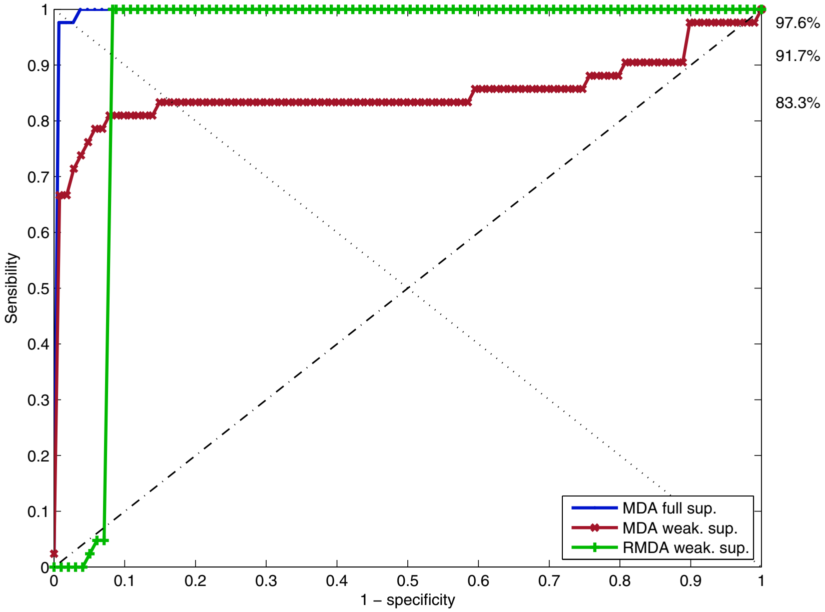


Fig. 3. ROC curves for MDA in both supervised and weakly-supervised cases and RMDA in the weakly-supervised case on the test dataset

3.2 Experimental Setup

For this experiment, actual labels of learning data have been replaced according the following rule: all observations (cells) belonging to a healthy smear are labeled as "normal" and all observations belonging to infected smears are labeled as "abnormal". Therefore, all observations labeled as "normal" are correct whereas a major part of the observations labeled as "abnormal" are incorrectly labeled. Our approach using RMDA in a weakly supervised context is compared to the supervised classification method MDA [6] based on mixture models as well. The method MDA is used in both supervised and weakly-supervised contexts. Therefore, the results of MDA in the supervised context (using the actual labels) will be used as the reference result. For the three methods in competition, the Gaussian mixture model is used and the total number of components K is fixed to 8 which is the number of components providing the best results for MDA in the supervised context.

3.3 Experimental Results

Figure 3 presents the Receiver Operating Characteristic (ROC) curves for the three studied approaches: MDA in the supervised case, MDA in the weakly-supervised case and RMDA in weakly-supervised case. On the first hand, it can be observed that MDA performs very well when the actual labels are used for

Table 1. Classification results for MDA in both supervised and weakly-supervised cases and RMDA in the weakly-supervised case on the test dataset

Method	AUC	Cut-off pt.	FN rate	FP rate
MDA full sup.	0.99872	0.9069	0	0.0037
MDA weak. sup.	0.85176	0.1344	0.1904	0.0765
RMDA weak. sup.	0.92222	0.3223	0	0.0830

learning (supervised context). This confirms that automatic strategies can be used to help doctors for screening the very large number of smears studied every day. On the other hand, MDA has a very disappointing behavior in the weakly-supervised context. It is however not surprising to observe such a behavior for MDA in this case since it gives a full confidence to the labels. Finally, the use of RMDA in the weakly-supervised context appears to be a good way to solve this problem. Indeed, the performance of RMDA in the weakly-supervised case are quite close to the one of MDA in the supervised case.

Table 1 provides for each of the three studied approaches the area under the ROC curve (AUC), the false negative (FN) and the false positive (FP) rates at the optimal cut-off point for the decision rule. However, the false negative and the false positive rates have a different importance is such a medical study. Indeed, it is important to not detect too many false positive cells (healthy cells) but it is crucial to detect every abnormal cell and to have the lowest FN rate. First, the AUC results confirm that RMDA in the weakly-supervised context is significantly more efficient than MDA in the same context and quite close to the performance of MDA in the supervised case. More importantly, it appears that MDA in the weakly-supervised context misses 19% of the potential cancer cells with a false detection rate equal to 7.65% whereas RMDA does not miss any abnormal cell with a comparable false detection rate (8.3%). It is therefore possible to conclude that RMDA could be used with a weak supervision for cancer detection.

4 Conclusion

This works has focused on the problem of the increasing difficulty of the human supervision in classification and has considered a new kind of classification, called weakly-supervised classification, which uses contextual informations to label the learning observations and build a classifier. This work has proposed to recast the problem of weakly-supervised classification in a problem of supervised classification with uncertain labels and to solve it using a mixture model approach. The proposed strategy has been applied to an important public health field: cervical cancer detection. Currently, cervical cancer detection is done by human experts eventually assisted by an automatic decision tool. Our experiments have shown that a weak supervision combined to the strategy proposed in this work allows to detect all abnormal cells with a satisfying false detection rate. In addition, this approach could be combined with an human supervision to increase again the cancer detection rate and help the doctors in their diagnosis.

References

1. Banfield, J., Raftery, A.: Model-based Gaussian and non-Gaussian clustering. Biometrics 49, 803–821 (1993)
2. Bouveyron, C., Girard, S.: Robust supervised classification with mixture models: learning from data with uncertain labels. Pattern Recognition (to appear, 2009)
3. Bouveyron, C., Girard, S., Schmid, C.: High-Dimensional Data Clustering. Computational Statistics and Data Analysis 52(1), 502–519 (2007)
4. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39(1), 1–38 (1977)
5. Guillaud, M., Benedet, J.L., Cantor, S.B., Staerkel, G., Follen, M., MacAulay, C.: Dna ploidy compared with human papilloma virustesting (hybrid capture II) and conventional cervical cytology as a primary screening test for cervical high-grade lesions and cancer in 1555 patients with biopsy confirmation. Cancer 107(2) (2006)
6. Hastie, T., Tibshirani, R.: Discriminant analysis by Gaussian mixtures. Journal of the Royal Statistical Society B 58, 155–176 (1996)
7. McLachlan, G.: Discriminant Analysis and Statistical Pattern Recognition. Wiley, New York (1992)
8. McLachlan, G., Peel, D.: Finite Mixture Models. John Wiley & Sons, New York (2000)

Edges Detection of Clusters of Microcalcifications with SOM and Coordinate Logic Filters

J. Quintanilla-Domínguez[1], B. Ojeda-Magaña[1], J. Seijas[1],
A. Vega-Corona[2], and D. Andina[1]

[1] Universidad Politécnica de Madrid, Spain
joelq@salamanca.ugto.mx
[2] Universidad de Guanajuato, México

Abstract. Breast cancer is one of the leading causes to women mortality in the world. Clusters of Microcalcifications (MCCs) in mammograms can be an important early sign of breast cancer, the detection is important to prevent and treat the disease. Coordinate Logic Filters (CLF), are very efficient in digital signal processing applications, such as noise removal, magnification, opening, closing, skeletonization, and coding, as well as in edge detection, feature extraction, and fractal modelling. This paper presents an edge detector of MCCs in Regions of Interest (ROI) from mammograms using a novel combination. The edge detector consist in the combination of image enhancement by histogram adaptive technique, a Self Organizing Map (SOM) Neural Network and CLF. The experiment results show that the proposed method can locate MCCs edges. Moreover, the proposed method is quantitatively evaluated by Pratt's figure of merit together with two widely used edge detectors and visually compared, achieving the best results.[1]

1 Introduction

Breast cancer is one of the most dangerous types of cancer among women around the world. Early detection of breast cancer is essential in reducing life loss. Currently the most effective method for early detection and screening of breast cancers is mammography. However, achieving this early cancer detection is not an easy task. Although the most accurate detection method in medical environment is biopsy, it is an aggressive, invasive procedure that involves some risks, patient's discomfort and high cost. Microcalcifications (MCs) and masses are two important early signs of the diseases. The MCs are tiny deposits of calcium in breast tissue. The MCs appear in the small clusters of a few pixels with relatively high intensity and closed contours compared with their neighboring pixels.

---

[1] This research has been partially supported by National (MICINN) and Madrid (CAM) Spanish institutions under the following projects: PTFNN (MCINN ref: AGL2006-12689/AGR), CES3D (CAM ref: CCG07-UPM/AMB-1998) and GASC/UPM (CAM ref: CCG07-UPM/000-1995) and Consejo Nacional de Ciencia y Tecnología (CONACYT) of México.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1029–1036, 2009.
© Springer-Verlag Berlin Heidelberg 2009

MCCs are primary indicators of malignant types of breast cancer, the detection is important to prevent and treat the disease. But it is still a hard work to detect all the MCs due to the fact that, in mammograms there is a poor contrast between MCs and the tissue around them. However, many studies have been focused on the general issue of detection of MCs in mammograms, using as a tool neural networks, wavelets, support vector machines, mathematical morphology, bayesian image analysis models, high order statistic, fuzzy logic, etc. Yuan Wu [1] presents an approach for detecting MCs employing a dual-threshold and LoG edge detection method. Gholamali Rezai-rad and Sepehr Jamarani [2] present an approach for detecting MCs in mammograms employing combination of Artificial Neural Networks (ANN) and wavelet-based subband image decomposition. Sung-Nien Yu *et al.* [3] developed a Computer-Aided Diagnosis (CAD) system for detection of MCs in mammograms. Their work is divided in two stages. First, all suspicious MCs are preserved by thresholding a filtered mammogram via a wavelet filter according to the Mean Pixel Value (MPV) of that image. Secondly, Markov random field parameters based on the Derin-Elliott model are extracted from the neighborhood of every suspicious MCs as the primary texture features. Both Bayes classifier and backpropagation neural network were used for computer experiments. Lixin Song *et al.* [4] apply mathematical morphology and wavelet transform to locate the MCs in digital mammogram.

Vega-Corona *et al.* [5]. propose and test a method for the detection of MCs in digital mammography. The method combines selections of Region of Interest (ROI) where MCs were diagnosed, enhanced of the image by histogram adaptive techniques, processing by multiscale wavelet and gray level statistical techniques, clustering and labelling of suboptimal feature vectors applying an unsupervised statistical method base on improved K-means algorithm and a neural feature selector based in a GRNN and detector based in a MLP to finally classify the MCs. Bhattacharya *et al.* [6]. propose a method based on discrete wavelet transform due to its multiresolution properties with the goal of segment MCs in digital mammograms. Morphological Tophat algorithm is applied for contrast enhancement of the MCCs. Finally fuzzy c-means clustering (FCM) algorithm has been implemented for intensity-based segmentation. Veni *et al.* [7] presents a method based in SUSAN edge detector and adaptive contrast thresholding technique and spatial filters for detection of MCs.

This paper is organized as follows: In section 2; model and theoretical background is presented; experimental results are presented in section 3; in the last section the conclusions are presented.

2 Model and Theoretical Background

In this section we will give an overview of the system for detection of MCCs edges in mammograms. Fig. 1 shows a block diagram of our system. In the first stage, many ROIs were selected from the mammogram image, then, in the second stage we perform the histogram analysis of the ROI and it is then processed. Where, the high bright values in the ROI image are enhanced and the

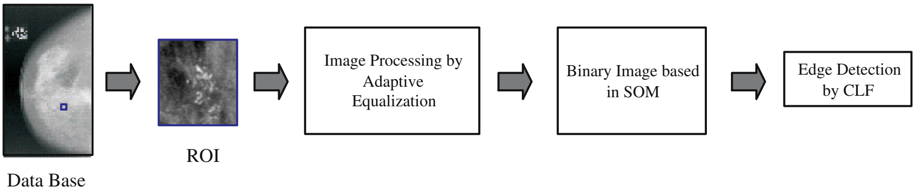

Data Base
ROI
Image Processing by Adaptive Equalization
Binary Image based in SOM
Edge Detection by CLF

Fig. 1. Blocks diagram to edge detection of MCCs

low bright values are diminished. In the next block we generate a binary image using a SOM Neural Network to obtain groups of label pixels corresponding to MCs and healthy tissue. In the last block, we apply an edge detector involving Coordinate Logic Filters (CLF).

2.1 Processing the Region of Interest

We achieve the analysis of the ROI images because the relevant information of MCs is concentrated in this area. MCs are relatively high-frequency components buried in the background of low-frequency components and very high-frequency noise in the mammograms. Image enhancement algorithms have been used for the improvement of contrast features and the suppression of noise. In this work we applied adaptive histogram enhancement, a technique widely used and well-established for the image enhancement [8]. This technique suppress pixel values of very small amplitude and enhance only those pixels that are larger than an determined threshold within each level of transform space. This is formulated with the following equation:

$$G(f) = \alpha[sigm(k(f-\beta)) - sigm(-k(f+\beta))] \qquad (1)$$

where

$$\alpha = \frac{1}{sigm(k(1-\beta)) - sigm(-k(1+\beta))} \qquad 0 < \beta < 1 \qquad (2)$$

where $f = f(x,y)$ is the gray value of a pixel at (x,y) of the input image and $sigm(x) = \frac{1}{1+e^{-x}}$. $\beta \in \mathbb{R}$ and $k \in \mathbb{N}$ are the threshold control and enhancement rate, respectively.

2.2 Binary Image Based in SOM

In previous phase, we applied an image enhancement technique with the idea of suppressing pixel values of very small amplitude and enhancing only those pixels that are larger than a certain threshold within each level of transform space. In this step, the ROI image is treated as binary image achieved through SOM Neural Network. The idea is that the ROI may be clustered in two possible classes, one class represents background and healthy tissue (Class 0) and the other one represent MCs (Class 1). In SOM structure each neuron is connected to input vector through a synaptic weight vector $w_i = [w_{i,1}, \ldots, w_{i,m}]$. When an input pattern is presented to the network, the best-matching (winning) neuron v is determined by minimizing the following cost function $v(x^{(q)}) = \min_i \|x^{(q)} - w_i\|$, $i = 1, \ldots, l$, where $x^{(q)}$, belongs to a m-dimensional input space, $\| \cdot \|$, denoting the Euclidean distance. Then, the synaptic weight vectors are updated as follow: $w_i^{(q+i)} = w_i^{(q)} + \eta(q) h_{i,v(x)}(q) \mid x^{(q)} - w_i^{(q)} \mid$, $i = 1, \ldots, l$, where $\eta(q)$ and $h_{i,v(x)}(q)$ are the learning rate and neighborhood function centered on the winner, respectively. Although the algorithm is simple, its convergence and accuracy depend of the selection of neighborhood function, the topology of the output space, a scheme for decreasing the learning rate parameter, and the total number of neuronal units [9].

2.3 Coordinate Logic Filters (CLF)

CLF constitute a class of nonlinear digital filters that are based on the execution of Coordinate Logic Operations (CLO). CLF are very efficient in digital signal processing applications, such as noise removal, magnification, skeletonization, coding, as well as in edge detection, feature extraction, fractal modelling and can also execute the morphological operations (erosion, dilation, opening and closing). The main feature of CLF is the direct execution of CLO between the pixel values without keeping any carry bit [10,11]. The CLO are the basic logic operations (NOT, AND, OR, and XOR, and their combinations) applied to corresponding individual binary values or pixels found within 2D signals (images). CNOT, CAND, COR, and CXOR represent the coordinate equivalents for each respectively as applied to multi-bit digital data [12]. Given a binary image G defined by $G = \{g(i,j); i = 1, 2 \ldots, M, j = 1, 2, \ldots, N\}$, the evaluation of a CLO (i.e. CXOR) between two images (here: *G1* and *G2*) is performed on a pixel-by-pixel basis and results in the output image F:

$$F = G_1 \,\text{CXOR}\, G_2 = \{g_1(i,j) \,\text{CXOR}\, g_2(i,j);\ i = 1, 2 \ldots, M,\ j = 1, 2, \ldots, N\} \quad (3)$$

CLF are the application of the CLO to a single image as dictated by a binary structuring element B. Since the dimensions of B are often much smaller in size than the original input image G, the resulting output represents local neighborhood characteristics of the image. A configuration for B, used in this work as in [12], is shown in (4),

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4)$$

Given structuring element B from (4) centered on input pixel *g(i,j)* (in image G), the output pixel *f(i,j)* (in image F) is calculated by:

$$f(i,j) = g(i-1,j) \,\mathbf{o}\, g(i,j-1) \,\mathbf{o}\, g(i,j) \,\mathbf{o}\, g(i+1,j) \,\mathbf{o}\, g(i,j+1) \quad (5)$$

where **o** represents any of the CLOs. Since the new state of each pixel depends only on the present state of that pixel and those of its neighbors, the new state for each pixel in the filtered image can be computed independently and simultaneously.

2.4 Edge Detection

Mertzios and Tsirikolias have presented the idea of using CLF for the purpose of edge extraction [10]. Danahy *et al.* in [12] introduces an alternative method for calculating CLF using Coordinate Logic Transforms (CLT). Moreover, a new edge detection technique is introduced, enhancing the capabilities of the basic CLT for the application of detecting edges within 2D signals. In this work we applied the edge detector proposed in [10] together with the combination of image enhancement and SOM Neural Network. Edge detection of MCCs in a ROI image with CLF can be achieved by:

$$F = [(G_B^{\mathrm{CAND}} \textsc{cxor}\ G) - (G_B^{\mathrm{COR}} \textsc{cxor}\ G)] \tag{6}$$

Where G is the resulting binary image after applying (1) and SOM Neural Network. G_B^{CAND} and G_B^{COR} represent the erosion and dilation of the image G respectively. Using the filter structure of (5), the erosion of the image G is given by:

$$f(i,j) = g(i-1,j)\,\textsc{cand}\,g(i,j-1)\,\textsc{cand}\,g(i,j)\,\textsc{cand}\,g(i+1,j)\,\textsc{cand}\,g(i,j+1) \tag{7}$$

and the dilatation is given by:

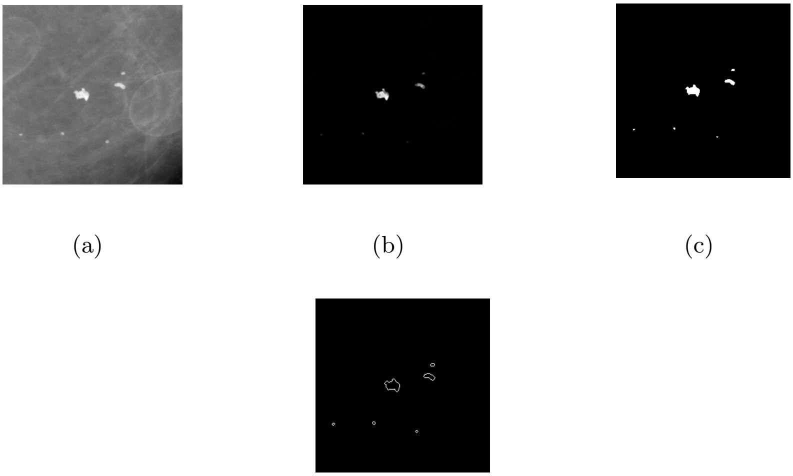
$$f(i,j) = g(i-1,j)\ \textsc{cor}\ g(i,j-1)\ \textsc{cor}\ g(i,j)\ \textsc{cor}\ g(i+1,j)\ \textsc{cor}\ g(i,j+1) \tag{8}$$

$G_B^{\mathrm{CAND}} \textsc{cxor}\ G$ represents the evaluation of a CLO (CXOR) between two images performed on a pixel-by-pixel.

3 Experimental Results

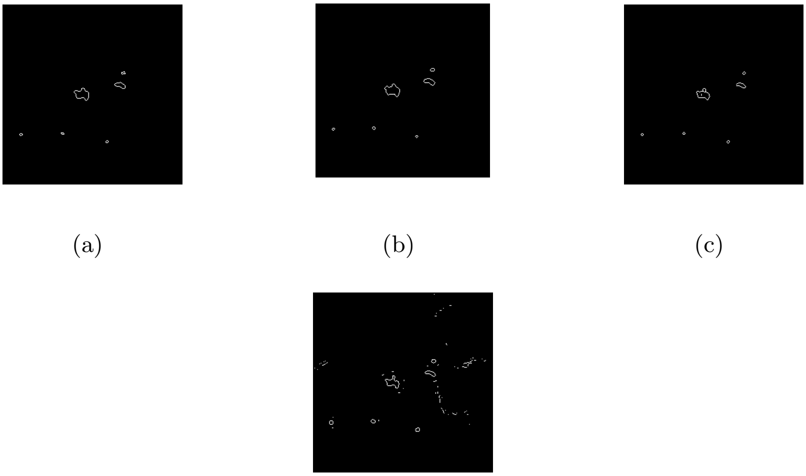
This section presents the results of our method in a quantitative way using Pratt's figure of merit defined in [13] and visual way. The digital mammogram database used in this paper is provided by the Digital Database for Screening Mammography (DDSM) of the University of South Florida [14]. In this work, we have selected 10 ROIs images from 20 cases mammograms diagnosed with MCs, of size of 256 × 256 pixels with resolution 43.5 microns at 12 bits by pixel. The ROI images are extracted out of the database with an overlay image previously marked by an expert. A test ROI image is selected to show our results.

Computer simulations demonstrated, that our method can locate MCCs edges in mammograms satisfactorily, moreover a visual comparison of the edge detection procedure output compared to traditional edge detection techniques indicates the proposed method produces results that are both comparable and competitive. The Fig.2 shows, (a) the original ROI image, (b) the image ROI after having applied the equation (1) with parameters control of threshold $\beta = 0.85$ and rate of enhancement $k = 15$. This parameters that must be predetermined manually and produces good results in image processing step, (c) the binary ROI image based in SOM, (d) show edge detection of MCs on the image ROI by our method.



(a) (b) (c) (d)

Fig. 2. (a) Original ROI image.(b) Enhanced ROI image. (c) Binary ROI image. (d) Edge detection of MCs obtained by our method.



(a) (b) (c) (d)

Fig. 3. (a) Ideal edge of MCs on the original ROI image. (b) Edge detection of MCs obtained by our method. (c) Edge detection of MCs obtained by Canny edge detector. (d) Edge detection of MCs obtained by LoG edge detector.

Table 1. Measured values of Pratt's figure of merit of the proposed method and some conventional edge detectors

Edge detector:	Our method	Canny	LoG
	0.7607	0.6198	0.3203

Finally, a quantitative comparison of our method was performed whit respect to two traditional edge detectors. The reference image for each ROI image was obtained manually. The Fig.3 shows (a) ideal edge of MCs on the original ROI image made by hand, (b) the output image after applying our edge detector method, (c) the output image after applying Canny edge detector, (d) the output image after applied LoG edge detector. Table 1 shows the obtained results using Pratt's figure of merit for testing the ROI image. The values indicate that our method more accurately detects the edges.

4 Conclusions

Edge detection is a fundamental and essential pre-processing step in applications such as image segmentation and computer vision, because edges represent important contour features in the corresponding image. CLF are efficient in image-processing tasks for example edge detection. The main feature of CLF is the direct execution of CLO between the pixel values without keeping any carry bit. This feature makes the CLF simple and fast. For these reasons, in this paper, a method for edge detection of MCCs in ROI images of digital mammograms based in an application of SOM and CLF has been proposed. The method employs image enhancement by histogram adaptive technique, SOM and CLF. The experimental results indicate that the proposed method can locate MCCs edges in mammograms satisfactorily. To evaluate the performance of our method, Pratt's figure of merit was used. According to the values of Pratt's figure of merit, the proposed method performs better than those obtained by the other edge detectors presented in this paper.

References

1. Wu, Y., Huang, Q., Peng, Y.H., Situ, W.: Detection of microcalcifications in digital mammograms based on dual-threshold. In: Astley, S.M., Brady, M., Rose, C., Zwiggelaar, R. (eds.) IWDM 2006. LNCS, vol. 4046, pp. 347–354. Springer, Heidelberg (2006)
2. Rezai-rad, G., Jamarani, S.: Detecting microcalcification clusters in digital mammograms using combination of wavelet and neural network. In: CGIV 2005: Proceedings of the International Conference on Computer Graphics, Imaging and Visualization, pp. 197–201 (2005)
3. Sung-Nien, Y., Kuan-Yuei, L., Yu-Kun, H.: Detection of microcalcifications in digital mammograms using wavelet filter and markov random field model. Computerized Medical Imaging and Graphics 30(3), 163–173 (2006)

4. Song, L., Wang, Q., Gao, J.: Microcalcification detection using combination of wavelet transform and morphology. In: 2006 8th International Conference on Signal Processing, vol. 4 (2006)
5. Vega-Corona, A., Álvarez, A., Andina, D.: Feature vectors generation for detection of microcalcifications in digitized mammography using neural networks. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2687, p. 583. Springer, Heidelberg (2003)
6. Bhattacharya, M., Das, A.: Fuzzy logic based segmentation of microcalcification in breast using digital mammograms considering multiresolution. In: Machine Vision and Image Processing Conference, 2007, pp. 98–105 (2007)
7. Veni, G., Regentova, E.E., Zhang, L.: Detection of clustered microcalcifications with susan edge detector, adaptive contrast thresholding and spatial filters. In: Campilho, A., Kamel, M.S. (eds.) ICIAR 2008. LNCS, vol. 5112, pp. 837–843. Springer, Heidelberg (2008)
8. Laine, A.F., Schuler, Fan, S.J., Huda, W.: Mammographic feature enhancement by multiscale analysis. IEEE Transactions on Medical Imaging 13(4), 725–740 (1994)
9. Vega-Corona, A., Sánchez-García, M., González-Romo, M., Quintanilla-Domínguez, J., Barrón-Adame, J.M.: Contextual and non-contextual features extraction and selection method for microcalcifications detection. In: Proceedings of the World Automation Congress, July 2006, vol. 5 (2006)
10. Mertzios, B.G., Tsirikolias, K.: Applications of coordinate logic filters in image analysis and pattern recognition. In: Proceedings of the 2nd International Symposium on em Image and Signal Processing and Analysis, 2001. ISPA 2001, pp. 125–130 (2001)
11. Boutalis, Y.S., Tsirikolias, K., Metrtzios, B.G., Andreadis, I.T.: Implementation of morphological filters using coordinate logic operations. Pattern Recognition 35(1), 187–198 (2002)
12. Danahy, E.E., Panetta, K.A., Agaian, S.S.: Coordinate logic transforms and their use in the detection of edges within binary and grayscale images. In: IEEE International Conference on Image Processing, ICIP 2007, vol. 3, pp. III–53–III56 (2007)
13. Pratt, W.: Digital Image Processing, 3rd edn. John Wiley & Sons, Chichester (2001)
14. University of South Florida. Digital database for screening mammography (2001), `ftp://figment.csee.usf.edu/pub/ddsm/cases/`

A New Methodology for Feature Selection Based on Machine Learning Methods Applied to Glaucoma

Diego García-Morate[1], Arancha Simón-Hurtado[2], Carlos Vivaracho-Pascual[2], and Alfonso Antón-López[3]

[1] MetaEmotion, Parque Científico
diego.garcia@metaemotion.com
[2] Department of Computer Science, E.T.S.I. Informática
{arancha,cevp}@infor.uva.es
[3] IOBA (Institute of Ophthalmology and Visual Sciences) University of Valladolid,
Campus Miguel Delibes s/n, 47011 Valladolid, Spain
aanton@ioba.med.uva.es

Abstract. In this paper we present a new methodology based on machine learning methods that allows to select from the available features that define a problem, a subset with the most discriminant ones to outperform a classification. As an application, we have used it to select, from the attributes of the optic nerve obtained by Heidelberg Retina Tomograph II, the most informative ones to discriminate between glaucoma and non-glaucoma. Applying this methodology we have identified 7 attributes from the original 103 attributes, improving the ROC area a 2.38%. These attributes match to a large extent with the most informative ones according to the ophthalmologist's experience in clinic as well as the literature.

1 Introduction

Glaucoma is an eye disease that can, eventually, cause irreversible damage to the optic nerve even producing blindness. Early and accurate diagnosis of this disease is fundamental for stopping or delaying its progress, for which there is no known cure. Optic disc tomography, one of the most important tests to detect glaucoma, gives numerical values for many attributes of the optic nerve. This large quantity of numerical data is difficult to be analyzed by an ophthalmologist. Therefore, this is a field where Machine Learning may be very helpful.

Previous researches have tried to rank the attributes of the optic nerve obtained by Heidelberg Retina Tomograph II (HRT II; Heidelberg Engineering, Heidelberg, Germany) [2,7,10]. Their authors show the difficulty of finding a consistent solution because of the many differences between rankings, the difficult interpretation of the Machine Learning methods [5] and the high correlation between the performance of a method and the data set used.

In this paper we present a methodology to search a set of experimental gold standard attributes which make a contribution to the study of this disease and

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1037–1044, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the development of more accurate diagnosis methods. This methodology applies several Machine Learning methods and combines their results, trying to select the best attributes from those yielded by optic disc tomography measurements performed by HRT II. The criterion used to identify these attributes was that they experimentally optimize the application of Support Vector Machines (SVM) trained with the SMO (Sequential Minimal Optimization) algorithm [15] in the diagnosis of glaucoma.

This paper is outlined as follows: In section 2 we describe the data set used in the experiments. Section 3 introduces the Attribute Selection Methods, section 4 presents the configuration with which we have worked. Section 5 shows an application to the glaucoma case. In the last section, section 6, we summarize the results and outline the discussion.

2 Data Set Study

The population used in this study was formed by a set of 166 patients, 68 of them have been diagnosed as glaucomatous patients by an ophthalmologist. Our final data set consisted of 253 eyes (136 glaucomatous, 117 non glaucomatous). The average age of the patients was 58.15 years (Standard Deviation (SD) 16.42), being the average age of non glaucomatous patients 51.71 (SD 15.75) and that of glaucomatous ones 67.43 (SD 12.27). Thus, the non glaucomatous patients are significantly younger than the patients with glaucoma (t-test; $P < 0.05$).

The data set was obtained by the HRT II from patients of IOBA (Institute of Ophthalmology and Visual Sciences - University of Valladolid, Spain); each instance is composed of the values of 103 attributes of the optic nerve, in addition to a classification performed by an ophthalmologist.

3 Attribute Selection Methods

One possible approach to the search of the set of experimental gold standard attributes among all the $(2^{103} - 1)$ possible combinations is given by the machine learning methods. Unfortunately, exhaustive search among all the possible combinations is not computable, being only possible a partial cover of the state space. One possible approach to solve this disadvantage is to combine the results obtained by different machine learning methods, because they will cover differently the state space.

The attribute selection methods are made up of two parts: an Attribute Evaluator and a Search Method. The first one tests a subset and rates it to select the best subsets. The search method performs a search in the set of possible configurations of attributes $(2^{\text{number of attributes}} - 1)$. This architecture is shown in Figure 1.

The attribute evaluation method ranks each attribute with a 1 (relevant attribute) or 0 (irrelevant) for every evaluation test. In all the experiments we used a 10 fold stratified cross-validation as evaluation test, so this is the reason why all the rankings are between 0 and 10.

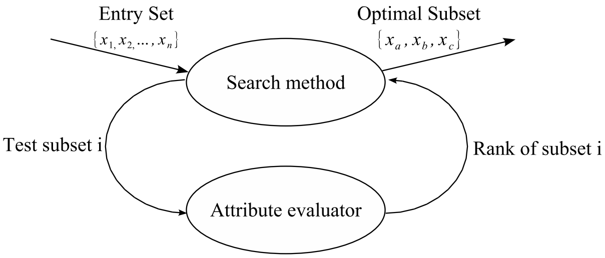


Fig. 1. Scheme of the implementation of the Attribute Selection Methods

In the current section we will describe the methods applied, and in section 5 we will explain how to combine them.

3.1 Attribute Evaluators

Cfssubseteval. The Correlation-base Feature Subset evaluation (cfssubseteval) [6] ranks every subset by considering the individual predictive ability of each attribute along with the redundancy between them.

Subsets of attributes that are highly correlated with the class while having low inter-correlation are preferred.

Wrapper. This attribute evaluator method uses a learning scheme to rate the subset provided by the Search Algorithm. In these experiments we use Support Vector Machines with a Gaussian kernel (SVM-G) as learning scheme [4].

3.2 Search Methods

The search methods [17] used in our experiments are the following:

- **Best First.** It is a search algorithm that searches in the space of attribute subsets by a greedy hillclimbing enhanced with a backtracking facility. It assigns a heuristic function for each node. The evaluation function is $f(n) = g(n) + h(n)$ where $g(n)$ is the cost of the path from the node n to the root and $h(n)$ is an approximation of the cost from the node n to the solution. Best First can perform forward, backward and bidirectional search. that uses the two methods at once.
- **Greedy Stepwise.** Greedy Stepwise performs a greedy forward (or backward) search that begins with a subset and stops when the addition (or deletion) of any remaining attribute results in a decrease in evaluation.
- **Genetic Search.** It is based on genetic algorithms. Its goal is to imitate the natural process of evolution. New data sets are obtained by cross-over or mutation of old ones.
- **Rank Search.** It tries to determine the best subset of attributes by a forward search.

4 Experimental Settings

All these experiments have been done with Weka 3-4-7 [17], an Open Source data mining software from the University of Waikato. A 10 fold stratified cross-validation has been used as performance measure. It is not interesting to use a greater number of folds, because the data set is not big enough to do more partitions. The parameters for the methods were the default ones that Weka assigns to them, except in the case of Support Vector Machines in which we used a gaussian kernel function, with a gamma value of 1.05 and a complexity parameter of 4.0. These parameters were determined in the base case (with all the attributes of HRT) calculating the area under the ROC curve (AUC) for an extensive range of them and obtaining the optimum.

5 Application and Decision Criteria

As mentioned above, the purpose of this paper is to obtain a methodology that helps to find the best attributes that define a characteristic in a problem. As an example we used this methodology to try to select the subset (or subsets) that reasonably maximize the performance of the SVM applied to the diagnosis of Glaucoma. In order to achieve this, we need a ranking which allows us to select the attributes which are to be included in that subset.

By applying the six search methods described above (Bestfirst Backward, Bestfirst Forward, Bestfirst Bidirectional, Greedy Stepwise, Genetic Search, and Rank Search) to our data set with the two attribute evaluators showed, we obtained twelve different rankings for our 103 attributes. These twelve methods are divided in two groups, depending on the attribute evaluator used: cfsubseteval or wrapper. We shall combine the results obtained by the six search methods that form part of each group. For each of these groups we have $(2^6 - 1)$ possible combinations of rankings (adding up the ranks of each attribute, and sorting them again, we obtain an unique ranking for each combination).

The method used to build the subsets is a forward search based on the rankings. We begin with the best attribute, and add an extra one in each step. In this way we obtain 103 subsets per ranking. Since we have $2 * (2^6 - 1)$ rankings we end up with $(2^7 - 2) * 103 = 12978$ subsets.

We select the best subset following two criteria:

1. **Kappa statistic:** It reflects the correlation between two judgements; in this case we compare the real classification (the diagnosis given by an ophthalmologist) with the estimated one.
2. **Combined test:** In this test, we evaluate different statistic measures and rate them to assign a score to each one. The studied measures were: diagnostic precision, specificity, sensibility, f-measure, kappa statistic, number of support vectors, square root of the mean squared error and square root of the relative error. We performed a polynomial equation to obtain a score useful as a measure of the capability of each subset to discriminate between glaucoma and non-glaucoma.

6 Results and Discussion

The subset selected by the kappa statistic is formed by 22 attributes with a kappa value of 0.707 obtained with the wrapper method. The combined test selected a subset of 7 attributes which also corresponded to a combination of rankings obtained by wrapper. Both subsets are showed in Table 1.

Once we evaluated the perfomance of our best results with attribute selection we compared them with linear discriminant analysis (LDA) formulas developed by Mikelberg *et al.* [13], Bathija *et al.* [1] and Mardin *et al.* [12]. This set of formulas ranks patients in two groups (glaucomatous and non glaucomatous) using some attributes obtained by HRT II.

- LDF Mikelberg *et al.* [13]: (rim volume × 1.95) + (height variation contour × 30.12) − (corrected cup shape × 28.52) − 10.08, where *corrected cup shape* is cup shape + [0.0019 × (50 − age)]
- LDF Bathija *et al.* [1]: −3.72 − (5.57 × height variation contour) + (11.78 × RNFL thickness) − (4.37 × cup shape) + (1.85 × rim area)

Table 1. The best subsets obtained

Kappa Statistic		**Combined Test**	
Area	Attribute	Area	Attribute
	Age		Age
T	Mean RNFL thickness (mm)	T	Mean RNFL thickness (mm)
Ti	Rim area (mm^2)	Ti	Rim area (mm^2)
G	Cup shape measure	G	Cup shape measure
G	Rim volume (mm^3)	T	RNFL cross sectional area (mm^2)
Ti	Mean RNFL thickness (mm)	G	Rim volume (mm^3)
T	RNFL cross sectional area (mm^2)	G	Average variability (SD) (mm)
Ts	Rim volume (mm^3)		
G	Average variability (SD) (mm)		
G	FSM discriminant function value		
Ti	Rim volume (mm^3)		
N	Rim area (mm^2)		
Ns	Rim area (mm^2)		
Ni	Cup area (mm^2)		
Ni	Cup disc area ratio		
Ni	Rim volume (mm^3)		
Ni	Mean cup depth (mm)		
G	Rim area (mm^2)		
G	Vertical cupdisk ratio		
G	Maximum contour depression (mm)		
T	Rim volume (mm^3)		
Ts	Cup shape measure		

G: Global, T: Temporal, TS: Temporal Superior, TI: Temporal Inferior, N: Nasal, NS: Nasal Superior, NI: Nasal Inferior.

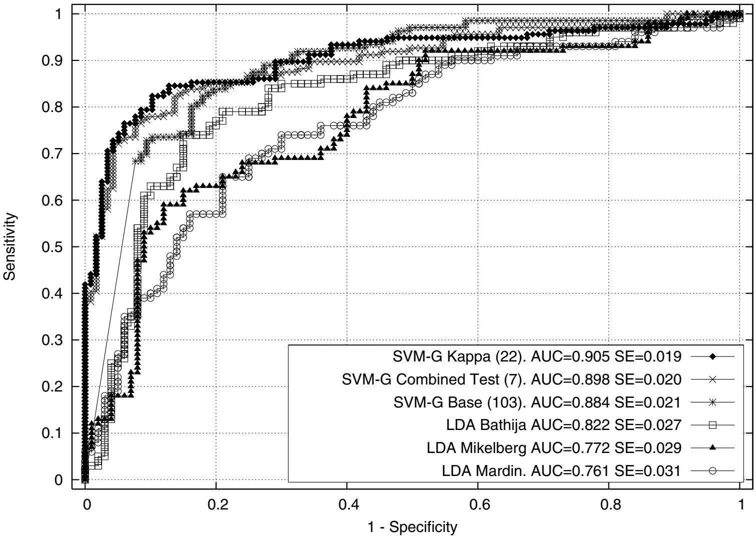


Fig. 2. Full Roc Report

- LDF Mardin *et al.* [12]: $-2.77+(0.3\times \text{rim area})+(3.70\times \text{rim volume})+(4.30\times \text{RNFL thickness})-(3.70\times \text{cup shape})-(3.10\times \text{cup volume})-(0.90\times \text{cup area})$

The AUC obtained with the Bathija formula is 0.822 with a standard error of 0.027, in case of Mikelberg formula we obtained an AUC of 0.722 with a standard error of 0.029 and at last the Mardin formula with AUC of 0.761 and SE of 0.031. So there are significative differences between the best SVM results and the best LDA results (Figure 2).

Other papers, like [2], had tried to rank the attributes that define Glaucoma obtained with the HRT using a sample selected with inclusion criteria similar to this paper. In this case they obtained two sets of 32 and 31 attributes, chosen with forward and backward selection respectively, using SVM with a Gaussian Kernel as decision criteria. Over the base case (with all the attributes of HRT) they obtained an enhancement in the AUC of 3.28% and 2.12% in each case. In our case we obtained a set composed by only 7 attributes (Age, Mean RNFL thickness, Rim area, Cup shape measure, RNFL cross sectional area, Rim Volume, Average Variability) with an enhancement in the AUC of 2,38%. Most of them were chosen in other papers as good to discriminate between glaucoma and non-glaucoma. Kirstein [11] indicated that Rim area, Rim volume, Cup shape measure, Height variation contour and Mean retinal thickness seem to be the most useful in making a baseline evaluation and for tracking glaucoma progression. Four of the 7 attributes selected by the Combined Test match with those. Uchida *et al.* [16] showed that 5 of the 7 attributes selected in our research are between the most informative ones. Other works confirmed the importance of

Table 2. Summary of some results obtained with SVM and LDF

Technique	Number of Attributes	ROC area	SE	Sensitivity (%) Specificity of 75%	 Specificity of 90%
SVM-G Base	103	0.884	0.021	86.0	72.8
SVM-G Kappa	22	0.905	0.019	85.3	79.4
SVM-G Combined Test	7	0.898	0.020	86.8	78.0
LDF Bathija	4	0.822	0.027	78.7	61.0
LDF Mikelberg	3	0.772	0.029	68.4	52.9
LDF Mardin	6	0.761	0.031	69.1	39.0

attributes as Cup shape measure [9], Rim area [8], Rim volume [14] and Mean RNFL Thickness [3].

The aim of this paper is to show a methodology that can help to find simple sets of parameters that define a problem leading the search by the informative capabilities of each set. The reduction of the number of attributes helps to a further computational analysis, and also experience shows that frequently the simple solutions are the best.

In our problem, we have achieved to reduce the attributes to 7 with a very similar enhancement over the base case than in other works [2].

The results showed in the Table 2 were obtained without modifying the parameters of the SVM from the initial ones (complexity parameter 4, gamma value 1.05). We think that they might improve by adjusting those parameters experimentally.

7 Conclusions

In this paper a methodology for feature selection based on machine learning methods has been successfully applied to glaucoma diagnosis to select the most discriminant attributes of the optic nerve obtained by HRT II. We have improved the ROC area obtained with the 103 measurements from HRT using only 7 attributes while we improve the AUC. These match for the most part with the most informative ones according to the ophthalmologist's experience in clinic as well as the literature.

Acknowledgments. This work has been partially supported by Junta de Castilla y León (Spain) under project VA100A08.

References

1. Bathija, R., Zangwill, L., Berry, C.C., Sample, P.A., Weinreb, R.N.: Detection of early glaucomatous structural damage with confocal scanning laser tomography. J. Glaucoma 7, 121–127 (1998)

2. Bowd, C., Chan, K., Zangwill, L.M., Goldbaum, M.H., Lee, T.-W., Sejnowski, T.J., Weinreb, R.: Comparing neural networks and linear discriminant functions for glaucoma detection using confocal scanning laser ophthalmoscopy of the optic disc. Invest. Ophthalmol. Vis. Sci. 43(11), 3444–3454 (2002)
3. Bowd, C., Weinreb, R.N., Williams, J.M., Zangwill, L.M.: The retinal nerve fiber layer thickness in ocular hypertensive, normal, and glaucomatous eyes with optical coherence tomography. Arch. Ophthalmol. 118, 22–26 (2000)
4. Cortes, C., Vapnick, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
5. Fu, X., Ong, C., Keerthi, S., Guang Hung, G., Goh, L.: Extracting the knowledge embedded in support vector machines. In: IJCNN 2004. International Joint Conference on Neural Networks, pp. 291–296 (2004)
6. Hall, M.: Correlation-based feature selection for machine learning. Ph.D diss, Waikato University, Department of Computer Science, Hamilton, NZ (1998), `citeseer.ist.psu.edu/hall99correlationbased.html`
7. Hatch, W.V., Flanagan, J.G., Etchells, E.E., Williams-Lyn, D.E., Trope, G.E.: Laser scanning tomography of the optic nerve head in ocular hypertension and glaucoma. British Journal of Ophthalmology 81, 871–876 (1997)
8. Iester, M., Mikelberg, F.S., Courtright, P., Drance, S.M.: Correlation between the visual field indices and heidelberg retina tomograph parameters. J. Glaucoma 6(2), 78–82 (1997)
9. Iester, M., Mikelberg, F.S., Swindale, N.V., Drance, S.M.: Roc analysis of heidelberg retina tomograph optic disc shape measures in glaucoma. Can. J. Ophthalmol. 32(6), 382–388 (1997)
10. Jonas, J.B., Bergua, A., Schmitz-Valckenberg, P., Papastathopoulos, K.I., Budde, W.M.: Ranking of optic disc variables for detection of glaucomatous optic nerve damage. Invest. Ophthalmol. Vis. Sci. 41(7), 1764–1772 (2000)
11. Kirstein, E.M.: Use of the heidelberg retinal tomograph ii for optic nerve head documentation and analysis. College of Optometry Continuing Education, `http://www.opt.pacificu.edu/ce/catalog/9451-GL/HRT-Kirstein.html`
12. Mardin, C.Y., Horn, F.K., Jonas, J.B., Budde, W.M.: Preperimetric glaucoma diagnosis by confocal scanning laser tomography of the optic disc. Br. J. Ophthalmol. 83, 299–304 (1999)
13. Mikelberg, F.S., Parfitt, C.M., Swindale, N.V., Graham, S.L., Drance, S.M., Gosine, R.: Ability of the heidelberg retina tomograph to detect early glaucomatous visual field loss. J. Glaucoma 4, 242–247 (1995)
14. Nakamura, H., Maeda, T., Suzuki, Y., Inoue, Y.: Scanning laser tomography to evaluate optic discs of normal eyes. Jpn. J. Ophthalmol. 43(5), 410–414 (1999)
15. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Advances in kernel methods: support vector learning, pp. 185–208. MIT Press, Cambridge (1999)
16. Uchida, H., Brigatti, L., Caprioli, J.: Detection of structural damage from glaucoma with confocal laser image analysis. Invest. Ophthalmol. Vis. Sci. 37, 2393–2401 (1996)
17. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)

Tissue Recognition Approach to Pressure Ulcer Area Estimation with Neural Networks

Francisco J. Veredas[1], Héctor Mesa[1], and Laura Morente[2]

[1] Dpto. Lenguajes y Ciencias de la Computación, Universidad de Málaga
Málaga, Spain
{fvn,hector}@lcc.uma.es
[2] Escuela Universitaria de Enfermería, Diputación Provincial de Málaga
Málaga, Spain
lmorente@malaga.es

Abstract. Pressure ulcer is a clinical pathology of localized damage to the skin and underlying tissue with high prevalence rates in aged people. Diagnosis and treatment of pressure ulcers involve high costs for sanitary systems. Accurate wound-state evaluation is a critical task for optimizing the effectiveness of treatments. Reliable trace of wound-state evolution can be done by precisely registering the wound area. Clinicians estimate the wound area with often subjective and imprecise manual methods. This article presents a computer-vision approach based on machine hybrid-learning techniques to precise automatic estimation of wound dimensions on pressure ulcer real images taken under non-controlled illumination conditions. The system combines neural networks and Bayesian classifiers to effectively recognize and separate skin and healing regions from wound-tissue regions to be measured. This tissue-recognition approach to wound area estimation gives high performance rates and operates better than a widespread clinical method when approximating real wound areas of variable size.

1 Introduction

The European Pressure Ulcer Advisory Panel (EPUAP) defines a pressure ulcer as an area of localized damage to the skin and underlying tissue caused by pressure, shear, friction or a combination of these factors. Prevalence studies on this pathology show a high impact of age on pressure ulcer occurrence [9, see table 7]. Prevention, care and treatment of pressure ulcers involve high costs for sanitary systems and have important consequences on the health of the aged population. As suggested by the EPUAP's *Pressure Ulcer Treatment Guidelines* precise wound management requires a first classification of the pressure ulcer — in one of the four grades EPUAP proposes— together with an accurate wound assessment. This way, measurement of the wound area becomes a critical operation to evaluate and trace the evolution of the pressure ulcer state. There are several documented methods on wound area estimation, which include the measurement of the wound perimeter (i.e. circumference), the estimation of the

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1045–1052, 2009.
© Springer-Verlag Berlin Heidelberg 2009

perpendicular maximal diameters of the wound, the manual tracing of the wound perimeter by disposing an acetate paper on the ulcer-bed or even on a photograph of the ulcer, the use of computer assisted planimetry or stereophotogrammetry, or the use of classical image processing techniques. In clinical exercise, one of the simplest, fastest and more widespread methods for wound area estimation is approximating the wound area with a rectangle which is manually fitted to the wound-plane by measuring the two perpendicular main axes of the wound with a simple standard ruler [8]. However, this technique suffers from an important limitation of being very imprecise. In this article, we analyze the performance results from the application of this rectangular estimation method, while comparing them with the outcomes from an indirect estimation method which is a computational intelligence approach to tissue recognition.

Computer vision and computational intelligence techniques could become effective approaches to address more precise region segmentation and tissue identification on pressure ulcer images. Current studies on the same or similar application areas concentrate on two partial objectives of the problem: 1) the identification of the wound area, by contour detection with histogram segmentation, active contours modeling, region growing, clustering approaches or skin texture models [2–4]; and 2) the detection of the different tissues existing in the wound, by using diverse segmentation techniques —such as histogram thresholding, watersheds, meanshift smoothing, region growing, classification or graphs— sometimes combined with machine learning strategies [5, 11]. The main handicap for all these methods is the particular nature of pressure ulcer images —which present very heterogeneous colorations and irregular and vague boundaries.

This article presents a machine hybrid-learning approach based on neural networks (NN) and Bayesian classifiers (BC) to design an automatic procedure for effective identification of significant tissues on pressure ulcer color-digital images. With this methodology the accurate recognition of the most significant tissues surrounding the wound-bed, i.e. healthy skin and healing tissues, is achieved and the separation and measuring of the wound area is then possible. By this indirect method, the number of pixels in the wound-bed can be estimated and converted later into standard measurement units (i.e. squared millimeters) by using an standardized marker of $1\,cm^2$-size which is included in each image frame.

2 Tissue Recognition Methodology

The clinicians associated to this project took color photographs of pressure ulcers from patients with home-care assistance. Sacrum and hip pressure ulcers were photographed under non-controlled illumination conditions by using a Canon EOS ®40D digital camera. The images were acquired with a ring flashlight Sigma ®EM-140 to get well-illuminated scenes, and at a distance of approximately $30 - 40\,cm$ from the wound plane. A macrophotography lens Canon ®EF-S $60\,mm$ f/2,8 USM was used to ensure well-focused pictures within these short distances. A group of clinical experts selected a total of 113 photographs which were considered to be an appropriate data set for analysis because of the presence of all the tissues significant for pressure ulcer evaluation. These 113 images have been used in the designing of a machine-learning approach to tissue recognition with image processing and computational intelligence techniques. This resultant system is used afterwards to estimate the wound area on a set of 50 randomly selected images from the data base of images taken by the clinicians. The results from this automatic wound area estimation process are compared with the real wound areas as well as the areas estimated by the rectangular fitting method above.

One of the most critical concerns for effective tissue identification on wound images is the accurate segmentation of the specific regions present in the image. Pressure ulcers mostly present irregular shapes, vague boundaries and very heterogeneous colorations. These conditions make precise automatic segmentation to be a non-trivial computational task. In this paper an adaptive mean shift procedure [1] for border-preserving region smoothing is used as a preliminary stage for region growing segmentation. The mean shift procedure has shown a high reliability in different image segmentation tasks [1].

Once the pressure ulcer images are segmented, a set of color and texture features is extracted from each resultant region. A principal component analysis (PCA) allows the dimensionality reduction of the initial color and texture feature space. Finally, this pattern set is used to train supervised NNs —multilayer perceptrons (MLP). A two-stage cascade architecture is arranged: at a first stage, k-fold cross-validation is used to generate a set of k NNs [6] trained to distinguish between skin regions and *other tissues* (non skin). Therefore, the outputs from the ensemble of k NNs for each input pattern is used to train a BC to form a Bayesian Committee Machine (BCM) [10] which effectively combines the predictions of the k NNs to improve the effectiveness of the classification [7]. At the second stage of the system, this scheme above is repeated: a new architecture of k NNs which are driven by a k-fold cross-validation training procedure is designed; again, this set of NNs feeds a BC to classify the remaining regions in two tissue classes: healing and wound-bed tissue. The clinical experts supplied the networks and the classifiers with the appropriate output labels (i.e. tissue classes) during the training processes.

In next sections, each particular stage of this general methodology above will be explained in detail.

2.1 Tissue Classification with Machine Learning

Once the 113 images have been segmented by the mean-shift segmentation procedure [1], a set of color and texture features from each of the $15,879$ resulting regions is extracted. After feature extraction, PCA is applied on the pattern set to reduce the feature space dimensionality, giving 19 resulting features which explain the 99% of the total variance. These 19 features constitute the inputs to the tissue classification hybrid-learning system.

A collaborative group of expert clinicians labeled each one of the regions from the set of 113 images by assigning one of the three possible tissue classes (skin, healing or wound) to each particular region in the automatically segmented images: these labels constitute the desired outputs from the NNs and the BCs during the training phases. A non-uniform distribution of the number of patterns in the different tissue classes resulted from this expert labeling process, which could be a severe impediment for the effective training of a single MLP which follows a classical cross-validation scheme (i.e. splitting the input data in two subsets, one for training and one for validation). As suggested in [7], to effectively surpass these limitations, the tissue classifiers are designed by following a k-fold cross-validation strategy to train a hybrid-learning system which consists of a set of k MLPs which subsequently feeds a BC to form a BCM.

The Hybrid Architectures. The system for tissue recognition has been designed as a two-stage cascade procedure. The first stage consists of a hybrid-learning architecture to separate skin tissues from the others. Moreover, the second stage is a new hybrid-learning architecture that is fed with the patterns labeled as “non skin” by the preceding classification stage, and therefore this second stage classifies these patterns as healing or wound-tissue regions.

In [6], Liu presents an approach to create a stable NN from a set of NNs trained by following a cross-validation strategy. Based on this Liu’s proposal, we have first split the input data set of $15,879$ patterns in 10 independent subsets. Those regions from the same segmented image are always located in the same subset: a mean of $1,587.9$ patterns (standard deviation $= 119.6$) is included in each subset. A k-fold cross-validation procedure is used to generate a set of NNs, in which the parameter k was set to be 9. To have a precise estimation of the generalization error, at each cross-validation stage i, one of the 10 pattern subsets was taken out as the testing set. Each of rest 9 subsets is taken in turn as the validation set, while the other 8 subsets are used to train a NN. Early stop based on the mean squared error (m.s.e.) of the validation subset is used to preclude the overfitting on the training set. Therefore, 9 NNs are generated at each cross-validation step i. Next, a Naive Bayes classifier [12] is trained with the outputs from the 9 NNs to have a combination of the 9 predictions (i.e. tissue classifications) from the 9 NNs, for each one of the input feature patterns, so that a BCM [10] is formed. The m.s.e. of the testing set is computed on this BCM. At the stage $i + 1$, the following pattern subset is used as the testing set, and the procedure above repeats. Finally, an average of m.s.e from the 10 testing sets is computed to get an estimator of the BCM generalization error. The m.s.e. from the BCM on the 10 testing sets was of 0.1552 for skin regions and 0.1388 for healing tissues. Although different MLP architectures for the NNs have been tested, the best error and performance are obtained with three-layer MLPs (with one hidden layer).

This approach based on Bayesian committee machines has shown statistical and clinically significant differences when classifying tissue patterns from a non-uniform distribution of classes on wound images [7].

Table 1. Averaged performance rates (top) and confusion matrix (bottom) (%) from patterns in the 10 testing sets for a k-fold cross-validation strategy

	`Sensitivity`	`Specificity`	`Success`	`Accuracy`
`Skin`	89.31	86.28	87.80	87.90
`Healing`	61.63	91.56	76.59	86.09
`Wound-bed`	86.59	95.48	91.03	92.98

	`Skin`	`Healing`	`Wound-bed`
`Skin`	89.31	8.85	1.84
`Healing`	25.99	61.63	12.38
`Wound-bed`	5.75	7.66	86.59

Results from Tissue Recognition. The table 1,*top* shows the averaged performance rates from patterns in the 10 testing sets from the k-fold cross-validation training. For each tissue type, the *sensitivity* (i.e. correct-classification probability) is the percentage of regions simultaneously marked as "that-tissue" by the BCMs and also by the expert clinicians; the *specificity* means the percentage of regions simultaneously marked as "non-that-tissue" by the BCMs and the clinicians; *success* rates are the average of sensitivity and specificity scores; finally, the *accuracy* is the proportion of regions being correctly classified. These results show high success and accuracy rates, which address to high effectiveness and fitness of the classifiers.

Given the confusion matrix of table 1,*bottom*, if skin and healing regions are included in a same "peripheral tissue" class, the following performance rates could be then considered as results from the classification (see table 1,*top*): sensitivity $= 98,59\%$, specificity $= 93,77\%$, success $= 96,17\%$ accuracy $= 97,27\%$. These global results from "peripheral tissue" recognition performance rates can be considered as very high efficacy scores in separating the wound-bed regions from the rest of the regions in the image (i.e. segmenting the wound area). These outcomes point to also a high efficacy in determining the wound area with the automatic tissue recognition method based on machine hybrid-learning techniques proposed here.

The figure 1,*A, B, D* and *E* shows two examples from the application of the automatic tissue recognition system on two different real pressure ulcer images. As is shown in this figure, the automatic recognition of skin and healing tissue regions in the images allows the precise separation of the wound-bed tissues from the rest of the image, this way permitting the automatic estimation of the wound area.

3 Wound Area Measurement by Tissue Recognition

The hybrid-learning tissue-recognition system based on NNs and BCs presented in the previous sections has been used to estimate the wound area on a set of pressure ulcer images consisting of 50 real images which were not used during the

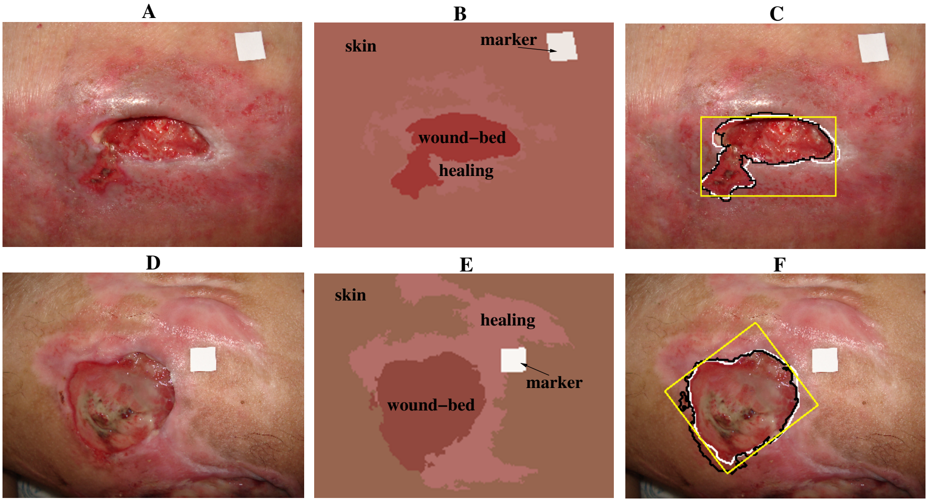


Fig. 1. Examples of tissue recognition with NNs and BCs. In A and D the original pressure ulcer images are shown. In B and E the results from the automatic tissue recognition system on the images in A and C are shown. In each image, those regions with the same class-label are given a same pseudocolor which results from an average of the color of the regions in the same class. The marker for measurement normalization is also shown in the images (discussion about the marker recognition procedure is not included in this paper). In C and F wound area has been estimated with expert manual tracing of the real area (white), our tissue recognition system (black) and rectangular area fitting (yellow).

training of the system. The figure 1 shows two pressure ulcer images from this set of 50 images (see original images in *A* and *D*). Each image in figure 1,*C* and *F* shows the real wound area, the area estimated with our hybrid-learning tissue-recognition system (see figure 1,*B* and *E*), and finally an optimized rectangular fitting which idealizes the widespread clinical manual technique of wound-area estimation (so that wound-bed is fully covered with this ideal rectangular shape but also the minimum peripheral tissue is included in this optimized rectangular area). As can be observed in figure 1,*C* and *F*, even the optimized rectangular fitting overestimates the real area significantly. For the two sample images of the figure, the estimation of the area by tissue-recognition results in not very significant differences with the real areas traced by the experts.

The figure 2 shows a graph of the real wound area estimated by the expert manual tracing of the wound border on each one of the 50 images of the image set of this study. The images were sorted based on their real wound area and then assigned an index from 1 to 50 to be disposed in the graph. The area estimated by the machine-learning tissue-recognition system as well as the estimation of the wound are by rectangular fitting are also shown in the figure. The distances between the real and the estimated areas for the tissue-recognition and the rectangular-fitting approaches are also represented in the figure. As can

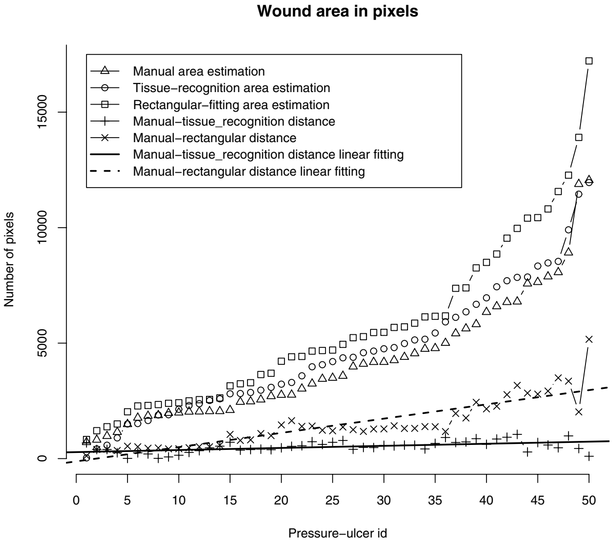


Fig. 2. Wound area measurement graphs: comparative estimation of wound area by tissue-recognition and rectangular fitting versus real wound size

be observed in this figure, the rectangular-fitting wound area estimation always overestimates the real wound size, and the distance between the rectangular-fitting area and the real area increases as the real wound size is incremented (linear slope $\sim 61,71$). The wound area estimation with our machine-learning tissue-recognition approach stays nearby the real wound area as the wound size increases for all the 50 cases analyzed in the figure 2 (linear slope $\sim 9,06$). As can be concluded from these results, the wound area estimation with our tissue-recognition approach improves significantly (for the 50 cases analyzed) the estimation done with the classical rectangular fitting. Moreover, the tissue-recognition approach shows a lower sensitivity to variations in wound size, while the rectangular-fitting overestimation increases as the wound size becomes larger.

4 Conclusions

A hybrid-learning approach to image segmentation and automatic tissue recognition on pressure ulcer digital images taken in non-controlled illumination environments has been presented. This approach is based on the design of a machine learning tissue-classification system consisting of two sequential stages which combine NNs and BCs trained by following a k-fold cross-validation strategy. This system performs automatic recognition of skin and healing tissue regions to segment the wound-bed area in the image. High performance classification rates result from this approach, and the appropriateness of this tissue-recognition procedure has been proved on real pressure ulcer images. Our approach to automatic tissue recognition has been used to estimate the wound area in pressure ulcer images, and it has been compared with the clinical procedure of rectangular-fitting wound area estimation. Our results show the appropriateness of the tissue-recognition approach to wound area estimation, give wound area measures which stay nearby the real wound area delineated by the experts, and show low sensitivity to wound size changes. The methodology and results presented in this paper could have important implications to the field of clinical pressure ulcer diagnosis. Clinical decisions on treatments are now supported on an objective and reliable computational tool for automatically detecting, estimating and registering important tissue measurements.

Acknowledgments

This research has been funded by the Spanish *Ministerio de Sanidad y Consumo* (*Fondo de Investigación Sanitaria*), project PI060131, and *Consejería de Salud, Servicio Andaluz de Salud*, project PI-0197/2007.

References

[1] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space anaylsis. IEEE Trans. Pattern Anal. Mach. Intell. 24, 603–619 (2002)

[2] Cula, O., Dana, K., Murphy, F., Rao, B.: Skin texture modeling. Int. J. Comput. Vision 62(1-2), 97–119 (2005)

[3] Jones, T.D., Plassmann, P.: An active contour model for measuring the area of leg ulcers. IEEE Trans. Med. Imaging 19(12), 1202–1210 (2000)

[4] Karkanis, S.A., Iakovidis, D.K., Maroulis, D.E., Karras, D.A., Tzivras, M.: Computer-aided tumor detection in endoscopic video using color wavelet features. IEEE Trans. Inf. Technol. Biomed. 7(3), 141–152 (2003)

[5] Kosmopoulos, D., Tzevelekou, F.: Automated pressure ulcer lesion diagnosis for telemedicine systems. IEEE Eng. Med. Biol. Mag. 26(5), 18–22 (2007)

[6] Liu, Y.: Create stable neural networks by cross-validation. In: Proc. of the IEEE International Joint Conference on Neural Networks, IJCNN 2006, Vancouver, BC, Canada, pp. 3925–3928 (2006)

[7] Mesa, H., Veredas, F., Morente, L.: Tissue recognition for pressure ulcer evaluation. In: Proceedings of the 4th European Conference of the IFMBE (MBEC 2008), Antwerp, Belgium, pp. 1524–1527. Springer, Heidelberg (2008)

[8] Stacy, M., Burnand, K., Layer, G., Pattison, N.: Measurement of the healing of venous ulcer. ANZ Journal of Surgery 61(11), 844–848 (1991)

[9] Tannen, A., Dassen, T., Bours, G., Halfens, R.: A comparison of pressure ulcer prevalence: concerted data collection in the netherlands and germany. International Journal of Nursing Studies 41(6), 607–612 (2004)

[10] Tresp, V.: A bayesian committee machine. Neural Comput. 12, 2719–2741 (2000)

[11] Wannous, H., Treuillet, S., Lucas, Y.: Supervised tissue classification from color images for a complete wound assessment tool. In: Proc. of the 29th Annual International Conference of the IEEE EMBS, Lyon, France, pp. 6031–6034 (2007)

[12] Zhang, H.: The optimality of Naïve Bayes. In: Proc. 17th Internat. FLAIRS Conf., Florida, USA (2004)

Classification of Schistosomiasis Prevalence Using Fuzzy Case-Based Reasoning

Flávia T. Martins-Bedé[1], Lluís Godo[2], Sandra Sandri[1,2], Luciano V. Dutra[1],
Corina C. Freitas[1], Omar S. Carvalho[3], Ricardo J.P.S. Guimarães[3],
and Ronaldo S. Amaral[4]

[1] Instituto Nacional de Pesquisas Espaciais - INPE
CP 515, 12201-970 - São José dos Campos - SP, Brazil
[2] Institut d'Investigació en Intel·ligència Artificial, IIIA - CSIC
Campus UAB s/n, 08193 Bellaterra, Spain
[3] Centro de Pesquisas René Rachou/FIOCRUZ, MG, Brazil
[4] Secretaria de Vigilância em Saúde/MS, Brazil

Abstract. In this work we propose the use of a similarity-based fuzzy CBR approach to classify the prevalence of Schistosomiasis in the state of Minas Gerais in Brazil.

1 Introduction

Case-based reasoning [1], CBR for short, can be considered as a form of similarity-based or analogical reasoning since the basic principle implicitly followed in this problem solving methodology is that *similar problems have similar solutions*. A weaker version of this principle stated is given by "it is only plausible (but not necessary) that similar problems have similar solutions". In [2] the authors propose a novel Fuzzy CBR algorithm for classification, in which this weaker principle is read, in the classification context [5] as:

> "The more similar are the problem descriptions of two cases,
> the more *possible* their classification values are similar"

Basically, in this context, a CBR base is composed of cases of the form $c = (d, cl)$, where d is the case description modeled as a vector of values for a set of attributes and cl its associated class. The assignment of a class to a new case description d_0 will depend on the similarity between d_0 and the descriptions of the cases in the CBR base. In the approach proposed in [2], this similarity is calculated as a weighted mean of the similarity between each attribute addressed in the description part of the cases in the learning data set. In this approach, the weight vectors are learnt in such a way as to minimize the misclassification of the cases already contained in the base.

Schistosomiasis mansoni is a disease with social and behavioral characteristics. Snails of the *Biomphalaria* species, the disease intermediate host, uses water as a vehicle to infect man, the disease main host. In Brazil, six million people are infected by it, mainly in poor regions of the country [10]. According to the data

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1053–1060, 2009.
© Springer-Verlag Berlin Heidelberg 2009

presented at the Brazilian Information System for Notifiable Diseases (SINAN) of the Ministry of Health, from 1995 to 2005, more than a million positive cases were reported, 27% of them in the State of Minas Gerais.

In [7], the authors present a classification Schistosomiasis prevalence for the State of Minas Gerais, using remote sensing, climate, socioeconomic and neighborhood related variables. Two approaches were used, a global and a regional one. In the first approach, a unique regression model was generated and used to estimate the disease risk for the entire state. In the second approach, the state was divided in four regions, and a model was generated for each one of them. Imprecise classifications were also generated for both approaches, using the estimated standard deviation and several reliability levels as basis.

The aim of this paper is to check the usefulness of the fuzzy CBR approach to classification proposed in [2] in order to estimate and classify schistosomiasis prevalence, as an alternative to the linear regression model approach developed in [7]. To allow the comparison of the results, we present two approaches for solving the problem, a global and a regional one, following the guidelines in [7].

This work is organized as follows. In Section 2 we describe the similarity-based fuzzy CBR model we have used. In Section 3 we present our application context and discuss the experiments we have performed. Section 4 finally brings the conclusion.

2 A Similarity-Based Fuzzy CBR Model for Classification

2.1 Working Framework

Before going into more details, let us specify our working framework for classification-like case-based reasoning problems. Let us assume we have a base of cases CB consisting of an already solved set of cases, where a case is represented by a (complete) tuple of attribute values describing the situation or problem to solve together with a solution class or result. To fix ideas, let $\mathbf{A} = \{a_1, \ldots, a_n\}$ be the set of description attributes and let *class* denote the class attribute. Moreover, let us denote by $D(a_i)$ and $D(class)$ the domains of the attributes a_i and *class* respectively (so $D(class)$ is the set of solution classes). Then a case $c \in CB$ will be represented as a pair $c = (d, cl)$, where $d = (a_1(c), \ldots, a_n(c))$ is a n-tuple with the problem description values and $cl = class(c)$ is the solution class for the case c. If we write $\mathbf{D} = D(a_1) \times \ldots \times D(a_n)$ ($\mathbf{D}$ for descriptions) and $\mathbf{Cl} = D(class)$, then a case base CB is just a subset of $\mathbf{D} \times \mathbf{Cl}$. In the following, all definitions will use this classification-oriented notation.

In this framework, given a case base $CB = \{c_i = (d_i, cl_i)\}_{i \in I}$ and a new problem description d^*, the CBR task is to find (guess) a solution class cl^* for d^*, by applying the above general principle in some form, i.e. taking into account the similarity of d^* with already solved cases $c_i \in CB$.

In its most general sense, a fuzzy similarity relation on a domain Ω is a mapping $S : \Omega \times \Omega \to [0, 1]$ which assigns to every pair (w, w') of elements of Ω a number measuring how much w and w' resemble each other according to some given criteria, in the sense that the higher $S(w, w')$, the larger their resemblance.

In particular, $S(w,w')=1$ means that w and w' are undistinguishable, while $S(w,w')=0$ means that w and w' have nothing in common. One can also understand $\delta(w,w')=1-S(w,w')$ as a kind of distance between w and w'. Usual and reasonable properties (see e.g. [6]) required of such functions are reflexivity and symmetry, i.e. $S(w,w)=1$ and $S(w,w')=S(w,w')$, for any $w,w'\in\Omega$. S is called *separating* if it verifies that $S(w,w')=1$ iff $w=w'$. Sometimes they are also required to fulfill a weak form of transitivity, namely $S(w,w')\otimes S(w',w'')\leq S(w,w'')$, where $\otimes$ is a t-norm. For our purposes, and unless stated otherwise, we shall consider similarity relations as reflexive and symmetric fuzzy binary relations but neither necessarily transitive nor separating.

2.2 Main Elements of the Model Proposed

The approach we will describe in the rest of this section requires that, for each attribute $a\in\mathbf{A}$, there is an available fuzzy similarity relation S_a on $D(a)$, as well as a fuzzy similarity relation S_{class} over the set of classes $\mathbf{Cl}$, as defined in Section 2.1.

The main step in the method is to define a suitable similarity relation S_D between the case descriptions in the case base CB and an arbitrary problem description. Our working assumption is that such similarity will be defined as a weighted average of the existing similarity functions S_a for each attribute. Then, of course, we need additional information to assess the relevance (weight) of each attribute for retrieving a particular case. In particular, we will assume there is a best[1] *set of weighs vector for each case* that properly evaluates the importance of each attribute when computing the similarity between that case and another arbitrary case description in the case base CB.

Definition 1. *Let* $\mathbf{A}=\{a_1,\ldots,a_n\}$ *be the set of attributes considered in* $\mathbf{D}$ *and, for each* $a\in\mathbf{A}$*, let* S_a *be the corresponding similarity relation on* $D(a)$*, and let* $\mathbf{w}:\mathbf{A}\to[0,1]$ *be a weight assignment to attributes, i.e. an assignment such that* $\sum_{a\in\mathbf{A}}\mathbf{w}(a)=1$*. Then, we define the induced fuzzy similarity relation* $S_D^{\mathbf{w}}:\mathbf{D}\times\mathbf{D}\to[0,1]$ *over case descriptions as follows:*

$$S_D^{\mathbf{w}}(d_1,d_2)=\sum_{a\in\mathbf{A}}\mathbf{w}(a)\cdot S_a(a(d_1),a(d_2)) \tag{1}$$

using the notation $d_1=(a_1(d_1),\ldots,a_n(d_1))$ *and* $d_2=(a_1(d_2),\ldots,a_n(d_2))$*.*

Note that, so defined, $S_D^{\mathbf{w}}$ is a indeed similarity relation in the sense of Section 2.1, i.e. it is reflexive and symmetric.

Once we have defined the similarity relation $S_D^{\mathbf{w}}$, we can define how adequate a solution class *cl* is for a problem description d^* just by comparing d^* to the descriptions of all those cases in CB sharing that solution class *cl*, and aggregating all these similarity values.

---

[1] In the sense explained in Section 2.3.

Definition 2. *Let be* $CB \subseteq \mathbf{D} \times \mathbf{Cl}$ *a case base. Given a set of weight assignments* $\mathbf{W} = \{\mathbf{w}_c\}_{c \in CB}$ *(one per each case in* CB*), and a suitable aggregation function* F *on* $[0,1]$*, the adequacy degree between a case description* $d^* \in \mathbf{D}$ *and a solution class* $cl \in \mathbf{Cl}$ *is defined as follows:*

$$\Pi_{\mathbf{W},F}(d^*, cl) = F(\{S_D^{\mathbf{w}_c}(d^*, d) \mid c \in CB, c = (d, cl\}).$$

where $S_D^{\mathbf{w}_c}$ *is defined as in Definition 1.*

Depending on the application, suitable aggregation functions [4] may be for example disjunctive functions, like the maximum or other t-conorms, or some kinds of average functions, like quasi-arithmetic means or even more sophisticated aggregation functions.

Finally, given case base CB, a set of weight assignments $\mathbf{W} = \{\mathbf{w}_c\}_{c \in CB}$ and a suitable aggregation function F on $[0,1]$, the last step in the fuzzy CBR approach consists in assigning to a case description $d^* \in \mathbf{D}$ the solution class cl^* such that

$$cl^* = \arg\max_{cl \in \mathbf{CL}} \Pi_{\mathbf{W},F}(d^*, cl).$$

2.3 Learning the Weight Assignments for Each Case

Next, we describe how to learn the appropriate weight assignment for each case in the base CB. So, in the following, we will assume that a case $c_0 \in CB$ is known and fixed along the learning process. In fact, the same process we describe below for c_0 will be applied for each case in CB. Naturally, for each case c_0, the process would lead to its corresponding weight assignment $\mathbf{w}_0$.

To do so, in addition to the case c_0, we need to fix a subset of cases $LS_0 \subseteq CB$, i.e. a collection of problem descriptions whose solution class is known. This is the learning set. Then, the weights determination can be formulated in the following way:

Problem 1 (Weight Determination Problem). Let $c_0 = (d_0, cl_0)$ be a case in base CB and let $LS_0 \subseteq CB$ be the learning set relative to c_0. Then the weight determination problem relative to c_0 is to determine a weight assignment $\mathbf{w}_0 : \mathbf{A} \rightarrow [0,1]$ such that, for each case $c = (d, cl) \in LS$, the similarity between d_0 and d, $S_D^{\mathbf{w}_0}(d_0, d)$, approximates as much as possible the similarity between the solution classes cl_0 and cl, $S_{Class}(cl_0, cl)$.

Using the square difference to measure the divergence between the two similarities (i.e., the similarity between the two case descriptions and the similarity between their classes), we can reformulate the problem as follows:

Problem 2. Let $c_0 = (d_0, cl_0)$ be a case in the base CB and let $LS_0 \subseteq CB$ be the learning set from which the weight assignment $\mathbf{w}_0 : \mathbf{A} \rightarrow [0,1]$ relative to c_0 will be determined. Then the weight determination problem relative to c_0 is to find the values $\mathbf{w}(a_1), ..., \mathbf{w}(a_n)$ that minimize the following expression:

$$\sum_{(d,cl)\in LS_0} \left[S_D^{\mathbf{w}}(d,d_0) - S_{Class}(cl,cl_0) \right]^2$$

subject to the following constraints over w_{c_0}:

(1) $\sum_{a\in\mathbf{A}} \mathbf{w}(a) = 1$, and
(2) $\mathbf{w}(a) \geq 0$ for all $a \in \mathbf{A}$.

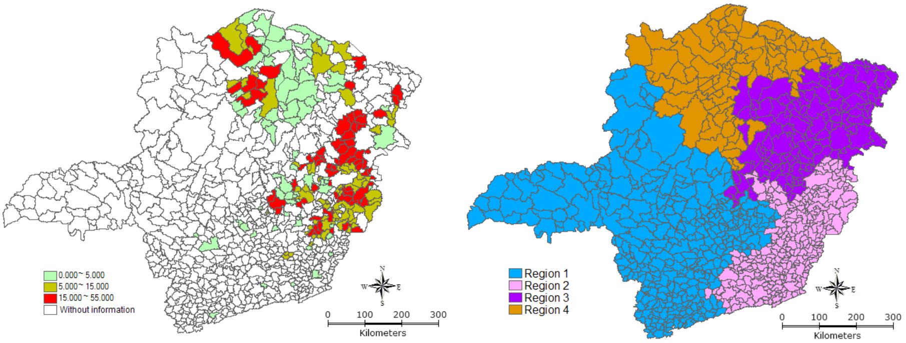
In the experiments described in Section 3 below, we have divided case base CB in two parts, a training set CBt and a validation set $CBv = CB - CBt$, and for each case $c_0 \in CBt$ we obtain a weight assignment $\mathbf{w}_0$ by solving the above minimization problem in a leave-one-out fashion, by taking as learning set the whole CBt after removing case c_0, i.e., for each $c_0 \in CBt$, $LS_0 = CBt - \{c_0\}$.

In order to solve this minimization problem, we apply the algorithm introduced in [8] with the extension described in [9]. It is worth pointing out that this minimization algorithm, as similar ones existing in the literature, fails to give a (unique) solution when there exists a particular kind of linear dependence among the columns in the data matrix of the problem. In our case, this would refer above to the matrix of similarity values $\mathbf{S} = \{s_{i,j}\}$ with $i = 1,\ldots,n$ and $j = 1,\ldots,m$, where $s_{ij} = S_{a_i}(a_i(d_j), a_i(d_0))$, assuming that $\mathbf{A} = \{a_1,\ldots,a_n\}$ and $LS_0 = \{(d_1,cl_1),\ldots,(d_m,cl_m)\}$. In fact, as shown in [9], the problem only arises when there is a column $\mathbf{s}_i = (s_{1i},\ldots,s_{ni})$ that can be written as a linear combination of the others in the form $\mathbf{s}_i = \sum_{j\neq i} p_j \cdot \mathbf{s}_j$ with $p_j \geq 0$ and such that $\sum_j p_j = 1$. In such a case, when removing one of the linearly dependent columns we get the same minimum we would get when considering all the attributes. Therefore, an alternative approach is to consider as many subproblems as dependent attributes, where each subproblem corresponds to the original one after removing one of the dependent attributes. The solution with a minimum error would correspond to the solution of the original problem.

3 Classification of Schistosomiasis Prevalence

3.1 The Original Experiments: Materials and Methods

In the original experiments presented in [7], the disease prevalence data was provided by the Health Secretary of the State of Minas Gerais state. The prevalence is known for 197 municipalities out of the 853 composing the state (see Figure 3.1.a). In the original experiments, 86 independent variables of various types were used to classify prevalence: Remote Sensing (22), climatic (6), socioeconomic (34) and neighborhood characterization (24). The Remote Sensing variables were derived from sensors MODIS (Moderate Resolution Imaging Spectroradiometer) and SRTM (Shuttle Radar Topography Mission), and are supposedly related to the snail habitat type. The climatic variables were obtained from the Weather Forecast and Climate Studies Center (CPTEC) from the National Institute for Space Research (INPE) and reflects the conditions of survival of the snail and the various forms of the larvae of Schistosoma mansoni. The socioeconomic variables were obtained from SNIU (National System of Urban Indicators) such as



(*a*) (*b*)

Fig. 1. The state of Minas Gerais in Brazil: a) known prevalence of Schistosomiasis, b) regionalization obtained through algorithm SKATER

the water accessing means and sanitation condition aspects. The neighborhood characterization variables measure the disparity between neighboring municipalities with relation to variables of income, education, sewerage, water access and water accumulation.

From the original 86 variables, a smaller set was selected, according to tests using multiple linear regression [7]; the independent variables chosen were those that had high correlation with the dependent variable and low correlation with other independent variables. Two main approaches were used: i) a global one, in which all the municipalities with known disease prevalence were used, for either constructing or validating a linear regression model, and ii) a regional one, in which the state was divided in four homogeneous regions and a linear regression model was created for each one of them. The number of independent variables used in the experiment varied; in the global approach 5 variables were used, while in the regional approach 2 variables were used for region R1, 5 for region R2, 4 for region R3 and 3 for region R4 (see details in [7]). In both the global and regional approaches, approximately 2/3 of the samples were used as training set, and the remainder 1/3 as the test set. Algorithm SKATER [3] was used to obtain the homogeneous regions in the regional model; this algorithm creates areas such that neighboring areas with similar characteristics belong to the same region (see Figure 3.1.b).

3.2 The Fuzzy CBR Experiments: Materials and Methods

This work uses the same data, variables and regionalization than those used in [7]. As already mentioned, to construct the similarities $S_D^{\mathbf{w}^c}$, our approach needs a similarity relation S_a for each independent attribute $a \in \mathbf{A}$ to be given. In our experiments all considered attributes are real-valued attributes. Then, for each attribute a we have taken S_a to be of the form $S_a = S_{\lambda_a}$, where S_{λ_a} is a parametrized similarity relation on $[0, 1]$ defined as

$$S_{\lambda_a}(x, y) = \max(0, 1 - \frac{|x - y|)}{\lambda_a \cdot length(a)})$$

where $\lambda_a \in (0, 1]$ and $length(a) = \max_{c \in CB} a(c) - \min_{c \in CB} a(c)$ is the maximal variation of a in the whole case base.

On the following, we will simply denote a similarity relation for a description attribute as S_λ, $\lambda \in (0, 1]$, and we will synthesize the notation of a set of such similarity relations as $S_{(\lambda_1,\ldots,\lambda_n)}$, meaning that S_{λ_1} is applied to attribute a_1, S_{λ_2} to a_2, etc.

The dependent attribute *Class* is defined in the domain $\mathbf{Cl} = \{L, M, H\}$, for *low*, *medium* and *high* disease prevalence, respectively. In the experiments, we have taken the similarity relation S_{Class} on $\mathbf{Cl}$ to be of the form $S_{Class} = T_\lambda$ for some $\lambda \in [0, 1]$, where T_λ is defined as $T_\lambda(w, w) = 1$ and $T_\lambda(w, w') = T_\lambda(w', w)$, for all $w, w' \in \mathbf{Cl}$, $T_\lambda(H, M) = T_\lambda(M, L) = \lambda$ and $T_\lambda(H, L) = 0$.

3.3 Experimental Results and Analysis

Table 1 brings the best results obtained from experiments made with the data, for a) the regression models employed in [7] and b) by the fuzzy CBR method. On Table 1.b, besides the CBR approach accuracy value, we have indicated the similarity relations used in the description and solution variables.

We have applied the fuzzy CBR method using various parameters sets for the description and class similarity relations. As aggregation function F in Definition 2, we have used the maximum (as proposed in [2]) and other operators; the best results were obtained with the arithmetic means (see Table 1). Notice that for region R1, we have obtained the same accuracy (.56) using similarities $(S_{(.3,.4)}, T_{.5})$ and $(S_{(.1,.1)}, T_0)$ for the regional learning approach.

Table 1. Classification accuracy, with learning made on either a global or a regional basis: (a) regression models and (b) fuzzy CBR models

Region	regional	global
R1	0.56	0.50
R2	0.51	0.40
R3	0.72	0.48
R4	0.76	0.59

(a)

Region	regional	global
R1	0.56 $(S_{(.3,.4)}, T_{.5})$	0.56 $(V_{(.2,.2)}, T_0)$
R2	0.56 $(S_{(.2,.2,.2,.2,.2)}, T_{.5})$	0.49 $(V_{(.1,.1,.1,.1,.1)}, T_0)$
R3	0.62 $(S_{(.2,.4,.3,.3)}, T_0)$	0.71 $(V_{(.2,.2,.2,.2)}, T_0$
R4	0.38 $(S_{(.2,.2,.2)}, T_{.5})$	0.65 $(V_{(.1,.1,.1)}, T_{.5})$

(b)

The results obtained with the fuzzy CBR approach are comparable to those obtained with the regression models and, in one case, the fuzzy CBR approach is better than regression (region R_2). It is interesting to note that in the fuzzy CBR approach, contrary to what happened with the regression models, the global learning approach have very often outperformed the regional one. As a matter of fact, the global fuzzy CBR approach obtained invariably better results than regression in the global approach.

4 Conclusions

In this work we have described the use of a similarity based fuzzy CBR approach to classify the prevalence of Schistosomiasis in the state of Minas Gerais in Brazil. We have compared our results to the ones obtained from the literature that uses linear regression. The comparison results shows the suitability of the approach.

The classification method is such that at the end of an experiment, we obtain a similarity degree between the description of a new case and each one of those in the case base. Then we derive the compatibility of the new case with each class value, by aggregating all the similarity degrees obtained from the cases in the case base that are classified with that value. Here we have verified that the usual operator, max, is not always the most suitable for a given application. In particular, in our problem, the best aggregation operator from those tested was found to be the arithmetic means.

Acknowledgments. The auhors acknowledge partial support of the CSIC-CNPq bilateral project, Ref. 2007BR0053. Godo and Sandri are also partially supported by the Spanish projects IEA (TIN2006-15662-C02-01), MULOG2 (TIN2007-68005-C04-01) and "Agreement Technologies" (CONSOLIDER CSD2007-0022, INGENIO 2010). The authors thank Vicenç Torra for unvaluable help in realizing the experiments and to an anonymous reviewer for helpful comments.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7, 39–59 (1994)
2. Armengol, E., Esteva, F., Godo, L., Torra, V.: On learning similarity relations in fuzzy case-based reasoning. In: Peters, J.F., Skowron, A., Dubois, D., Grzymała-Busse, J.W., Inuiguchi, M., Polkowski, L. (eds.) Transactions on Rough Sets II. LNCS, vol. 3135, pp. 14–32. Springer, Heidelberg (2004)
3. Assuno, R.M., Neves, M.C., Cmara, G., Freitas, C.C.: Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. International Journal of Geographical Information Science 20, 797–811 (2006)
4. Calvo, T., Kolesarova, A., Komornikova, M., Mesiar, R.: Aggregation operators: Properties, classes and construction methods. In: Calvo, T., Mayor, G., Mesiar, R. (eds.) Studies in Fuzziness and Soft Computing, vol. 97, pp. 3–104 (2002)
5. Dubois, D., Esteva, F., Garcia, P., Godo, L., Lòpez de Mantaras, R., Prade, H.: Fuzzy Set Modelling in Case-based Reasoning. International Journal of Intelligent Systems 13(4), 345–373 (1998)
6. Dubois, D., Prade, H. (eds.): Fundamentals of Fuzzy Sets. The Handbooks of Fuzzy Sets Series. Kluwer Academic, Dordrecht (2000)
7. Martins, F., Freitas, C., Dutra, D., Sandri, L., Drummond, S., Fonseca, I., Guimarães, F., Amaral, R., Carvalho, O.: Risk mapping of Schistosomiasis in the state of Minas Gerais, Brazil, using MODIS and socioeconomic spatial data (submitted)
8. Torra, V.: On the learning of weights in some aggregation operators: the weighted mean and OWA operators. Math. and Soft Comp. 6, 249–265 (2000)
9. Torra, V.: Learning weights for the quasi-weighted means. IEEE Trans. on Fuzzy Systems 10(5), 653–666 (2002)
10. World Health Organization, The Control of Schistosomiasis. Second Report of the WHO Expert Committee. Technical Report Series no. 830, Geneva (1993)

BAC Overlap Identification Based on Bit-Vectors

Jens-Uwe Krause and Jürgen Kleffe

Institute for Molecularbiology and Bioinformatics, Charité, Berlin, Germany
`Jens-Uwe.Krause@gmx.de, juergen.kleffe@charite.de`

Abstract. There is no software that accurately calculates the overlap of two BACs fast enough for application to thousands of cases in turn. The problems include unacceptably low speed of dynamic programming algorithms for sequences of the considered size and failure of the faster local alignment methods to identify complete sequence overlaps. Lower sequence quality at both BAC ends and internal difference blocks, being small enough to not significantly increase relative error rates but large enough to terminate local alignments, cause output of multiple overlapping local matches which do not extend to both sequence ends. Based on Myers' bit-vector algorithm for fast edit distance calculation, we developed the program BACOLAP, that identifies overlapping BACs just as sensitive as global dynamic programming alignment and as fast as local heuristic alignment.

1 Introduction

BLAST [2] is the most frequently used program for sequence comparison and known for its speed and reliability. It serves not that well for finding overlaps of BAC size sequences. Lower quality at sequence ends and difference blocks cause termination of local alignments before sequence ends are reached. Semi-global dynamic programming alignment is required to accurately solve the overlap problem. GAP3 [6] is such a program, extended to also handle difference blocks, and used by the read assembler CAP3 [5]. Just as CLUSTALW [12], this program takes about 30 minutes to process a single pair of BACs. Both methods are far to slow for large scale BAC comparison.

A number of programs like BL2SEQ [11], WABA [7], YASS [10] and NUCmer [3] are known, which join local matches to longer alignments but are not specialized to the overlap problem. For instance all these programs miss an overlap of length 37,493 with relative error 0.003, formed by the BACs AC124217 and AC198008, since the start of the overlap matches poorly over the first 250 nucleotides. The programs fail due to their inability to cope with gaps of length 17 to 85 and perhaps also programming errors. WABA even reports one case of two overlapping exact matches in place of one extended match. Efficient testing of large numbers of BAC pairs needs fast and reliable software, that directly outputs overlap events, including all sequence strand combinations and multiple overlap solutions if possible.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1061–1069, 2009.
© Springer-Verlag Berlin Heidelberg 2009

2 Methods

For edit distance Myers [9] developed a very fast bit-vector algorithm, that finds all approximate matches of a short sequence of length M within a longer sequence of length N. The time of computation depends on the maximum number of errors admitted for each match and on the number of feasible matches found. It rarely comes close to the worst case time that is proportional to $M \times N/W$, with the divisor W representing the length of the machine word, 32 or 64 depending on hardware. This algorithm provides an efficient method to match short prefixes or suffixes of one BAC with all of the other in order to find potential start or end positions of overlaps.

Next each potential overlap is tested using a global alignment tool, also based on the bit vector algorithm, in combination with the divide-and-conquer method by Hirschberg [4]. This programm made by Aiche et al. [1] takes time proportional to $2 \times V^2/W$, where V is the length of the overlap, that is often much smaller than sequence length. Both methods together find overlaps much faster than full semi-global alignment. Testing alternative pairs of start and end positions yields alternative potential overlaps.

2.1 Deriving Potential Sequence Overlaps

Assuming two BACs A and B to overlap we use Myers' bit-vector algorithm to match the first M characters of sequence B with sequence A. All matches with edit distance not greater than E are reported. But since Myers' bit-vector algorithm only outputs end positions of qualifying matches, it is actually applied to the reverse complement sequences and the end positions of matches are translated to start positions on the direct strand sequences.

It follows a list of positions in sequence A, where sequence B can possibly start as shown in Figure 1. Only start positions with locally minimum edit distance are considered. Given the global minimum edit distance is $G \leq E$, we even confine to local minima below the threshold $G + b(E - G)$, where b is a parameter between zero and one. It is not sufficient to select only the best matching start position but considering all is computationally impossible.

Given a potential overlap begins in position sA of sequence A and the maximal admitted error rate of the overlap is RE, the end of the overlap in sequence B is limited to the interval between $(1 - RE)(lA - sA + 1)$ and $(1 + RE)(lA - sA + 1)$, where lA is the length of sequence A. Both extreme values are attained if all errors are gaps, located in only one of both sequences. Matching the last M characters of sequence A with the derived interval of sequence B yields a number of potential end positions of overlaps in sequence B. If $minsA$ and $maxsA$ denote the most upstream and downstream start positions, respectively, a simultaneous interval of all potential end positions in sequence B is given by $(1 - RE)(lA - maxsA + 1)$ and $(1 + RE)(lA - minsA + 1)$. We actually search this interval for matches of the last M characters of sequence A with edit distance not greater than E. Myers' algorithm now correctly returns the end positions of matches. Again, we only consider end positions with locally minimum edit

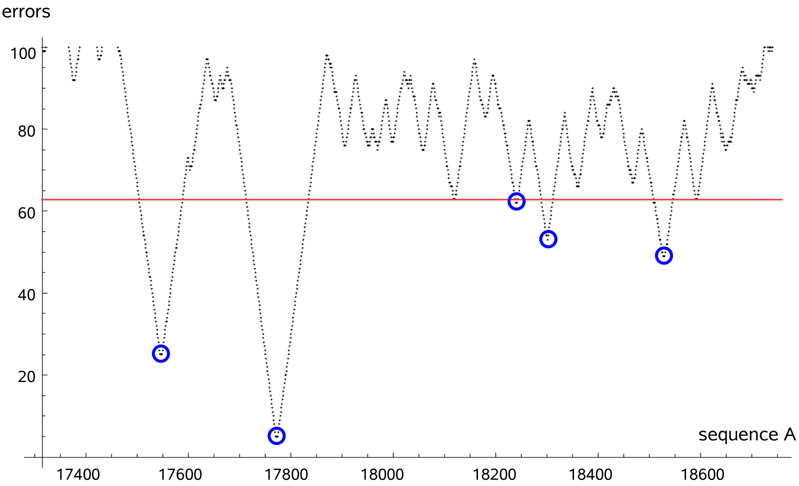


Fig. 1. BACOLAP's rule to select start and end positions of potential sequence overlaps. The vertical axis shows edit distance while the horizontal axis shows match positions. Circles show match positions with locally minimum edit distance below the considered threshold $G + b(E - G)$ for $E = 100$ and $b = 0,6$. Only these positions are considered possible start positions of overlaps.

distance below the threshold $G + b(E - G)$ where G is the global minimum edit distance.

If sequence B is sufficiently short, we also test it for being contained in sequence A. If lB denotes its length we match the last M characters of sequence B with the interval between $minsA + (1 - RE) \times lB$ and $maxsA + (1 + RE) \times lB$ of sequence A in order to locate possible ends of matches. We arrive at a list of paired sequence sections representing potential overlaps. The absolute length difference of each pair is a lower bound for the number of alignment gaps. It is not allowed to exceed the product of RE and the length of the longer sequence section.

Generally most pairs are ruled out by this condition. All others are aligned and those with relative error greater than RE are withdrawn from the list. A single pair remains in most cases. The entire analysis is repeated with switched roles of sequences A and B as well as sequence B replaced by its reverse complement.

2.2 Longest Common Substring Speed Up

Long overlaps with small relative error often contain long perfect matches. This suggests first identifying the longest common substring (LCS) based on suffix-trees and then limiting alignment calculation to both sides of the LCS. The resulting overlap includes the LCS but in case its edit distance may be seriously increased by a large number of gaps only necessary to force the LCS into the alignment. The speed increase grows with LCS length. To minimize the effort of LCS computation, it is only calculated for the pair formed by the longest subsequence of A and the longest subsequence of B, or its reverse complement, found in the list of derived potential overlaps. Then, whenever possible, this LCS is used to speed up calculation of individual overlap alignments, supposed it covers more than 50% of the overlap. While it is often fine to use a long LCS, using a short LCS is dangerous.

3 Data

Two data sets were studied to test BACOLAP. Data set 1 was derived from 1605 pairs of overlapping BACs published under `http://www.medicago.org`. Removing 421 pairs, that contained unfinished BACs or BACs with runs of more than 20 N-letters, we found 1184 pairs to study how fast and reliable BACOLAP confirms sequence overlaps.

A second set of 2505 unpaired BACs was downloaded from the same website. Using ClustDB [8], we derived 43,727 BAC pairs, that have a common substring of length 300. This sample represents data set 2 and is used to study how quickly BACOLAP rejects pairs which are not really overlapping.

4 Results

All computations were performed using a PC with a Pentium 4 processor and 2GB of memory running under Linux OS.

4.1 Speed and Sensitivity

Table 1 shows the time BACOLAP takes to process data set 1 and how many pairs it identifies to overlap. The sensitivity depends on finding start and stop positions of potential overlaps, regulated by parameters M (length of start and stop sections), E (maximum number of errors in these sections) and the maximum relative error ER, allowed for the complete overlap. The speed depends on these parameters and the use of the LCS option. Therefore Table 1 is vertically organized into five blocks for M ranging from 300 to 1000. Each block horizontally displays results for different values of $E = (M \times RE)/100$, where RE ranges from 5% to 33.3%. The reported results include the number of pairs identified to overlap and the total time of computation.

BACOLAP did not confirm 19 pairs to overlap using parameters $M = 300$ and $RE = 33.3\%$. Eight of these pairs were found to overlap using alternative parameters. In all other cases the true overlaps are too short, starting or ending subsequences do not match sufficiently well, or evidence of a sequence overlap was not found by any working alignment tool.

By its nature our method does not reliably detect overlaps, that are shorter than M. A detailed analysis shows, that overlap detection fails if $(M - E)$ plus the number of overlap errors exceeds overlap length. This fact in part causes the sensitivity to decrease for growing M and given $RE = 5\%$. Table 2 illustrates

Table 1. Run time of BACOLAP to confirm 1184 BAC overlaps of data set 1

error rate	5%	10%	20%	25%	30%	$33.\bar{3}\%$	
$M = 300$							
time measurement							∅
without lcs	32 min	31 min	36 min	37 min	40 min	44 min	36:40 min
with lcs	9 min	9 min	12 min	12 min	14 min	16 min	12:00 min
found overlaps	1152	1155	1160	1161	1164	1165	
$M = 500$							
time measurement							∅
without lcs	30 min	32 min	37 min	42 min	41 min	44 min	37:40 min
with lcs	10 min	10 min	13 min	14 min	15 min	14 min	12:40 min
found overlaps	1151	1156	1163	1166	1166	1166	
$M = 600$							
time measurement							∅
without lcs	31 min	32 min	36 min	41 min	45 min	45 min	38:20 min
with lcs	9 min	10 min	12 min	14 min	14 min	16 min	12:30 min
found overlaps	1150	1156	1165	1166	1166	1166	
$M = 750$							
time measurement							∅
without lcs	32 min	36 min	39 min	41 min	44 min	47 min	39:50 min
with lcs	9 min	10 min	12 min	13 min	14 min	18 min	12:40 min
gefundene Overlaps	1148	1153	1164	1166	1167	1168	
$M = 1000$							
time measurement							∅
without lcs	32 min	34 min	37 min	40 min	46 min	50 min	39:50 min
with lcs	10 min	11 min	13 min	13 min	15 min	16 min	13:00 min
found overlaps	1146	1155	1162	1165	1165	1165	

this rule for three overlaps less than 750 nucleotides long, which were identified for $M = 750$ but not for $M = 1000$, given $RE = 33.3\%$. At the other hand, for constant RE, growing M yields larger E values, which provide more freedom for errors to clump in the start and end sections of overlaps and hence increase sensitivity. These two antagonistic effects explain the complex relation between M and the number of identified overlaps seen in Table 1.

Table 1 also shows a growing time of computation for increasing RE, since more overlaps are tested. The LCS speed acceleration proved to be effective since almost every calculated overlap contains the LCS. It had no effect on sensitivity. However, by increased probability of providing not optimal edit distances, this option could imply rejection of an otherwise accepted overlap. This probability is the smaller the longer the overlap and the less errors we admit.

4.2 Comparison with Other Programs

Using parameters $M = 300$, $E = 100$ and $RE = 33.3\%$, we compared BACOLAP with BL2SEQ [11], WABA [7], YASS [10], NUCmer [3] and ClustDB [8]. The

Table 2. Three short BAC overlaps of data set 1 which could not be identified using window size $M = 1000$ and $E = 333$

BAC pair	**overlap length**	**errors**	`M-E+errors`
CR931737\AC144516	604	0	667
CR954197\AC136286	617	0	667
DQ009671\AC140551	750	185	852

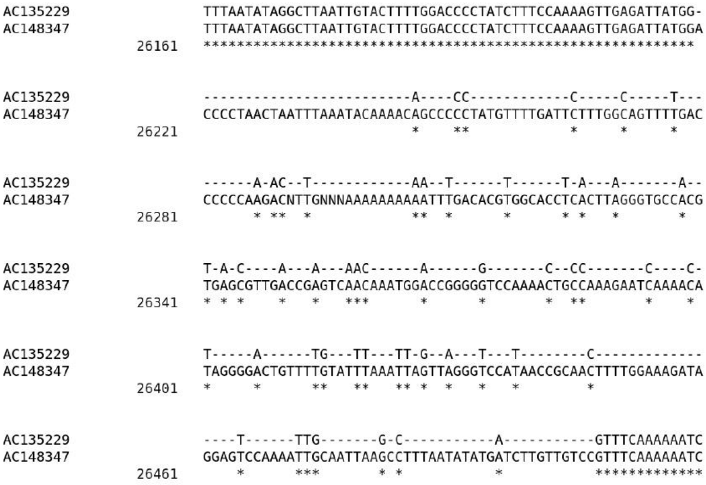
Table 3. Performance comparison of BACOLAP, ClustDB, BL2SEQ, WABA, yass and NUCmer using data set 1

	BACOLAP	YASS	ClustDB	WABA	NUCmer	BL2SEQ
confirmed overlaps	1165	1138	1133	1123	1119	1114
results compared with BACOLAP						
equal positions		1103	1093	1086	1081	1059
position difference maximal 20		34	36	35	37	45
position difference between 21 und 100		1	4	2	1	10
unconfirmed overlaps	19	46	51	61	65	70
time of computation	16 min	44 min	42 min	43h 59 min	11 min	12 min

parameters for the other programs were $gapcost = -2$, $gapextensioncost = -1$, $mismatchcost = -1$ and $matchcost = 1$ for BL2SEQ, default parameters for WABA and YASS, $minmatch = 300$ and $maxgap = 100$ for NUCmer, and 100 errors within each alignment window of size 300 for ClustDB. Table 3 shows how many overlaps the different programs found and how long they took for data set 1. BACOLAP found the largest number of pairs. The 19 pairs, BACOLAP failed on, were missed by all other programs, too. In many cases the competing software produced local alignments, very close to BACOLAP's results, with start and end positions differing by not more than 20 or 100 nucleotides, respectively. These cases were counted successful in Table 3 and still BACOLAP failed on less than half the cases the best competing program did. The programs YASS and ClustDB are closest to BACOLAP's sensitivity but take more than twice the time. WABA is extremely slow and clearly insufficiently sensitive compared to its long time of computation. The programs NUCmer and BL2SEQ are faster than BACOLAP but not enough to justify the loss of sensitivity and inability to report multiple overlap solutions.

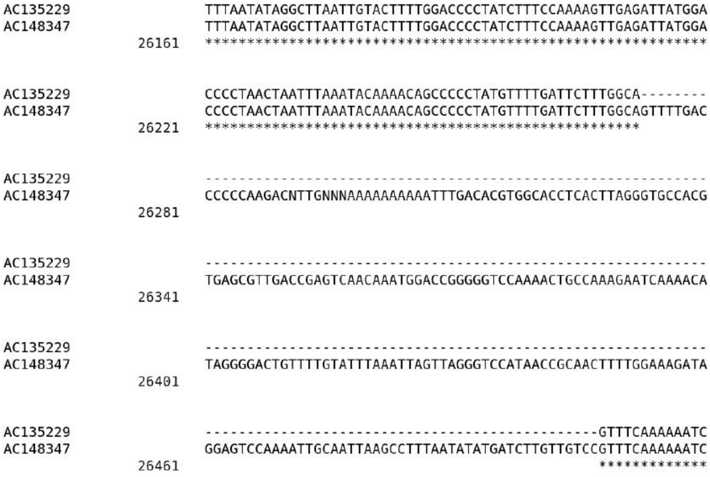
4.3 Alignment Post-processing

ClustDB failed on 32 BAC pairs of data set 1 which BACOLAP confirmed to overlap. These pairs all contain poorly matching regions, called difference blocks. We used these pairs to compare BACOLAP's difference block post-processing feature with the program GAP3 [6]. BACOLAP detected 48 difference blocks for which it produced the same result as GAP3 in 34 cases. In 7 cases neither GAP3 nor BACOLAP could improve the alignment, and in only one of the 7



```
AC135229                    TTTAATATAGGCTTAATTGTACTTTTGGACCCCTATCTTTCCAAAAGTTGAGATTATGG-
AC148347                    TTTAATATAGGCTTAATTGTACTTTTGGACCCCTATCTTTCCAAAAGTTGAGATTATGGA
                    26161   ***********************************************************

AC135229                    -------------------------A----CC------------C-----C-----T---
AC148347                    CCCCTAACTAATTTAAATACAAAACAGCCCCCTATGTTTTGATTCTTTGGCAGTTTTGAC
                    26221                            *    **            *     *     *

AC135229                    ------A-AC--T------------AA--T------T------T-A---A-------A--
AC148347                    CCCCCAAGACNTTGNNNAAAAAAAAAATTTGACACGTGGCACCTCACTTAGGGTGCCACG
                    26281         * **  *            **  *      *      * *   *       *

AC135229                    T-A-C----A---A---AAC------A------G-------C--CC-------C----C-
AC148347                    TGAGCGTTGACCGAGTCAACAAATGGACCGGGGGTCCAAAACTGCCAAAGAATCAAAACA
                    26341   * * *    *   *   ***      *      *       *  **       *    *

AC135229                    T-----A------TG---TT---TT-G--A---T---T--------C-------------
AC148347                    TAGGGGACTGTTTTGTATTTAAATTAGTTAGGGTCCATAACCGCAACTTTTGGAAAGATA
                    26401   *     *      **   **   ** *  *   *   *        *

AC135229                    ----T------TTG-------G-C-----------A-----------GTTTCAAAAAATC
AC148347                    GGAGTCCAAAATTGCAATTAAGCCTTTAATATATGATCTTGTTGTCCGTTTCAAAAAATC
                    26461       *      ***       * *           *           *************
```

Fig. 2. A poorly matching alignment section as presented by global alignment



```
AC135229                    TTTAATATAGGCTTAATTGTACTTTTGGACCCCTATCTTTCCAAAAGTTGAGATTATGGA
AC148347                    TTTAATATAGGCTTAATTGTACTTTTGGACCCCTATCTTTCCAAAAGTTGAGATTATGGA
                    26161   ************************************************************

AC135229                    CCCCTAACTAATTTAAATACAAAACAGCCCCCTATGTTTTGATTCTTTGGCA--------
AC148347                    CCCCTAACTAATTTAAATACAAAACAGCCCCCTATGTTTTGATTCTTTGGCAGTTTTGAC
                    26221   ****************************************************

AC135229                    ------------------------------------------------------------
AC148347                    CCCCCAAGACNTTGNNNAAAAAAAAAATTTGACACGTGGCACCTCACTTAGGGTGCCACG
                    26281

AC135229                    ------------------------------------------------------------
AC148347                    TGAGCGTTGACCGAGTCAACAAATGGACCGGGGGTCCAAAACTGCCAAAGAATCAAAACA
                    26341

AC135229                    ------------------------------------------------------------
AC148347                    TAGGGGACTGTTTTGTATTTAAATTAGTTAGGGTCCATAACCGCAACTTTTGGAAAGATA
                    26401

AC135229                    -----------------------------------------------GTTTCAAAAAATC
AC148347                    GGAGTCCAAAATTGCAATTAAGCCTTTAATATATGATCTTGTTGTCCGTTTCAAAAAATC
                    26461                                                  *************
```

Fig. 3. Alignment post-processing of the sequence section considered in Figure 2 reveals an insert in BAC AC148347

cases left, GAP3 found an alignment, that we would prefer over the result of BACOLAP. Figures 2 and 3 show an example of alignment post-processing.

4.4 Alternative Overlaps

An important feature of BACOLAP is its multiple output of solutions for 91 pairs of data set 1 while other programs provide just one. Such multiple

Table 4. Comparison of BACOLAP, YASS, BL2SEQ and NUCmer for 43,727 potential overlapping BAC pairs

	BACOLAP	YASS	BL2SEQ	NUCmer
confirmed overlaps	2538	2362	2031	1955
computation time	341 min	1123 min	347 min	407 min

solutions indicate repeats and possible assembly errors. For 22 pairs we even observed equally scoring multiple overlap alignments, where different start or end positions do not affect overlap length and number of errors. Such multiplicities are dangerous for chaining overlapping BACs.

4.5 Rejecting Pairs of Non-overlapping BACs

Table 4 provides the results obtained by application of BACOLAP, YASS, BL2SEQ and NUCmer to data set 2. Opposed to data set 1 it contains mostly non-overlapping pairs of BACs. BACOLAP identified the largest number of truly overlapping pairs and was the fastest. In difference to Table 3 BACOLAP was even faster than BL2SEQ and NUCmer. It more quickly rejects non-overlapping pairs by its fast bit-vector search for potential start and end positions. Hence it can afford an increased effort for identifying multiple overlap possibilities in the relatively small number of positive cases. Most importantly BL2SEQ and NUCmer found only 80% of the overlaps BACOLAP has identified. YASS is more sensitive than BL2SEQ and NUCmer but took by far the longest time of computation.

5 Discussion

We have applied bit-vector based functions implemented in the SeqAn C++ library [13] to solve the problem of finding overlapping regions between two BACs. The new tool was necessary since standard alignment software does not consider multiple possibilities of overlapping, decreased sequence quality at BAC ends and difference blocks, caused by assembly errors or copy number variations. The LCS speed up option helps to save time for handling large numbers of cases at very little loss of sensitivity. BACOLAP's field of application is best compared with that of GAP3 [6], which was developed to find overlaps of short reads but performs incredibly slow when it is applied to BAC size sequences. Standard alignment software does not produce a ready answer to the question if two BACs overlap.

Although, our algorithm performs satisfactory, it is of heuristic nature and more considered a beginning rather than the end of a necessary line of research that aims at efficiently solving the overlap problem for BAC size sequences. Different new generation sequencing technologies, equipped with new and efficient read assemblers, are going to produce giant sets of longer sequence contigs, which need to be carefully studied for many different reasons. In particular large plant genome projects, like that of sequencing Medicago truncatula, will profit using BACOLAP to derive BAC assembly tables.

References

1. Aiche, S., Döring, A., Kleffe, J.: Fast And Exact Global Sequence Alignment. In: German Conference on Bioinformatics (GCB 2007), Potsdam, Germany (September 2007)
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. 215, 403–410 (1990)
3. Delcher, A.L., Phillippy, A., Carlton, J., Salzberg, S.L.: Fast algorithms for large-scale genome alignment and comparison. Nucleic Acid Research 30, 2478–2483 (2002)
4. Hirschberg, D.S.: A linear space algorithm for computing maximal common subsequences. Commun. Assoc. Comput. Mach. 18(6), 341–343 (1975)
5. Huang, X., Madan, A.: Cap3: A dna sequence assembly program. Genome Research 9, 868–877 (1999)
6. Huang, X., Chao, K.-M.: A generalized global alignment algorithm. Bioinformatics 19(2), 228–233 (2003)
7. Kent, W.J., Zahler, A.M.: Conservation, regulation, synteny, and introns in a large-scale c.briggsae-c.elegans genomic alignment. Genome Research 10, 1115–1125 (2000)
8. Kleffe, J., Möller, F., Wittig, B.: Simultaneous identification of long similar substrings in large sets of sequences. BMC Bioinformatics 8(suppl. 5), S7 (2007)
9. Myers, G.: A fast bit-vector algorithm for approximate string matching based on dynamic programming. Journal of the Association for Computing Machinery, 46(3) (Mai 1999)
10. Noe, L., Kucherov, G.: Yass: enhancing the sensitivity of dna similarity search. Nucleic Acid Research 33, 540–543 (2005)
11. Tatusova, A.T., Madden, T.L.: Blast 2 sequences - a new tool for comparing protein and nucleotide sequences. FEMS Microbiol Lett. 174, 247–250 (1999)
12. Thompson, J.D., Higgins, D.G., Gibson, T.J.: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acid Research 22, 4673–4680 (1994)
13. SeqAn, `http://www.seqan.de`

AAL and the Mainstream of Digital Home

Esteban Pérez-Castrejón and Juan J. Andrés-Gutiérrez

Telefónica Investigación y Desarrollo SAU, Emilio Vargas 6,
28043 Madrid, Spain
`{esteban,jjangu}@tid.es`

Abstract. The technological standards for the Digital Home play a key role in providing entertainment services but also Ambient Assisted Living (AAL) services for the elderly. Both kind of services are multimedia and share the Digital Home infrastructure or Smart Home Ecosystem. This Ecosystem is made up of a growing number of devices that communicate with other local or remote devices. AAL services cannot be developed for the Digital Home in the "a technical solution for every problem" model and has to rely on existing and widespread technologies provided by this mainstream of technology.

Keywords: AAL, Ambient Assisted Living, digital home, ADL, activities of daily living, UPnP, DLNA, OSGi, HomeDA.

1 Introduction

1.1 Demographic Challenges and the Elderly Wellness

Before 2050, the percentage of people aged 60 years or more in Europe will be around 37%. In the developed world, the very old (aged plus than 80) is the fastest growing population group [1]. When adults get older they must face a range of important issues related to health, social inclusion, security, mobility and, in general, all factors that affect the wellbeing. The challenge that health and social care providers are facing to answer these demands is generating a huge social spending that are deemed unsustainable by all involved players. Today there are four working people for every retired but it is expected that in 2050 there will be only two working people for every retired. Furthermore, the quality of life of older people must be ensured to promote their independence and, in general, their social inclusion.

Many of these challenges are not new in our societies and are related to the diseases that may eventually accompany the aging process. This health demand is well known, deeply studied with a very complete statistical data [2] and is well structured due to the secular exercise of medicine (even though with the problem of coping with chronicity as a new factor). Our greatest challenge is, rather, that the elderly will be able to maintain their welfare state and to obtain the care they need when some of their functional abilities (memory, mobility, cognitive skills ...) decline, so that, they remain active members of the society in which they are developing their daily lives [3].

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1070–1082, 2009.
© Springer-Verlag Berlin Heidelberg 2009

1.2 Ambient Assisted Living (AAL)

Ambient Assisted Living (AAL) is a concept which offers a way to solve the challenges we are facing related to our ageing societies. AAL is related both to Information and Communication Technologies (ICT) as well as organizational innovations. In this context:

> "AAL aims to prolong the time people can live in a decent way in their own home by increasing their autonomy and self-confidence, the discharge of monotonously everyday activities, to monitor and care for the elderly or ill person, to enhance the security and to save resources."[4]

AAL intends to maintain the wellness and skills of the elderly not merely by providing an external human help but enhancing a self-managed life. Like some other technologies that have been in use for some time now, ICT comes to the help of the elderly in order to provide a longer and independent life.

Often, the concept of AAL is reduced to the health status associated with aging. But as shown in the following figure, AAL is neither "Health & Wellness" in a broad sense nor "Disease Management" for the chronically ill elders. AAL includes these concepts but focus on the whole person so that its ability to age independently in every aspects of daily life is strengthened. Actually and regarded in a comprehensive way, the structure of AAL could be applied to any member of the society irrespective of their age.

Only just to understand the point of view of AAL related to the independent life of the elderly, we can focus on the health component. AAL, in this case, is not seeking just to meet the needs in case of a health *crisis* (due to an acute or chronic illness) or to detect situations of a *decline* in health, but rather to *motivate* the development of habits that promote a healthy life.

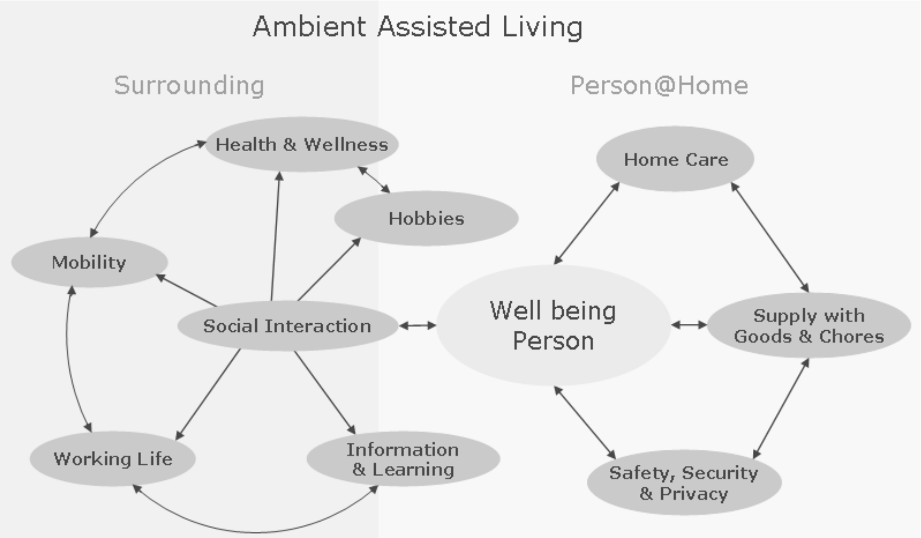


Fig. 1. Ambient Assisted Living: needs and opportunities [5]

1.3 Technology and Activities of Daily Living (ADL)

Activities of Daily Living (ADL) include a broad range of complex activities performed by human beings. They include physical activities like eating, bathing, walking and putting on clothes but also complex social and cognitive abilities like shopping and using money or recognizing other people. The possibility of successfully performing daily life activities determines the chance of remaining or not in the community. As a consequence, ADL have been chosen as markers of global human independence and functionality.

The overall functionality of a human being (as a complete set of ADL successfully performed) can be represented against lifetime as shown in the next figure.

A flat line meaning no loss of functionality over a lifetime is the ideal ADL profile. A real profile shows a decline in the capability to perform all the ADL and therefore results in a loss of the ability to maintain an independent living when people get older. [6]

As a result, a more or less vague line of dependency intersects the vital falling ADL function. After someone enters the dependency area (large shaded rectangle), human assistance by means of relatives or formal carers is needed to complete ADL and to maintain a good quality of life.

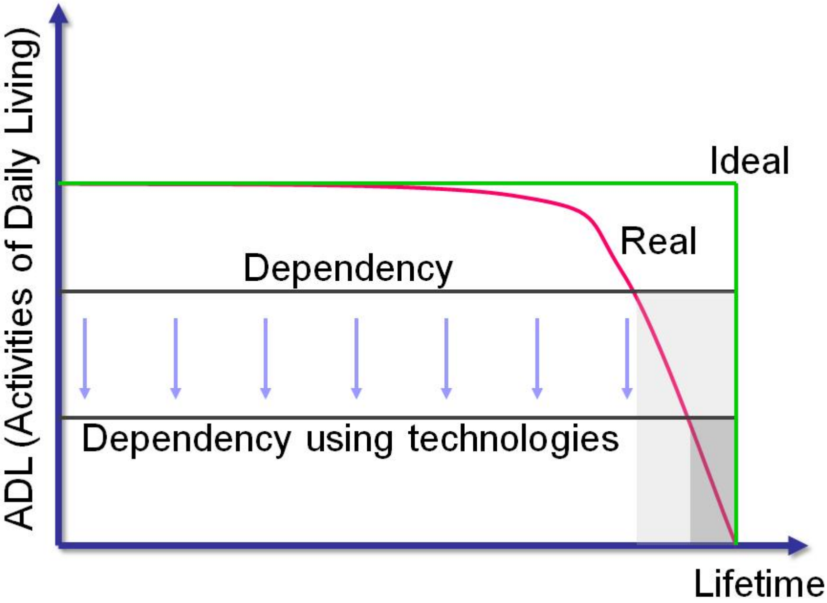


Fig. 2. Ideal and real ADL vs. lifetime. Dependency period reduced by technologies [6].

In this context, the Ambient Assisted Living paradigm considers information and communication technologies (ICT) as a powerful tool to reduce the need for human aid involved in dependency situations besides reducing the time and intensity of care needed for elderly people. Following our graphical approach the appropriate use of ICTs in an AAL context means to take down the dependency line. The area of the “dependency rectangle” is then reduced in terms of time and dependency depth.

In fact, IC technologies follow the track of mechanical technologies in the help of the elderly and in general terms dependent people: walking sticks, ramps, bathroom accessories but also glasses, wheelchairs and hearing aids are more or less sophisticated technologies that have been in use for some time now. Applying new technologies to this problem is a must as the population gets older and the sum of "dependency rectangles" surpasses the capabilities of society.

The AAL paradigm by means of ICT includes the support for older people performing ADLs in different scenarios: at home, giving better quality of life and more independence; at work, allowing an active and productive life, and in the community, overcoming isolation and loneliness and keeping up social networks.

Most of the AAL services an elderly person is offered when staying at home can be mobilized outside the home by means of the ICT: when travelling, walking, working or visiting other members of the community, the AAL-enabled home remains the centre of every Personal Digital Environment as the services an elderly person needs travel with him or her. As a consequence services should be designed to be flexible because they are not provided in a single user context or using a single user interface or physical device.

For future research, it will be desirable to establish the semantics of ADL to help modeling AAL services. Current definitions of ADL are health oriented or not sufficiently descriptive, such as the ICF (International Classification of Functionality). A semantic model of this type could be used by intelligent agents to detect ADL and provide a technological support to accomplish them without external human support.

2 The Mainstream of Technology for the Digital Home

Nowadays the Digital Home or smart home is being driven primarily by entertainment services. These kind of multimedia services make up the mainstream of technology for the Digital Home. Social or health services with technological components near the concept of AAL that are currently being developed, run the risk of being left outside the mainstream of Digital Home. These services do not reflect the standards of the Digital Home in areas such as service and device discovery, software and hardware architecture, user interfaces, remote management... Unlike present applications of telemedicine for niche patients, the AAL services at home should be thought as a cheap and mass market due to its dimension. At present, systems based on the AAL paradigm are being planed in an ad-hoc basis: a solution to every problem, regardless of the whole real life conditions and complete needs of the elderly. And often, this solution is designed monolithically and without the capability to adapt and reuse its components.

As an example, devices that are used for AAL applications in the Digital Home should offer their services in a standardized way. To achieve this we need a common layer that allows the personal or environmental sensors announcing its existence, the services they offer and a method to access them. When a new device is introduced or removed this common layer allows to use it with minimal user intervention. The AAL service provider should be able to remotely manage these devices and services, to add new services even without the user knowledge and to diagnose faults even before they occur. Only in this way mass services beyond a pilot with a limited number of users will have a significant social relevance.

The Digital Home can be seen as a number of technological areas that deal with different user needs: home automation and appliances, entertainment, security, home health and telecare. All of them have developed their own standards that cover specific services but gives as a result a fragmented Digital Home.

The mainstream of Digital Home is, at the moment, dominated by the entertainment sector not by health, social, wellness or, in a broad sense, AAL services. The European Commission recognizes that the ageing needs are not yet in the mainstream of IC products development for European companies due to [7]:

- Insufficient awareness of market opportunities and users' needs
- Older users' needs not built into new designs and technologies
- Fragmentation of research and innovation efforts

The entertainment sector has a key role in providing the information conduit to and from all the other smart systems in the Digital Home as it is nowadays probably the most standardized. Entertainment and AAL services are multimedia by nature. They rely on service components such as video, audio, formatted data, a growing number of devices (with services they offer to other local or remote devices), user interfaces over different devices and remote service management. Because of its similarity, this is the time for AAL services to join the mainstream of technology for the Digital Home if we want to have cost effective and not isolated AAL services.

Current homes must face an ever-growing number of consumer electronic devices. These are intended for family entertainment and also, to a lesser extent, to help in Activities of Daily Living (ADL). However, most of them work in isolation, and require some effort in order to make them operate using the full range of their capabilities. AAL users need simplicity, that is to say, they want technology to be a servant of users, and not the opposite. The challenge is to make the complex as easy to use as possible.

Because entertainment at home relies on consumer electronics and broadband communications, we will have a look on the standards these sectors are using. The AAL services for the elderly are *not required*, of course, to use these technologies as a must but the underlying ideas behind them are shaping the smart home today and will do it in the future. AAL services and the ADL support they provide will not be ubiquitous, cheap (and then mass market) and manageable if they continue to be built from scratch in every research or commercial project.

2.1 Universal Plug and Play (UPnP)

UPnP [8] is an industrial initiative to simplify the interconnection of devices in local networks. The Universal Plug and Play allows to discover devices and also to enumerate the features and services provided by each device. It also defines the communication protocol to invoke the services offered by the UPnP devices. UPnP offers the following features, (1) provides automatic discovery of any devices connected to the home network, (2) allows zero-configuration networking, (3) is independent of the type of devices and networks, (4) uses standard protocols and (5) offers easy system extensibility.

Those manufacturers who do not want to start an AAL device production from scratch will be able to use this standard in order to distribute the information about services the device offers throughout the home network. A complex AAL service can be built on top of a number of devices which offer a standard view of their capabilities.

UPnP drawbacks have been identified by some projects but new extensions can be added in order to overcome them without breaking the UPnP standard. [9]

For future research, it will be useful to define an UPnP control point based on an AAL specific ontology for the provision of services at home. In this way, a non-standard UPnP device could be easily integrated into AAL services through a semantic reasoning.

2.2 Digital Living Network Alliance (DLNA)

DLNA [10], which stands for Digital Living Network Alliance, is a system that allows different devices in the same network to be connected with each other to share content easily and without the need of complex settings. DLNA uses a subset of UPnP-oriented features for exchanging multimedia content.

In a DLNA network there are two types of devices: clients and servers. Servers (or Digital Media Server DMS) are the ones which have contents available to offer, while clients (or DMP Digital Media Players) are those where we can enjoy these contents.

This way we can access multimedia contents from any device regardless of their location. In fact, provided that there is enough bandwidth, not only *home-to-home* content can be shared allowing for complex social networks but *home-to-provider* which means that an external system could help an elderly person to perform remotely an Activity of Daily Living (ADL).

In the frame of the AAL paradigm, the use of multimedia content is mainly enclosed within the “social interaction” group and its connected categories such as mobility and working life.

2.3 DSL Forum TR-069

Devices connected to the home network demand in many cases services which are located outside the home network. For this reason the central point element in the home network is a CPE (Customer Premises Equipment) such as a residential gateway. It will manage all subscribed services working as a central point to control all the devices in the home network and provinding an integration platform for several internal and external services.

The Broadband Forum TR-069 [11] standard allows the remote management of the customer equipment: auto-configuration and dynamic provisioning, updating the device's software, monitoring and diagnostics.

Besides that, the TR-111 standard enhances the TR-069 by giving the ability to remotely manage devices that are connected *inside* the home network via the residential gateway. In this way the management protocols are not limited to Internet access devices, but can reach all types of devices in the home network such as web cameras, IP phones, personal or environmental sensors, etc. Moreover, it allows to deploy and to manage services on demand. In most real situations, TR-069/TR-111 will be a standard used by the telecommunication operator or by a social or health services provider to access the low level services of the home devices.

TR-069 is designed to handle millions of devices due to its scalability and robustness. Also interoperability is guaranteed through standardized objects models and plug fest.

2.4 Open Services Gateway Initiative (OSGi)

OSGi [12] defines a development platform based on components and provide a service oriented architecture. One of its advantages is to allow the manufactures to develop services while minimizing the probability of platform fragmentation (as it happens in the Linux world and its uncountable distributions). OSGi provides a dynamic module system based on Java components that nowadays is working in a big number of environments.

Some benefits of using OSGi's component system are:

- Device Management: remote management of home devices and devices database administration.
- Component Communication: efficient communication interface between OSGi components.
- Class-sharing: OSGi support shared components between application packages, so the same components must not be duplicated.
- Runtime Environment: OSGi can be used in different Java environments. It defines the minimum runtime requirements and any Java implementation, that meets those requirements, can be adequate to run OSGi. As a result, it provides a great flexibility that allows OSGi to be applied on multiple devices.
- Component Model: sharing of OSGi components plays an important role during the services lifetime. Components can be remotely added, deleted or modified by an external operator.
- Service Model: accessing to services in OSGi is really simple. OSGi services can be used as plugins for other applications.
- Application Model: OSGi defines an application model that can host any existing implementation model. This allows to avoid applications fragmentation, as it happens with Linux. This is especially important for an external operator trying to deploy a number of dynamic services as it can choose a single application model and deploy it on all platforms.
- Mobile manageability: the MDM (Mobile Device Management) includes device configuration, application deployment, status monitoring and data backup. Also JSR-232 defines a service platform designed to manage mobile devices and to control the installed applications.

OSGi is a platform very suitable for the deployment of AAL applications. AAL services are not static and its evolution involves a change of their associated technology. The life cycle of AAL applications includes installing, starting, stopping, updating and uninstalling components.

In order to clarify the concepts, the next figure shows the Home Gateway and the remote Management System besides the OSGi and TR-069 management areas and relationships.

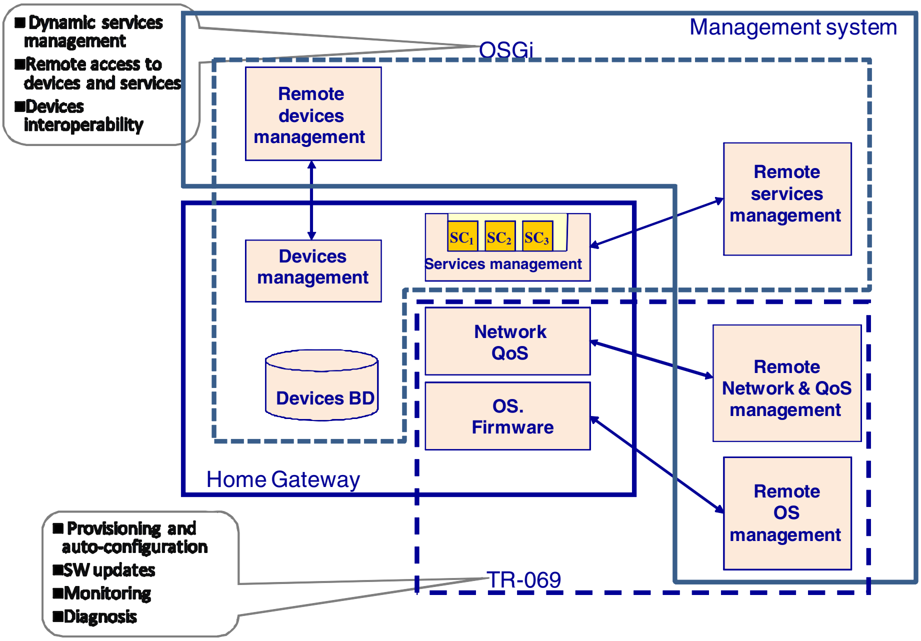

■Dynamic services management
■Remote access to devices and services
■Devices interoperability
Management system
OSGi
Remote devices management
Remote services management
Devices management
SC1 SC2 SC3
Services management
Network QoS
OS. Firmware
Devices BD
Remote Network & QoS management
Remote OS management
Home Gateway
■Provisioning and auto-configuration
■SW updates
■Monitoring
■Diagnosis
TR-069

Fig. 3. Combined management model based on TR-069 and OSGi

2.5 Universal Remote Console (URC)

The Universal Remote Console (URC) framework specified as a family of standards (ANSI INCITS 389-2005 to 393-2005 and ISO/IEC 24752) defines a "Protocol to Facilitate Operation of Information and Electronic Products through Remote and Alternative Interfaces and Intelligent Agents". [13] Its aim is to define a *user interface* layer over any existing framework for device discovery and control. The different user interfaces are defined by specifications based on XML. The "User Interface Socket" is the most important component of the URC framework. This socket is defined by a common semantic model for all user interfaces that can be used as a Machine-to-human (M2H) interface for a device such a as a TV set, a blood pressure cuff or a motion detector. Sophisticated or simple user interfaces can be connected to the socket therefore reusing software code enclosed in the user interface socket.

The Universal Control Hub (UCH) has been proposed as an approach for implementing the URC framework over UPnP networks. User interface clients can discover remote user interfaces to control networked devices. These UI are served by Remote UI Servers following the UPnP Remote UI specification. The URC provides the UPnP architecture with a common user interface level, so allowing UPnP devices to offer their own user interfaces on remote UI clients without any specific knowledge of them.

Some research projects such as Monami [14] and i2home [15] are using UCH as the framework for UI management in home applications including AAL ones.

2.6 Continua Health Alliance

The medical device manufacturers' viewpoint for the AAL at home is the Continua Health Alliance [16]. It is not a standardization forum but it applies existing standards primarily based on the ISO/IEEE 11073 family. However, as mentioned before, AAL is not only health and wellness. As a consequence this family of standards covers a certainly important field but not the entire spectrum of AAL applications. Also, these standards are designed to work as a "Data Optimized Exchange Protocol Layer" over Bluetooth and USB but they do not take into account features such as scalability, device discovery, services and devices management... The scenarios defined by Continua Health Alliance involve also HL7 standards and the IHE forum, which reinforces their health character. Besides that, mass market AAL applications have to include a full life cycle cost model (implementation, operation, remote management...). Also, social and health professionals are scarce resources and they have to deal with a large and growing number of patients and elders. But the Continua Alliance describes a future scenario for AAL applications from an overabundance of resources perspective [17].

2.7 Smart Home Ecosystem for AAL Services

The literature on AAL projects is plentiful of details associated with transport technologies (Bluetooth, USB, ZigBee, Z-wave, Wifi and RFI) but it lacks of solutions for the OSI application level. The features of the standards outlined in this section are faced, if they are considered at all, with ad-hoc developments that allow small pilots to be successful but certainly do not take into account a massive extension of the AAL services.

The former standards make possible to define an architectural structure that combines the individual benefits of all technologies to provide remote AAL services. This approach allows to discover devices in a local network and also enumerate the features and services provided by each device combining the following characteristics: (1) provides automatic discovery of any device connected to the home network; (2) allows zero-configuration networking; (3) is independent of the type of devices and networks; (4) uses standard protocols; (5) offers easy system extensibility and makes possible to integrate new AAL devices and services in a transparent way; (6) provides complex services thanks to services cooperation; (7) models the specific-AAL home domain for the use of a knowledge-based system according to context information sources; (8) takes into account a full life cycle cost model for mass market AAL services.

With these technologies we can create a Smart Home Ecosystem for AAL services. This ecosystem of devices would comprise the following components that include the features listed in this section:

- Home Gateway, acting as a communications hub and as a third party services deployment platform based on OSGi.
- Home Content Centre for content storage and aggregations. Most of the AAL services need ubiquitous multimedia content discovering and sharing.
- HomeDA or Fourth Home Screen acting as a tactile, vocal and visual user interface. Besides the three traditional screens (TV, computer, mobile) this Fourth Home Screen or Home Digital Assistant (HomeDA) offers rich user interface capabilities as well as AAL services processing.

This Smart Home Ecosystem is not intended to be the only way to provide multimedia AAL services at home. These devices can be seen as the *physical support* for the functionalities depicted in this section. However, its physical layout, interfaces, new capabilities, integration of existing technologies, etc, can be highly variable over the time and on the specific AAL application.

3 AAL Applications Using the Smart Home Ecosystem

3.1 Multi-Agent System for Personal Health Monitoring (MIMO) [6]

The MIMO (Multi-Agent System for Personal Health Monitoring) focus some of the above mentioned technologies on a Personal Health Monitor integrated with a Multi-Agent System for home environments.

The goal of this system is to explore the use of UPnP as the service and device discovery technology and Ambient Intelligence techniques in order to allow the user interaction at home as simple as possible. In this way we are trying to make the technology "disappear" because, after a short period of time, the user is not aware of using it. To achieve this goal a multi-agent system is proposed, focusing on knowledge representation for multi-agent communication and personal health monitoring.

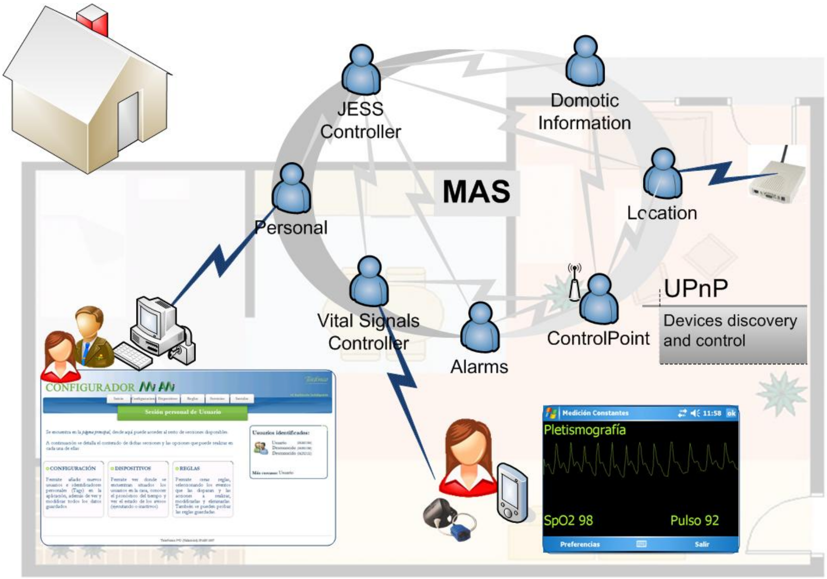


Fig. 6. Multi-Agent System for Personal Health Monitoring (MIMO)

When the user gets home with his new vital signs monitor, the device is discovered and plainly integrated within the Smart Home Ecosystem. Hence, one can anticipate the following challenges in a home environment: (1) standardizing device access and (2) making easy the introduction of new devices with a minimum of user interaction.

Briefly, the most relevant functional aspects of this proposal are:

- Intelligent Agents for monitoring vital signs of users at their homes
- To remotely consult vital constants from outside their homes
- An alert mechanism that can act in case of anomalous situations
- Automatic discovery and integration of e-health and domotic devices with the home network using UPnP.

While the MIMO is not detecting an abnormal situation, it does not require the involvement of a distant social or health professional. The intelligent agents include a number of events under which the external intervention is triggered. Also, this layer of ambient intelligence can be modified remotely without any user intervention to adapt itself to new circumstances.

3.2 AmIVital: Personal Digital Environment for Health and Welfare [18]

The main goal of the AmIVital project is to develop a new generation of technologies and ICT tools to model, design, implement and operate devices and systems for Ambient Intelligence (AmI). Its purpose is the provision of personal services and the support of independent living, welfare and health. It is about building a platform that makes easy to put into practice AAL services taking into account not only the technical applications but also new business models.

In line with the objectives of the Spanish CENIT program, AmIVital addresses R&D technology objectives in areas relevant such as AmI, advanced sensor networks such as intelligent sensors, wireless networks, embedded systems, intelligent systems based on context, multimodal interfaces, issues on accessibility and universal access, reliability, security, systems integration and interoperability, as well as remote software to support the management of AAL services and devices.

In particular, the main aim is to develop technologies and tools to implement a new generation of AAL applications and services for:

1. Supporting the independent living and mobility of the elderly.
2. Monitoring and controlling people with chronic illnesses, people requiring long term care, or frail people.
3. Helping people with disabilities.
4. Supporting services for families and informal carers.
5. Giving aid and services to vulnerable groups.
6. Making easy healthy lifestyles and promoting a proactive behavior of people in the course of their lives.
7. Supporting social and health professionals involved in the provision of care.
8. Operating and managing the AAL services included in the AmIVital platform.

3.3 Home Digital Assistant (HomeDA) or Fourth Home Screen

The HomeDA (Home Digital Assistant) or Fourth Home Screen is an evolved tactile, vocal and visual user interface that focuses on families. This Fourth Home Screen is usually linked to a fixed broadband line through the Residential Gateway and acts as a personalised window to a number of services related to entertainment as well as to AAL.

The Home DA is being driven by the emerging technological trends such as multi-touch screens, low power wireless, cheaper silicon, higher battery capacity, location services and the expansion of the open source movement in operating systems along with the explosion of digital information need at home. Commercially this kind of devices is a family HomeDA that allows a shared use for all the members of the family, apart from an individual use.

These new kind of devices will change the way the elderly access to the community and experiment the world around them:

- They will have access, at home and at anytime, to their virtual or real community and in particular to their family
- They will have an easy way to interact with the automation devices at their homes and will change the way they access and share their digital memories

The HomeDA for AAL is provided with a touch screen that has widgets for quick access to an enriched family agenda and calendar, personal content such as photographs and videos, telephony and messaging, emergencies, virtual communities, local services, shopping and so on. These features provide a new kind of devices with lots of applications for entertainment, communication and information at the Digital Home including automation and Activities of Daily Living monitoring.

Most manufacturers of these emerging devices use Linux [19] as OS but there is a trend to use Android [20] in some of the more advanced ones. Several communications operators are launching these devices and they are focused on families [21]. HomeDA are designed to support UPnP and OSGi and to be remotely controlled by TR-069. These standards allow for easy deployment of new AAL services, such as those based on the measurement of health parameters using the specifications of the Continua Alliance.

4 Conclusion

Ambient Assisted Living (AAL) is related to Information and Communication Technologies (ICT) as well as organizational innovations. The AAL concept offers a way to solve the challenges of our ageing societies and to provide an independent life for the elderly. Older people are helped by ICT while they carry out Activities of Daily Living at home or, in broad sense, in their Digital Personal Environment.

The entertainment sector can be seen as the *mainstream of technology* for the Digital Home and it is nowadays probably the most standardized. These standards, such as UPnP, DLNA, OSGi, URC and so on, play a key role in providing entertainment but also AAL services because both of them are multimedia and they share the Digital Home infrastructure or Smart Home Ecosystem. This Ecosystem is made up of a growing number of devices that communicate with other local or remote devices. Thanks to the aforesaid standards, these devices are remotely managed and can work together to provide a high number of AAL services.

Because of its similarity, this is the time for AAL services to join the mainstream of standards and technology for the Digital Home. Doing this we will be able to have AAL services that can be cost effective and not isolated. AAL services can not be developed for the Digital Home in the "a technical solution for every problem” model and has to rely on existing and widespread technologies.

AAL services and the ADL support they provide will not be ubiquitous, mass market and manageable if they continue to be built without taking into account the mainstream of technology for the Digital Home. As Ravel said about Brahms of writing concertos not for the piano but against it, sometimes we are building AAL services against the Digital Home not for it.

References

1. World Health Organization. Ageing and life course, `http://www.who.int/ageing/en/`
2. The Silver Book, `http://www.silverbook.org/`
3. World Health Organization. Strengthening active and healthy ageing, `http://www.who.int/nmh/a5819/en/index.html`
4. Report: Europe Is Facing a Demographic Challenge Ambient Assisted Living Offers Solutions. The Ambient Assisted Living (AAL) Joint Programme, `http://www.aal-europe.eu/`
5. The Ambient Assisted Living (AAL) Joint Programme, `http://www.aal-europe.eu/`
6. Vicente Tocino, A., Calvo Alcalde, A.I., Andrés Gutiérrez, J.J., Álvarez Navia, I., García Peñalvo, F.J., Pérez Castrejón, E., García Bermejo, J.R., Vegas Hernández, J., Cardeñoso Payo, V.: MIMO: Multi-Agent System for Personal Health Monitoring. In: Manuela Cunha, M., Tavares, A., Simoes, R. (eds.) Handbook of Research on Developments in e-Health and Telemedicine: Technological and Social Perspectives. IGI Global, Hershey (in press, 2009)
7. Stimulating ICT based R&D and Innovation on Ageing Well, `http://www.aal-europe.eu/Published/aal-2009-2/090213_ec_pwj`
8. Kim, D., Lee, K., Song, H., Sung, J.: UPnP-Based Sensor Network Management Architecture. In: Second International Conference on Mobile Computing and Ubiquitous Networking (2005)
9. Kistler, R., Knauth, S., Klapproth, A.: UPnP in Integrated Home- and Building Networks. In: IEEE International Workshop on Factory Communication Systems, Dresden (2008)
10. Digital Living Network Alliance, `http://www.dlna.org/home`
11. Customer Premises Equipment WAN Management Protocol v1.1, `http://www.broadband-forum.org/technical/download/TR-069Amendment2.pdf`
12. OSGi Alliance, `http://www.osgi.org`
13. Universal Control Hub & Task-Based User Interfaces, `http://myurc.org/publications/2006-Univ-Ctrl-Hub.php`
14. Mainstreaming on Ambient Intelligence, `http://www.monami.info/`
15. Intuitive Interaction for Everyone with Home Appliances based on Industry Standards, `http://www.i2home.org/`
16. Continua Health Alliance, `http://www.continuaalliance.org/home`
17. Continua Health Alliance. Vision Video, `http://www.youtube.com/watch?v=jbvmGqmIxXY`
18. AmIVital Entorno Personal Digital para la Salud y el Bienestar, `http://www.amivital.es`
19. Open Peak, `http://www.openpeak.com/OpenFrame2.php`
20. Touch Revolution, `http://www.touchrev.com/`
21. O2 Joggler, `http://o2family.o2.co.uk/`

Legal Concerns Regarding AmI Assisted Living in the Elderly, Worldwide and in Romania

Luiza Spiru[1], Lucian Stefan[1], Ileana Turcu[1], Camelia Ghita[1], Ioana Ioancio[1],
Costin Nuta[1], Mona Blaciotti[1], Mariana Martin[1], Ulises Cortes[2],
and Roberta Annicchiarico[3]

[1] Ana Aslan International Academy of Anti-Aging,
Piata Kogalniceanu Nr.1, 050064 Bucharest, Romania
[2] Universidad Politecnica de Catalunia, Jordi Girona 1, 08034 Barcelona, Spain
[3] Fondazione Santa Lucia, Via Ardeatina 354, Rome, Italy
`lsaslan@brainaging.ro`

Abstract. The recent concepts developed in order to meet the specific technology-related needs of older people are “Ambient Intelligence” and "Ambient Assisted Living". The major aim of AmI is to prolong the time older people can live decently in their own homes with an increased autonomy and self-confidence. One category of patients that would necessarily require intelligent ambient assistance devices in the future is the old one, characterized by a high incidence of co-morbid conditions and diseases, cognitive and/or physical impairment, functional loss from multiple disabilities, leading to impaired self-dependency. US and EU law and research programs are concerned to support the research, development and implementation of such devices able to reduce the social burden and to improve the geriatric healthcare systems.

Keywords: assistive technology, elderly, legislation.

1 Background

A recent concept to adequate technological supply in meeting the specific technology-related needs of older people include “ambient intelligence”(AmI) and "Ambient Assisted Living" (AAL) [1]. The founders of this concept are organizations from industry, science and governmental decision factors. AmI aims to prolong the time older people can live decently in their own homes by increasing their autonomy and self-confidence, by supporting their everyday activities and their disabilities, and to save resources. According to an ISTAG report (2004) [2], ambient intelligence can be seen as a starting point for AAL. AmI is mainly based on three technological fields: 1- ubiquitous computing, which consists of integrating microcomputers, sensors and actuators into everyday objects; 2 - ubiquitous communications, which enable objects to communicate with one another (ad hoc networking, self-organization, personalization, and context awareness play an important role); 3 - human-computer co-operation which means interaction with devices happens in a 'human' way. The main

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1083–1089, 2009.
© Springer-Verlag Berlin Heidelberg 2009

technologies are interactive and in some cases multimodal, and are based on speech, gesture, emotions, and artificial skins [3].

The fact that the decisions in AmI are based on profiles has raised the question whether they infringe existing non-discrimination legislation. Article 14 of the European Convention on Human Rights and Article 1 of Protocol No. 12 to the Convention prohibit the discrimination from the States towards individuals. This would mean, that "if profiling practices conducted by public bodies representing the state (i.e. the (local) government) would be proven to be discriminatory, these provisions could be invoked by an individual who could require from the government that the discriminatory profiling practice in that specific case is stopped ('vertical effect')". The use of specific criteria, such as sex, race, colour, language, religion, political or other opinion, national or social origin, association with a national minority, property, birth or other status, as described in Article 14 ECHR cannot be used as ground for discrimination. However the whole nature of AmI is based on a faddish way of discriminating among the users of a system and making decisions based on profiles that are built upon various criteria. Could it be sustained that AmI technologies are violating non-discriminating legislation as they are based on the processing of personal data and the consequent building of profiles? – asks Future of Identity in the Information Society (FIDIS) Network of Excellence [4].

2 Major Concerns about AMI's

First of all, the UN International Plan of Action on Ageing (Madrid 2002) states the development of supportive environments and Assistive Technology (AT) as a priority direction. Among the United Nations Principles for Older Persons, those of Independence ("older persons should be able to live in environments that are safe and adaptable to personal preferences and changing capacities"), Care ("older persons should have access to social and legal services to enhance their autonomy, protection and care"), and Dignity ("older persons should be treated fairly regardless of age, gender, racial or ethnic background, disability or other status …") are closely touching AmI and AAL developments.

Based on the basic pillars of the WHO concept on Active Ageing, it is recommended that the specific higher level needs of older people with regard to quality of life in developed countries should be classified into six groups:

- Health: the health needs of older people address three main aspects: non-existence of illness, a good functional status and an appropriate system of social support for the individual. In the upcoming years, it is expected that better informed patients will take on more responsibility for their own health and that they will be more proactive in pursuing their own illness or asking for a medical diagnosis and therapy or seeking AmI help.
- Safety: especially for the elderly and/or disabled people the need for personal safety is very important in everyday life and at home. Quite often it is focused on physical inviolability. The most common challenges are physical infirmity and the partial or total loss of mental abilities and cognitive performances. These restrictions lead older people to make the necessary adjustments and use safety products (e.g. detectors capable of noticing when a person falls down, like the sensor array on SHARE-It

developed platforms). In other cases, more serious restrictions lead to admission to an institution which means a very serious loss of individual autonomy. An AmI environment could provide solutions to all these problems, enhancing and improving the quality of the time one spends at home.

- Independence: usually, older people like to live an independent and autonomous life as long as possible. A central point is living independently at their homes. However, for older people who have an increasing number of serious handicaps it is much more difficult to continue independent living. AmI environments could extend the independence of one, the wheelchair-like or walker-like improve the walking range, thus permitting one to shop or make appointments and care for oneself for a longer period of time, even with some disabilities.
- Mobility: there is a strong need among older people for mobility. However, for individual reasons (e.g. health problems) this mobility can be constrained. A serious reduction of mobility can lead to enormous reductions in quality of life.
- Participation: is one of the central needs of the elderly. Contact and relations with other people have a positive influence on their well being and health. An increasing number of older people decide, for instance, to learn more about new topics and attend courses at universities and other educational institutions. Learning new e-skills for interacting with the environment or reach enhanced mobility could significantly maintain or improve their social insertion.
- Intimacy: this is one of the major concerns regarding AmI, one should not have the impression that is permanently watched by the system, and thus becoming a prisoner in his own home. The smart environment should interfere only in enhancing one's activities, or in the case of danger [3]. In Europe, PRIAM Project, an INRIA funded cooperative research action (ARC) addresses the privacy issues raised by ambient intelligence technology in a transversal and multidisciplinary way, favoring the exchange of ideas between lawyers and ICT experts. PRIAM position is that ubiquitous computing technology need not necessarily be an obstacle to privacy protection: if legal and social issues are given proper consideration, technology can also be used to allow individuals to exercise their rights. The key issue is to devise techniques able to support ambitious privacy protection policies while allowing for the flexibility required in the ambient intelligence context. Three main requirements are explored to reach this goal: formal specification of privacy policies, trust management, and auditability [5].

3 Classic Geriatric Health Providers Versus AMI'S

Interdisciplinary team care with high-functioning teams has been shown to have positive effects on the patients' health and lead to better clinical outcomes, higher patient satisfaction, and enhanced the delivery of care [6]. As the evidence on effective patient management changes, particularly for chronic disease or disabilities management, it is increasingly being shown that many patients benefit from the expertise provided when various disciplines work together [7,8,9]. The need for geriatric healthcare providers to work in teams or participate in primary care networks, and explore newer models of health care delivery (e.g. shared care) or relaying on an

Intelligent ambient to obtain data about the well being of their patients has become recognized as a priority in the health care system [10].

While inter professional education is thought to be a good idea [7] it can be difficult to implement it [11] and the cost of ambient intelligent devices, not yet proven on the field, might be too big for the health insurance companies. Attitudes and experiences with team care, regulatory requirements for the various professions, legal responsibility for care, variability in faculty support, and different levels of training can all work in setting up inter professional care teams. Creating an educational culture conducive to promoting inter professional care is a hurdle that has yet to be successfully negotiated [10, 12, 13]. We know that successful interdisciplinary care requires team members to work collaboratively; to have clear, measurable goals; to be supported by clinical and administrative systems; to have a clear division of labor; and to have received training in interdisciplinary care [6, 9].

Concerning the older people of tomorrow, some authors see that the exposure to computer applications will lead to a growing openness and increasing competency with respect to using technological innovations. The next generation of intelligent and semi-autonomous assistive devices for older persons and people with disabilities (both cognitive and/or motor) should provide support so that they can be self-dependent enough to autonomously live in the community, staying at home as long as possible with a maximum safety and comfort. However, they do not expect that all older people will be able to take advantage of these technological advances ("digital divide") since their experiences with future technology, their educational backgrounds and their incomes may vary significantly [3].

Disability is defined as “difficulty or dependency in carrying out activities necessary for independent living, including roles, tasks needed for self-care and household chores, and other activities important for a person’s quality of life” [14]. As defined within the ICF [15], disability depends on health conditions but also on environment factors and personal factors and at the end disability results in activity limitation and in restriction in participation. Nevertheless, the elderly are the population group for which defining disability and functional impairment is most difficult. The typical patient benefiting from intelligent ambient devices is an elderly patient, with comorbid conditions and diseases, cognitive and/or physical impairment, functional loss from multiple disabilities, leading to impaired self-dependency. AmI platforms may offer a range of services to the target population.

4 US Law on Geriatric Health Care

70 million by 2030 - that's one of the projections regarding the number of Americans over the age of 65 who will be living in communities across the United States. This unprecedented demographic shift will place additional pressure on healthcare providers, already in short supply. Important legislation has been recently introduced in the Congress regarding loan repayment for healthcare professionals who train in the field of geriatrics. This critical legislation will attract and retain trained healthcare professionals and direct care workers by providing loan forgiveness and career advancement opportunities in a variety of disciplines. The bill will also establish a health and long-term care workforce advisory panel for an aging America. Specifically, the bill establishes the

Geriatric and Gerontology Loan Repayment Program for physicians, physician assistants, advance practice nurses, psychologists and social workers with award payments of up to $35,000 a year during the first two years of practice. Currently there are approximately 7,000 certified geriatricians; however, the Alliance for Aging research estimated that 14,000 geriatricians are currently needed and by 2030, they estimate the need for 36,000 geriatricians. There is an urgent need to prepare the entire healthcare workforce, including physicians, nurses, social workers and psychologists, to better confront the increasing numbers of older adults and this legislation can play a major role in addressing this issue. On these issues the ambient intelligence technologies will provide some relief on an already overcrowded system, giving health care providers an actual picture of the needs of the elderly in their charge [16].

5 European Geriatric Care and the Integration of New EU Members in the System

In the developed countries life expectancies are higher, families are smaller, and the “young” are sometimes middle-aged. The elderly are mostly isolated from the rest of the families, the majority of them living alone. The system of nursing care is a last stage for the elderly. Usually it is owned and operated by the government, by the local municipality, by the church or by private commercial suppliers [17]. The part of the elderly population witch needs full time care is considered as equivalent to the percentage of severely disabled elderly, which in turn is estimated to be 5% for the 65-69 year-old age group, 10% for the 70-79 age group, and 30% for the 80 and over age group[16]; this population can sum up to ten million people in the EU 25 area [18]. The care of chronic and disabled patients involves lifelong treatment under continuous expert supervision.

Healthcare professionals and patients agree that institutionalization in hospitals or residential facilities may be unnecessary and even counterproductive. Home Care has been considered as a fundamental component of a network of long term care facilities, capable of reducing expenses, institutionalization and risk of death. An effective Home Care has direct social implication of helping partially or completely dependent people to live in their home and social environment for as long as possible, and to contrast the improper use of institutionalization. AmI is envisaged as a major component of home care.

However, AmI platforms are often complicated systems. As their constituent parts may be provided from different manufacturers, they require to be made compatible with each other. This is supported by standardization, which also serves other purposes: supports accessibility and usability requirements, product safety, etc. Standardisation may reduce the cost of Ambient Intelligence system by reducing the number of similar systems [19]. In this respect, important actions are going on in ISO (International Standardization Organisation). ISO and IEC have together produced the *ISO/IEC Guide 71:2001: Guidelines for standards developers to address the needs of older persons and persons with disabilities.* The ISO/IEC Joint Technical Committee One (JTC1) issued ISO/IEC JTC 1/SC 35; User Interfaces and ISO/IEC JTC1 SC36; Information Technology for Learning, Education, and Training [19]. Other activities are taking place in ITU (International Telecommunication Union) and also in individual EU countries.

European legislation includes many fields which are important when looking at Ambient Intelligence from a human point of view. An obvious area is data protection**.** It is strongly both an ethical and legal issue. The new Directive 2002/58/EC on the processing of personal data and the protection of privacy in the electronic communications sector includes provisions on security of networks and services, confidentiality of communications, access to information stored on terminal equipment, processing of traffic and location data, calling line identification, public subscriber directories and unsolicited commercial communications. This directive is currently being revised. Another important area of legislation is consumer protection and product safety**.** According to European legislation, a product is presumed safe once it conforms to specific Community provisions, national regulations or certain principles. The general safety requirement is imposed by the Directive 2001/95/EEC on general product safety on any product put on the market for consumers or likely to be used by them and not covered by sectorial legislation, including all products that provide a service. Since Ambient Intelligence system must adapt different needs of individual users, it is inevitable to use personal data of users in the system.

In addition to the above issues, consumers are also concerned with Electro Magnetic Compatibility issues that can affect compatibility. To comply with the EU directives individual components have the “CE” mark to show presumption of compliance. Consumers will be re-assured if they know that EMC could also be guaranteed for the whole AmI system, not just for the component parts. In addition to EMC, it must be guaranteed that an Ambient Intelligence system does not physically harm the user who may even “wear” some part of the system. Furthermore, Low Voltage Directive 2006/95/EC is important for domestic appliances. Also environment should be taken into account in order to keep with sustainable development [19].

Telecommunication provides an important infrastructure for Ambient Intelligence. Central to the legal framework are the telecommunications directives, especially the Directive on radio equipment, telecommunications terminal equipment and the mutual recognition of their conformity (RTTE) 1999/5/EC. Directive 2002/22/EC on universal service and user's rights to electronic communications networks and services (Universal Service Directive) shows an important user perspective. However, present and emerging challenges require a lot of work in all fields.

The countries that recently joined the European Union have different systems, with other characteristics: 1) the professional system of geriatric care is behind the times, 2) the elderly represent the poorest segment of society, the pension is usually insufficient for a normal way of live, they need help from the family, 3) a system for geriatric nursing care covering the whole country practically does not exist, 4) the home-care system is usually insufficient, 5) geriatric patients, who need only care instead of treatment, are often hospitalized, without medical reasons, 6) there is a lack of budgeting, from the state to the local level, 7) underpaid staff without enthusiasm.

References

1. Ambient Assisted Living – Preparation for an Article 169 Initiative, `http://www.aal169.org/members.html`
2. ISTAG report on Grand Challenges for IST (2004), `ftp://ftp.cordis.europa.eu/ub/ist/docs/2004_grant_challenges_web_en.pdf`

3. Malanowski, N., et al.: Active Ageing and Independent Living Services: The Role of Information and Communication Technology (2008)
4. Analysis of the legal framework, Future of Identity in the Information Society (FIDIS) Network of Excellence, `http://www.fidis.net/resources/deliverables/profiling/d79-a-vision-of-ambient-law/doc/21`
5. PRIAM: Privacy Issues and AMbient intelligence, `http://www.inria.fr/recherche/arc/index.fr.html`
6. Grumbach, K., Bodenheimer, T.: Can health care teams improve primary care practice? JAMA 291(10), 1246–1251 (2004)
7. McDonagh, T.A.: Lessons from the management of chronic heart failure. Heart 91, ii24 (2005)
8. Preen, D.B., Bailey, B.E., Wright, A., Kendall, P., Phillips, M., Hung, J., et al.: Effects of a multidisciplinary, post-discharge continuance of care intervention on quality of life, discharge satisfaction, and hospital length of stay. A randomized controlled trial. Int. J. Qual. Health Care 17(1), 43–51 (2005)
9. Headrick, L.A., Wilcock, P.M., Batalden, P.B.: Interprofessional working and continuing medical education. BMJ 316(7133), 771–774 (1998)
10. College of Family Physicians of Canada. Standards for accreditation of residency training programs: family medicine; emergency medicine; enhanced skills; palliative medicine (the Red Book). College of Family Physicians of Canada, Mississauga, ON (2003), `http://www.cfpc.ca/English/cfpc/education/accreditation/default.asp?s=1`
11. Reuben, D.B., Levy-Storms, L., Yee, M.N., Lee, M., Cole, K., Waite, M., et al.: Disciplinary split: a threat to geriatrics interdisciplinary team training. J. Am. Geriatr. Soc. 52, 1000–1006 (2004)
12. Leipzig, R.M., Hyer, K., Ek, K., Wallenstein, S., Vezina, M.L., Fairchild, S., et al.: Attitudes toward working on interdisciplinary healthcare teams: a comparison by discipline. J. Am. Geriatr. Soc. 50, 1141–1148 (2002)
13. Williams, B.C., Remington, T., Foulk, M.: Teaching interdisciplinary geriatric care [abstract]. Acad. Med. 77(9), 935 (2002)
14. Fried, L.P., Ferrucci, L., Darer, J., Williamson, J.D., Anderson, G.: Untangling the concepts of disability, frailty and comorbidity: implications for improved targeting and care. J. Gerontol. Biol. Sci. Med. Sci. 59, 255–263 (2004)
15. World Heath Organization, `http://www.who.int/classifications/icf/en/`
16. The American Geriatric Society, `http://www.americangeriatrics.org`
17. Rurik, I.: Geriatric care at the EU level? How to join? Z. Gerontol. Geriat. 38, 60 (2005)
18. Walker, A., Maltby, T.: Ageing Europe. Open University Press, Buckingham, Philadelphia (1997)
19. Kemppainen, E., Bringsrud, K.A., Engelen, J., Giovannini, C., Shipley, T., Yamada, H.: Cost 219ter. Standardisation and Legislation with Regard to Ambient Intelligence and Accessibility (2008), `http://www.tiresias.org/cost219ter/amileg.htm`
20. European Comission-Eurostat, `http://epp.eurostat.cec.eu.int/portal/`

Construction and Debugging of a Multi-Agent Based Simulation to Study Ambient Intelligence Applications*

Emilio Serrano, Juan A. Botia, and Jose M. Cadenas

University of Murcia, Murcia, Spain
emilioserra@um.es, juanbot@um.es, jcadenas@um.es

Abstract. This paper introduces the development of an infrastructure to study highly complex applications of Ambient Intelligence (AmI) which involve a large number of users. The key ideas about the development of a multi-agent based simulation (MABS) for such purposes, Ubik, are given. The paper also extrapolates effective technologies in the development of multi-agent systems (MAS) to the field of MABS. In particular, the basis for the use of forensic analysis as a method to assist the analysis, understanding and debugging of Ubik in particular and the general MABS are set up.

1 Introduction

"Ambient Intelligence" (AmI) stems from the convergence of ubiquitous computing, ubiquitous communication and intelligent user-friendly interfaces and creates environments that are characterized by their ubiquity, transparency and intelligence [6]. AmI is of great interest to society because in a near future it can help us to do our work, improve our health, do house chores, etc. The research on AmI presents new scientific challenges. Traditional usability engineering methods and tools fail in the development of AmI applications [3]. Established organizations such as the Usability Professional's Organization have recognized the need for new approaches to usability testing [3]. When AmI is applied to a large number of users, there is a point where the real tests are not feasible. This paper tries to provide solutions to these situations.

Multi-agent based simulation, MABS, allow modelers to handle different levels of representation (e.g., "individuals" and "groups", for instance) within an unified conceptual framework. Such versatility makes MABS one of the most favorite and interesting support for the simulation of complex systems [4]. MABS is used in more and more scientific domains [4]: sociology, biology, physics, chemistry,

---

* This research work is supported by the Spanish Ministry of Education and Science in the scope of the Research Project TIN-2005-08501-C03-02 and by the Project "Análisis, Estudio y Desarrollo de Sistemas Inteligentes y Servicios Telemáticos" through the Fundación Séneca within the Program "Generación del Conocimiento Científico de Excelencia".

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1090–1097, 2009.
© Springer-Verlag Berlin Heidelberg 2009

ecology, economy, etc. This paper proposes the use of MABS in AmI. Specifically, complex AmI applications with a large numbers of users are treated. The testing of the social behavior of user groups is interesting in these applications. That is, we are interested in the macro-social perspective. Therefore, the overall objective of our research is to increase the usability of this type of complex an AmI applications.

About the field of MABS, generally, researchers are more interested in model design than in model execution and the execution analysis. Specifically, the analysis is viewed as less methodical and hand crafted [4]. The field of Multi Agent Systems (MAS), a well-established branch of AI, is complementary in several aspects to MABS [2]. The MAS theory has scientific approaches to analyze, understand and debug the social level of agents. This paper aims to introduce that technologies used in the analysis of MAS can be used for MABS. In particular, *forensic analysis* is proposed. This technology has already been used successfully to debug MAS in a social level [13]. Forensic analysis is the process of understanding, re-creating, and analyzing arbitrary events that have occurred previously [11].

To achieve the overall objective, increase of the usability of complex AmI applications, four objectives are marked: (1) Developing of an infrastructure to study complex AmI applications. The multi-agent based simulation Ubik achieves this goal. (2) Developing and testing of the effectiveness of AmI applications with the infrastructure resulting in the first goal. This is achieved by extending the Ubik simulation. (3) Developing of an infrastructure to assist the analysis, understanding and debugging of group behaviors in general MABS through forensic analysis. (4) Use of the results of the third goal for the simulations resulting in the first two objectives. What we aim is to validate the simulated models, understand them, explain them and, ultimately, achieve the overall objective.

The next section deals with the related work. Then section 3 studies in detail the infrastructure for complex AmI applications and the sort of AmI applications which can be studied. The section 4 treats the infrastructure for the debugging of MABS. Finally the conclusions and future work are given.

2 Related Works

A *Living Lab* is a laboratory which consists of real life environments where real users together with researchers look together for new services in AmI. In 2000 started first serious plans to build a laboratory that could be used to conduct feasibility and usability studies in AmI [6]. Finally, the HomeLab of Philips was opened on April 24, 2002. Other well-known living labs are AwareHome Project [7] or Adaptative House [10]. This approach of direct testing in laboratories has proliferated. For example, the *European Network of Living Labs* is a grown up initiative coming from the own European Living Lab and sponsored by the European Community. On the open living labs web[1] can be found contact information

---

[1] Open living labs website: http://www.openlivinglabs.eu/

for living labs in 29 countries, only in Spain 19 living labs are joined with this project. However, these approaches are not feasible in applications for thousands of users or more. Research into ubiquitous computing has also used simulations such as TATUS [5] and Ubiwise [1] which extend first person shooter games (FPS). However, these simulations, which model the physical environment in a 3D view, do not model the user which is a real player of the game. Besides, they are valid for only a few users and not for thousands of them. Hence, this paper proposes the use of MABS.

Luke et al. [9] introduce the basis of the platform we use to program the simulation Ubik, MASON[2], as well as several simulations using it. For example, the simulation *Anthrax Propagation in the Human Body* has in common with Ubik that treats a field with which you can not experiment with in a laboratory. This simulation, like ours, is useful to help researchers in the field to explore what-if scenarios and to create strategies to attend humans effectively. The MABS have also been used to check the effectiveness of networking applications as we intend. In this way, *Network Intrusion* is an agent-based model designed to study computer network security [9]. The parameters of the model allow researchers to understand the effects of changes in security policies, just as Ubik does with the AmI applications policies. In general, any work on MABS offers interesting points of our research, the innovation of this paper is the use for AmI applications.

Simulation typically generates huge amounts of data, analysis of the simulations have not received the attention it deserves. In the MAS theory, we can find literature that deals with the analysis of the agents. In this way, Serrano et al. [13] detail the creation of an infrastructure for forensic analysis for MAS. Forensic analysis is often the basis of most proposals about debugging in MAS and it is often accompanied by comprehensible data representations. There are works that analyze the behavior of the agents groups with *petri nets*, *AUML diagrams*, extensions of the *propositional dynamic logic*, *statecharts*, *dooley graphs*, and so on. There are also works that combine forensic analysis and data mining to achieve simple representations of the agent society [14]. Each of these methods to analyze and debug the MAS are directly usable in this field of MABS.

3 Ubik, a Multi-Agent Based Simulation for the AmI Domain

Any AmI facility involves a financial outlay, much larger when AmI is applied in large domains as a skyscraper or even a whole city. For example, we do not know the cost of the HomeLab of Philips [3], a laboratory to study AmI applications, but we are quite sure that is much greater than that needed for the development of any MABS. The analysis of simulations helps to verify the effectiveness of AmI applications before install them. It also permit to check the results of the AmI in diverse and configurable societies, which make these applications more

---

[2] MASON website: http://cs.gmu.edu/ eclab/projects/mason/

robust. Anyway, the most important reason for the presented proposal is to be used in AmI applications which so many users that the real tests are totally unworkable. For example, HomeLab consist of a house with a living, a kitchen, two bedrooms and a study where such research cannot be performed.

Ubik[3] is an infrastructure to study complex AmI applications which involve a large number of users. It is a MABS programmed in MASON and aims to be as descriptive as possible to be useful for AmI. A longer description implies a more complex process of understanding things of interests for us to explore new technologies to assist the analysis of simulations (section 4). Figure 1 shows Ubik between MASON and AmI applications.

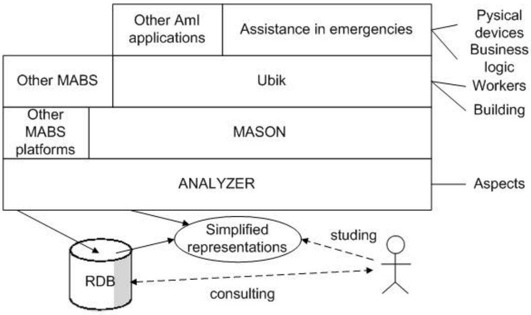


Fig. 1. Proposal of the paper

With regard to the modeled space in Ubik, it is treated as an environment compounded by buildings devoted to offices where the main activities are related to working tasks. Regarding the modeled agents, the simulation not model the human with a cognitive point of view, but reactive. We are interested in human behavior as a member of a group of other human which, predictably, will have same behaviors to events of interest. The simulated space and agents must be considered to select valid AmI applications to be studied in Ubik. For example, a scenario in which the human has a reactive behavior is in an emergency. Figure 1 shows this concrete AmI application over Ubik.

The simulated building has a number of floors, stairs, elevators, hallways, rooms (bathrooms, kitchens, resources...), etc. A key idea is that the whole building is configurable, so the user can simulate endless scenarios. It is considered that any area without any specific purpose, such as a hallway or room, is a valid work space where an agent can place a table and a work. More complex structures such as offices or cubicles would require more complex and laborious specifications to set the building, for example, expressed with ontologies. Figure 2 shows the floor of a building simulated by Ubik.

In Ubik, each worker is an agent. The workers have behaviors: staying on the work place, visiting another worker, downing to the street, walking around the

---

[3] Code, photos and videos of Ubik on: http://ants.dif.um.es/staff/emilioserra/Ubik/

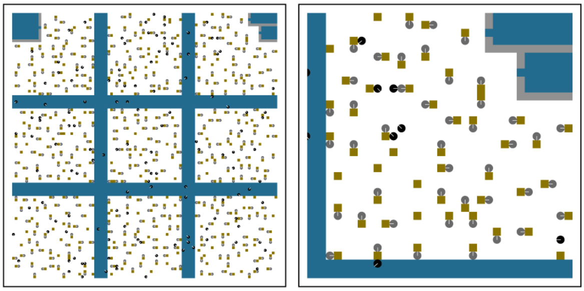


Fig. 2. On the left, a vectorial image of floor in Ubik with 500 agents. On the right, the zoom in the upper right corner of the floor.

office, convening meetings, attending meetings, attending emergency, going to certain rooms and so on. On the other hand, there are different types of workers: managers, subordinates, secretaries, service personnel, and so on. There is a fitness, linked to every worker, which determines the movement velocity. This velocity is lower if the agent is using the stairs. All these factors are configurable to simulate a multitude of societies in Ubik. Figure 2 represents the workers as an oval, more clearly whether they are on the work place, and a line indicates where they are watching.

To study AmI applications in Ubik, first the simulation is extended with the concepts of the application such as physical devices (i.e. displays, computers, domotic devices, etc.). Then, the business logic is modeled as AmI application (see figure 1). The effect is that the applications and devices have influence on the simulated space and, indirectly, in the agents. For example, an application of AmI in which the space and the agents are compatible with the Ubik simulation is the assistance in emergencies. With regard to the building, it is required to add sensors that detect emergencies, for example fire. Lights panel are also required to show the emergency situation and to guide workers toward the nearest stairs / elevator. With regard to the agents, new features which affect to the application must be considered. For example, people using wheelchairs move with a normal speed through the office but are unable to use the stairs. The simulation also must consider the concept of configurable emergencies. This is useful to simulate lots of scenarios in order to make the application more robust.

An important factor in Ubik, for its purpose, is the ability to operate with a large number of agents as well as a large space for these agents. The tallest completed building, Taipei 101, has 101 floors and a total construction area of 420,000 square meters. One of the main goals of the simulation platform MASON are supporting a large number of agents and their speed [12]. Ubik, developed in MASON, has supports studies with simulations of more than 200,000 agents and 200 floors efficiently. How could be made a real test of an AmI application for 200,000 people? We don not know. What is the best way to evacuate 200,000 people from a building? We hope that MABS in general and Ubik in particular get closer to the answer. The following section is a proposal for analysis, understanding and debugging of Ubik, or any other simulation, through forensic analysis.

4 Debugging Social Simulations from a Forensic Perspective

As noted above, the analysis phase of MABS has been traditionally performed in an exploratory and intuitive manner [4]. One of the weaknesses of MASON, and MABS platforms in general, is that they offer few facilities to monitor and debug the simulated models [12]. MASON offers the possibility of fix inspectors in individual properties (of the model or of the agents) to be monitored / recorded / modified from the simulation. However, we miss many options as views and records of the artificial society as a whole.

Forensic analysis is the process of understanding, re-creating, and analyzing arbitrary events that have previously occurred [11]. Serrano et al. [13] detail the creation of an infrastructure for forensic analysis in MAS. That technology can be brought to MABS. The key idea is to capture interesting elements of a simulation and store them in a relational database (RDB). Once the RDB has captured the essence of a simulation, it is simple to make simplified representations of the stored data. These representations assist the analysis process. The field of MAS has already work with representations for these porpoises. For example, showing summaries of cooperation cores or similarities in the societies of agents [14]. The figure 1 shows the analyzer that records data in a RDB. It also shows a developer consulting that RDB and studying certain representations to analyze MABS.

Regarding how to capture elements of the simulation, it could be thought that it is something trivial. Calls to RDB can be simply programmed into the model or platform. This presents a major problem, this code would serve only for the specific model or the specific version of the platform. In addition, the registration code would be dispersed throughout the model or platform code. The solution in this proposal is the use of *Aspect oriented programming (AOP)* [8]. AOP can isolate the aspect of register a MABS execution in order to treat it separately and in a modular manner. For now, our work is focused on the MASON platform. However, all the programmed aspects have equivalence in other platforms. Hence, replicating the results for other platforms is easy if they offers the source code as MASON does. Figure 1 shows how the analyzer can be used on different platforms. In this way we hope that our work is useful for the entire MABS community.

Regarding what elements are interesting to capture in a simulation, in a first approximation, it can be said that all those elements which were chosen to be monitored in the simulation. Here, the clarity of the platform code is a major factor. MASON is extremely clear. Any method that starts with "get" in the code of an agent (implements the *Steppable* interface) or a model (implements *SimState*) returns a property that is displayed when this agent or model respectively is being inspecting. Programming aspects which intercept all the calls to these methods is a trivial task. The body of the aspect store these properties in a RDB. Note that a complicated or confusing core of the platform would complicate extremely this stage. That is why MASON was chosen instead Repast [12]. Once this basic approach has been covered, interesting elements present in all the simulations can be added to be registered. For example, the moment of creation of agents, their destruction or the interaction between them. The data in RDB support simplified representations to be even more useful. In any case, the mere fact of having a RDB, which is a powerful tool for Querying, is a breakthrough that allows to do exploratory data mining.

5 Conclusions and Future Work

This paper set the basis of an infrastructure to study applications of ambient intelligence (AmI) which involve a large number of users. The infrastructure is a multi-agent based simulation (MABS) called Ubik. The paper also introduces an infrastructure to assist the analysis, understanding and debugging of a MABS extrapolated of multi-agent systems (MAS). The infrastructure consists of forensic analysis by aspect oriented programming (AOP).

The introduction to the paper has introduced how new techniques are needed to test AmI applications and how the MABS, a versatile technology which is at its very peak, can be used for that purpose. It has also pointed that MABS have opted for an analysis in an intuitive manner while the field of MAS has several proposals for such purposes. Related work has explained the classic approaches to test AmI applications, how MABS also serve for these purposes and the most interesting theories to analyze MAS. Section 3 has presented Ubik as a MABS to test AmI applications and the details and restrictions on those AmI applications. It also has given the example of a concrete application of high social interest and which can be tested using Ubik, emergency assistance in an office building. Section 4 has presented a proposal for analysis, understanding and debugging MABS. This proposal is based on forensic analysis and AOP. It has also shown that this infrastructure provides the MASON platform with a powerful analytical capacity.

Regarding future works, the first one is to extend Ubik to make it more descriptive. For example, with detailed configurations of space and agents expressed with ontologies. We also intend to include several AmI applications of social interest as emergency assistance has been included. The infrastructure to assist the analysis of MASON simulations also has promising future works. One of the first is a survey about representations to simplify the data of the forensic analysis. Specifically, those representations that were successful in the development of MAS and may be exploitable in MABS. In general, we are interested in integrating our work with any technology that increases the usability of AmI applications or assists the analysis of MABS.

References

1. Barton, J.J., Vijayaraghavan, V.: Ubiwise: A ubiquitous wireless infrastructure simulation environment, tech. report hpl2002-303. HP Labs (2002)
2. Conte, R., Gilbert, N., Sichman, J.S.: Mas and social simulation: A suitable sommitment. In: Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation, pp. 1–9. Springer, London, UK (1998)

3. de Ruyter, B., Aarts, E.: Ambient intelligence: visualizing the future. In: AVI 2004: Proceedings of the working conference on Advanced visual interfaces, pp. 203–208. ACM Press, New York (2004)
4. Drogoul, A., Vanbergue, D., Meurisse, T.: Multi-agent Based Simulation: Where Are the Agents? In: Sichman, J.S., Bousquet, F., Davidsson, P. (eds.) MABS 2002. LNCS, vol. 2581, pp. 1–15. Springer, Heidelberg (2003)
5. Lewis, D., O'Donnell, T., O'Sullivan, D., O'Neill, E., Klepal, M., Pesch, D.: A testbed for evaluating human interaction with ubiquitous computing environments. In: TRIDENTCOM 2005: Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities, pp. 60–69. IEEE Computer Society, Los Alamitos (2005)
6. Friedewald, M., Da Costa, O., Punie, Y., Alahuhta, P., Heinonen, S.: Perspectives of ambient intelligence in the home environment. Telemat. Inf. 22(3), 221–238 (2005)
7. Kidd, C.D., Orr, R., Abowd, G.D., Atkeson, C.G., Essa, I.A., MacIntyre, B., Mynatt, E., Starner, T.E., Newstetter, W., et al.: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. Lecture Notes in Computer Science, pp. 191–198 (1999)
8. Laddad, R.: AspectJ in Action: Practical Aspect-Oriented Programming. Manning Publications Co., Greenwich (2003)
9. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K.: Mason: A new multi-agent simulation toolkit. In: Proceedings of the 2004 Swarmfest Workshop (2004)
10. Mozer, M.C.: The neural network house: An environment that adapts to its inhabitants. In: Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, pp. 110–114 (1998)
11. Peisert, S.P.: A model of forensic analysis using goal-oriented logging. PhD thesis, La Jolla, CA, USA, Adviser-Sidney Karin (2007)
12. Railsback, S.F., Lytinen, S.L., Jackson, S.K.: Agent-based simulation platforms: Review and development recommendations. Simulation 82(9), 609–623 (2006)
13. Serrano, E., Botia, J.A.: Infrastructure for forensic analysis of multi-agent systems. In: Proceedings of the Programming Multi-Agent Systems Workshop at AAMAS 2008, Estoril, Portugal (2008)
14. Serrano, E., Gomez-Sanz, J.J., Botia, J.A., Pavon, J.: Intelligent data analysis applied to debug complex software systems. Neurocomputing (to appear)

Easing the Smart Home: Translating Human Hierarchies to Intelligent Environments

Manuel García–Herranz, Pablo A. Haya, and Xavier Alamán

AmILab, Ambient Intelligence Laboratory, E.P.S Universidad Autónoma of Madrid
C/ Francisco Tomás y Valiente 11,28049 Madrid, Spain
`{manuel.garciaherranz,pablo.haya,xavier.alaman}@uam.es`
`http://amilab.ii.uam.es`

Abstract. Ubiquitous computing research have extended traditional environments in the so–called Intelligent Environments. All of them use their capabilities for pursuing their inhabitants's satisfaction, but the ways of getting it are most of the times unclear and frequently unshared among different users. This last problem becomes patent in shared environments in which users with different preferences live together. This article presents a solution translating human hierarchies to the Ubicomp domain, in a continuing effort for leveraging the control capabilities of the inhabitants in their on–growing capable environments. This mechanism, as a natural ubicomp extension of the coordination mechanism used daily by humans, has been implemented over a real environment: a living room equipped with ambient intelligence capabilities, and installed in two more: an intelligent classroom and an intelligent secure room.

Keywords: Ubiquitous Computing, Human-centered computing, Rule-based processing, Command and control.

1 Motivation

Once environments have been supplied with perceiving and actuating capabilities, lots of research efforts have been put into creating "intelligent environments" to actuate on the user's behalf. In this sense, two main trends of research can be identified: *autonomous environments* and *automatic environments*. The former represent those systems that, without intervention of the user, try to reach the goals they are intended for (such as reducing energy consumption [1] or minimizing the number of tasks the user has to do [2]). Many of these systems are based on "black box technologies", meaning that they are not human–readable, such as neural networks, Hidden Markov models or Bayesian networks. Even though these last ones may have an interpretable graphical representation being, they are not strictly readable. Automatic systems, on the other hand, represent those systems trying to replicate the solutions that have been explicitly given by the user. These systems are normally based on "white box technologies" such as rule–based [3] or case–based [4] expert systems, since, as stated by Myers, *the closer the language is to the programmer's original plan, the easier the refinement process will be* [5]. Closer or farther from the end–user mental plans, those

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1098–1105, 2009.
© Springer-Verlag Berlin Heidelberg 2009

systems pretend to be programmable and, to that extent, they provide means for creating, structuring and organizing "code". In this sense, Myers points at rule-based languages as the ones naturally used by users in solving problems [5]. Summarizing, while autonomous environments try to "replace" the user, automatic environments try to extend their control.

While both approaches have advantages and disadvantages, they share the problem of conflicting inputs due to more than one inhabitant i.e. collisions. While autonomous solutions have a hard time to learn from a "noisy" environment, automatic approaches need some clarifying input for what to do in conflict situations.

Our choice has been to leverage the user's control over the environment through an automatic approach, focusing on personal environments and trying to **get as close as possible to the end–user**. While a more detailed description of our overall choice can be found in [6], this paper focuses on its capabilities for dealing with social conflicts.

2 Indirect Control in Intelligent Environments

The purpose of the indirect control mechanism is to allow users to "program" behaviors/preferences/implicit interaction on their environments. For doing so, we use a rule–based agent mechanism that will be briefly explained to provide the basis for the rest of the article.

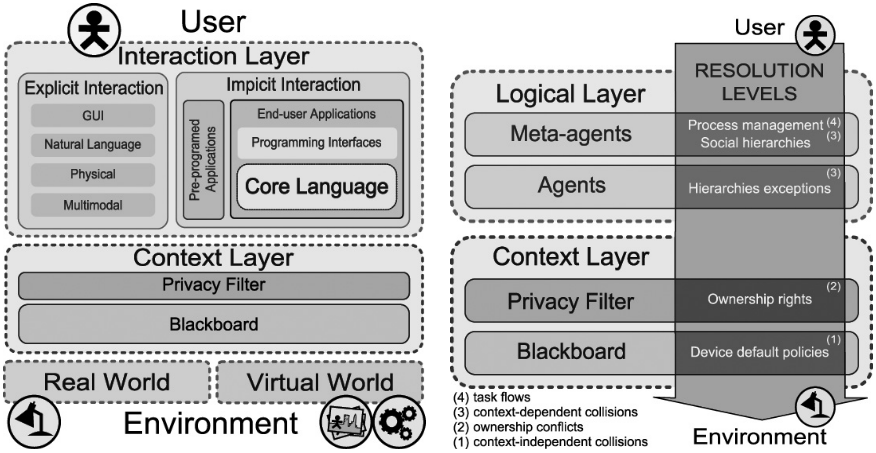
2.1 A Rule–Based Multi–Agent Programming System

As in every intelligent environment, the first step is always that of perception (see Figure 1(a)), done in our approach through a middleware layer ("the Blackboard") in which every element of the environment, either real or virtual, is represented as an *entity* with *properties* and *relations* between them [7]. Every change on the environment is reflected in the blackboard and vice versa.

Over this context layer there is an interaction layer in which all the applications are located, the indirect control mechanism among them [6]. This mechanism is designed to allow users to program their environment and is based on a rule–based core language. Rules can be created and organized into agents (the minimal self–sufficient structure). Agents are represented in the blackboard as agent entities with a *status* and *task* properties and the *is_owner*, *located_at* and *affects* relations, linking it with the user that create it, the location it acts in and the elements it affects, respectively. Following Paperts' ideal of "low–threshold no ceiling" [8] the rule–based language is based in a basic language with some expression extensions. The basic language has three parts: *triggers*, *conditions* and *actions*, in analogy with the natural *"When ... if ... then ...* structure.

3 Human Hierarchies

Through this system, the diversity of preferences and tasks coexisting in the environment, is reflected in a multitude of context–aware applications running



(a) From the structure point of view (b) From the conflict resolution point of view

Fig. 1. AmiLab layers. The World layer (i.e. what is in the environment), the Context layer (i.e. how ubicomp acces it: abstraction [7] and privileges [9]) and the Interaction layer (i.e. how can be used: Explicitly and Implicitly). This article focus on Implicit interaction programmed by the user.

over the same environment. In traditional environments, these preferences and tasks are prioritized according to some hierarchies accepted by the environment inhabitants. Analogously, since context–aware applications are automated user preferences, they must have a mechanism through which to represent and apply these same hierarchies if they are to coexist in the same environment.

Hierarchies are the natural social structures to establishing an order of preference but, linked to the social group that created them, they are multiple an dynamic and their complexity reflects the complexity of the social group they rule. In addition, every social group has its own ways for generating and accepting their hierarchies.

This work focus on social groups as the only generators/acceptors of their hierarchies, looking forward to enrich the control users' experience over their environments, rather than proposing a fixed social engineering solution.

In this sense, our indirect control mechanism is just an extension of the user's control. Decisions are automatic in the sense that they do not need direct human intervention but they were created by the end–user and should be governed by the same hierarchies governing users' daily life.

Given that every action in the environment is –at least indirectly– produced by a human and our believe that each social group generates/accepts its own hierarchies, this work is based on the principle that **neither the system automatically nor a third human party such as an engineer should specify the hierarchies** governing a social group, but they should allow instead particular social groups to express their own. This apparently naive principle presents a profound problem when designing a solution since any fixed structure will interfere with the possibly different natural structure of the user and, secondly, the flexibility of the system must not interfere with the simplicity of end–user oriented solutions. Both structure and flexibility must adapt to the ways humans use to create and organize their hierarchies. In this direction we will analyze two social factors: **purpose** and **complexity**.

3.1 Purpose and Complexity

As each type of conflict produces a different hierarchy domain, we find that the *purpose* for what hierarchies are build is varied and entangled. Social hierarchies (i.e. *who goes first?*) and task hierarchies (i.e. *What goes first?*) are just two examples that can be in conflict: What happens if there is a conflict between a high–priority person doing a low–priority task and a low–priority person doing a high–priority task? A new hierarchy (or an exception) will be born: A hierarchy of hierarchies, so to speak. Some systems have attacked this problem through classifying, organizing and structuring. Kakas et al. [10], for example, provide each agent of a multi–agent system with two types of priority rules: *Role priorities* and *Context priorities.* Role rules prioritize according to the *role* of the parts involved (some sort of social hierarchy), while context rules prioritize role rules according to some extra context (some sort of task hierarchy). Additionally, each agent is supplied with a *motivation* structure, a hierarchy on the goals it pursues. This goals are defined according to Maslow's need's categorization [11]. Finally,an additional hierarchy is specified to define the agent's "personality" (i.e. his decision policy on needs to accomplish the goals of its motivation). While this kind of structuring captures many of the flavors of human behavior, it presents some problems when applied to personal environments. First, it needs a professional (familiar with Maslow's theory, to begin) to program the system, and a clear a priori classification of the family goals, roles and tasks. This engineering solutions, while perfectly valid for domains such as business automation, should not be applied to home environments in which a programming third party, a too rigid categorization or the need of a deep a priori definition breaks some of Davidoff's principles for smart home control, such as "allow an organic evolution", "easily construct/modify behaviors", "Understand improvisation" or "participate in the construction of family identity" [12]. Thus, hierarchies should no be fixed to specific domains (e.g. roles or goals) neither presume the knowledge of complex concepts (e.g. Maslow theory).

Secondly, as a social structure, we considered *complexity* the second factor of interest while analyzing hierarchies. How do we allow the different degrees of complexity of different social groups? How can we measure those different degrees of complexity? Y. Bar–Yam [13] *interdependence* and *scale* concepts to measure complexity are quite useful in this. Interdependence describes the effects a part has over the rest of the system: "If we take one part of the system away, how will this part be affected, and how will the others be affected?". Scale, on the other hand, refers to the different degrees of complexity a system acquires depending on how close or far removed apart is the observer. These concepts characterize complex systems: both a microprocessor (a designed system) and the global economy (an spontaneous system) have millions of components but, while the former is easier to understand (the interdependence is clear), the latter is not fully understood nor controlled by anybody (unclear interdependence). On the other hand, the former's growth depends on a new design while the latter is robust and adaptive to changes in its subcomponents, to name some of the consequences of their complexity's idiosyncrasy.

Social networks range from strongly planned, as in military hierarchies, to highly spontaneous, as in the Internet, depending on the goal of the structure. Some have their complexity spread among the structure (e.g. the Ford T production chain) while others condense it in specific parts (e.g. a diamond cutting business). The different degrees of interdependence and scale a solution provides is easy to see and clearly shows what kind of complexities can be achieved with it.

Finally, from the necessity of creating flexible structures and hierarchies, a decentralized mechanism for hierarchy management is deduced. A mechanism of this kind, in which there is not necessarily a central agent coordinating the processes does not prevent the creation of centralized hierarchies but does not impose them either. As an homomorphism with natural hierarchies, individuals can create their own hierarchies and, if they want, coordinate them with those of other individuals or centralize them, according to the problem they are designed to solve.

3.2 Translation to Intelligent Environments

Any coordinating structure, such as hierarchies, is strictly bounded to the nature of the blocks it coordinates. Thus, we should refresh some of the design principles of the indirect control mechanism, to understand how hierarchies are extended and applied to it:

- Decision rules are grouped in sets: the agents explained in section 2.
- Agents are represented in the blackboard layer [7] and can be activated/deactivated through it.
- In order to replicate the natural organization of preferences, agents must *belong to* a user or group of users and may be tagged with the *activity/purpose* they are designed for and *located* in the place they affect.
- To keep track of the natural responsibility chain, agents must be related with the elements *affected* by their rules.

3.3 Hierarchies Structure and Definition: Multilayer Filter

Once decided not to impose any kind of structure to the users when creating their hierarchies, and once defined the indirect control blocks' structure, we must define the means by which users are to express their hierarchies. In a rough approach we distinguish two kinds of conflicts in the ubicomp domain: those with users directly involved and those without them. While in the former technologies can help them to ease their policies, in the latter technologies are the only mean for users to apply any policy. Thus, we will present our solution paying especial attention to hierarchy automation in the indirect control.

According to the nature of conflicts, we have implemented a layer structure to solve conflicts (See Figure 1(b)) acting as a series of filters between the user desire and its effect in the world. In doing so, the AmILab layer structure (see Figure 1(a)) has been used to deal with conflicts of different nature, mainly **ownership** and **collisions**. Collisions can be classified in context–dependent and context–independent, according to whether the only important factors to solve the collision are the elements colliding and the point of collision or whether there is any other relevant element in the resolution. This last type of conflict is the one with more elements involved in the solution. The other two can be solved in the *Context layer*: ownership conflicts in the *Privacy layer* and context–independent collisions in the *Blackboard*, respectively. Context–dependent collisions (the focus of this paper) must be solved in the interaction layer for what we propose the use of the same rule–based agents mechanism mentioned in section 2. Allowing end–users to **use the same programming structure they use to program their preferences to program the hierarchies to control that preferences**.

Even though any hierarchy can be programmed through the rule–based agent mechanism, ownership conflicts and context–independent collisions can be solved in the context layer too, being more natural for their purpose. The choice will depend on the users and, among other things, will reflect the degree of interdependence and scale of their natural hierarchy (see section 3.1).

Firstly, the Blackboard layer provides a priority queue to each element [14] with a default policy to apply in case of collision (i.e. if despite the rest of the layers a collision reaches this point, the default policy is applied).

Secondly, the Privacy layer allows users to establish access rights to the elements they own [9]. Besides being used as a privacy filter it helps in constructing hierarchies where the owner is the only relevant factor in the policy (i.e. users control freely the access to their elements).

Finally, through the Interaction layer, users can create structures to control the environment that are, in turn, represented as part of the environment. In this way they can create agents whose rules control the status of another agent (instead of a physical element of the environment). Similarly, a rule can be used to set the privacy preferences. This mechanism to control indirect control structures allows the creation of hierarchies in many levels as desired. The complexity of the system –interdependence and scale– is up to the inhabitants and their natural hierarchies. Interdependence is easy to see (while not always to understand) in the Blackboard as the graph created by all the relations *affects*. Depending on the scenario this graph will range from pyramidal structures to unconnected graphs or entangled networks. With a person/s behind each agent, an element of the environment in each leave of the graph and, in between, a complex structure of conditions that, as a whole, governs the overall automatic behavior of the environment. Scale, on the other hand, can be appreciated in the different levels in which hierarchies can be expressed. To illustrate it with an example, lets consider two users sharing a house. User A prefers the light level to be low while user B prefers it high. In this situation, they can control their preferences through a single agent (associated to both of them) in which three rules codify their preference: "if user A is in the house but not B then set the light level to low", "if user B is in the house but not A then set the light level to high" and if "both A and B are in the house set the light level to medium". Conversely, they can have an agent for each of them codifying their personal preferences, another shared agent codifying their mutual preferences (i.e. what they want when they are together) and a meta–agent deactivating their personal agents and activating the shared one when both of them are in the house and vice versa. Or, finally, they can do without agents and establish a default policy in the Blackboard (since no other factor but themselves and the light is present in the conflict) to establish *the average* as the desired value for the light when a conflict arise. While codifying the same behavior, the three approaches present a different scale and they will be preferred over the others according to the idiosyncrasy of the social group. Thus, the former will be more frequent in situations in which most of the preferences are shared (e.g. a marriage sharing a house), the second when each individual normally decides alone and some coordinating mechanism is required (e.g. student roommates) while the third one is more natural to sporadic environments in which personal preferences are secondary (e.g. a laboratory hallway). These structures have been observed in the three real environments in which the system is deployed: a simulated living–room in the AmILab laboratory (Autonomous University of Madrid, Spain), a simulated security chamber in Indra's facilities (Madrid) and an intelligent classroom in the Itechcalli laboratory (Zacatecas, Mexico).

4 Conclusions

To face the problem of having diverse inhabitants with different preferences, automatic environments need to provide means through which to **create coordination structures**. Personal environments present an specific domain in which some extra principles, such as the construction of family identity or an organic evolution, have to be consider.

To deal with the diversity of environments and social groups, the coordination mechanism has to be **flexible** enough to adapt to different social structures. We pointed at interdependence and scale (i.e. the complexity measures of Y. Bar–Yam [13]) as useful measures to characterize the adaptability of a system to different social organizations.

Finally, regarding the end–user programming factors, we stated that **the user must be able to create her own hierarchies**, excluding any third human parties or automatic systems from this process. This forced us to get as close as possible to the end user, for what three factors have been of main importance: a flexible and **multiple categorization of agents** allowing concepts such as ownership, location, task, goal or effect without imposing any; a **multi–layer structure** to deal with specific types of conflicts easily, but not restricting conflict's solutions to a specific layer and, finally, the **use of the same**

programming structures with which users program their preferences to program the hierarchical structures.

Acknowledgments. Thanks to Gemma de Castro and Dominik Schmidt for their contribution to this work. This work was partially funded by the Spanish Ministry of Science and Technology through the HADA project(TIN2007-64718) and by the chair UAM–Indra of Ambient Intelligence.

References

1. Mozer, M.M.: The neural network house: An environment that adapts to its inhabitants. In: Procs. of the AAAI Spring Symposium on Intelligent Environments. AAAI Press, Menlo Park (1998)
2. Cook, D.J., Youngblood, M., Heierman, E., Gopalratnam, K., Rao, S., Litvin, A., Khawaja, F.: Mavhome: An agent-based smart home. In: Procs. of the IEEE International Conference on Pervasive Computing and Communications, pp. 521–524 (2003)
3. Bischoff, U., Kortuem, G.: Rulecaster: A macroprogramming system for sensor networks. In: OOPSLA Workshop (2006)
4. Brdiczka, O., Reignier, P., Crowley, J.L.: Supervised learning of an abstract context model for an intelligent environment, smart objects and ambient intelligence. In: SOC-EUSAI 2005, Grenoble 2005 (2005)
5. Myers, B.A., Pane, J.F., Ko, A.: Natural programming languages and environments. Commun. ACM 47(9), 47–52 (2004)
6. García-Herranz, M., Haya, P., Esquivel, A., Montoro, G., Alamán, X.: Easing the smart home: Semi-automatic adaptation in perceptive environments. Journal of Universal Computer Science 14(9), 1529–1544 (2008)
7. Haya, P.A., Montoro, G., Alamán, X.: A prototype of a context-based architecture for intelligent home environments. In: Meersman, R., Tari, Z. (eds.) OTM 2004. LNCS, vol. 3290, pp. 477–491. Springer, Heidelberg (2004)
8. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York (1980)
9. Esquivel, A., Haya, P.A., García-Herranz, M., Alamán, X.: Managing pervasive environment privacy using the fair trade metaphor. In: International Workshop on Pervasive Systems, PerSys 2007 (2007)
10. Kakas, A.C., Moraitis, P.: Argumentation based decision making for autonomous agents. In: AAMAS, ACM, pp. 883–890 (2003)
11. Maslow, A.H.: Motivation and Personality. Harper, New York (1954)
12. Davidoff, S., Lee, M.K., Yiu, C., Zimmerman, J., Dey, A.K.: Principles of smart home control. In: Dourish, P., Friday, A. (eds.) UbiComp 2006. LNCS, vol. 4206, pp. 19–34. Springer, Heidelberg (2006)
13. Bar-Yam, Y.: Analyzing the effectiveness of social organizations using a quantitative scientific understanding of complexity and scale. NECSI (May 2007)
14. Haya, P.A., Montoro, G., Esquivel, A., García-Herranz, M., Alamán, X.: A mechanism for solving conflicts in ambient intelligent environments. Journal Of Universal Computer Science 12(3), 284–296 (2006)

Wireless Sensor Networks in Home Care

Dante I. Tapia[1], Juan A. Fraile[2], Sara Rodríguez[1], Juan F. de Paz[1], and Javier Bajo[2]

[1] Departamento de Informática y Automática, University of Salamanca,
Plaza de la Merced s/n, 37008 Salamanca, Spain
{dantetapia,srg,fcofds}@usal.es
[2] Pontifical University of Salamanca
Compañía 5, 37002 Salamanca, Spain
{jafraileni,jbajope}@upsa.es

Abstract. Ambient Intelligence has acquired great importance in recent years and requires the development of new innovative solutions. This paper presents a novel architecture which integrates a service oriented approach into Wireless Sensor Networks to optimize the construction of Ambient Intelligence environments. The architecture proposes a new and easier method to develop distributed intelligent ubiquitous systems, where devices can communicate in a distributed way independent of time and location restrictions. The architecture is easily adapted to changing environments. A prototype system has been proposed to test this architecture. This system is aimed to improve health care and assistance to dependent persons in their homes. Preliminary results are presented in this paper.

Keywords: Ambient Intelligence, Wireless Sensor Networks, Service Oriented Architectures, Health Care, Tele-monitoring, Embedded Devices.

1 Introduction

People are currently surrounded by technology which tries to increase their quality of life and facilitate the daily activities. However, there are situations where technology is difficult to handle or people have a lack of knowledge to use it. For these reasons, Ambient Intelligence tries to adapt the technology to the people's needs by proposing three basic concepts: ubiquitous computing; ubiquitous communication; and intelligent user interfaces [1]. To reach these objectives, it is necessary to develop new functional architectures capable of providing adaptable and compatible frameworks, allowing personalized access to services and applications regardless of time and location restrictions. A functional architecture defines the physical and logical structure of the components that make up a system, as well as the interactions between those components [1].

This paper briefly describes SYLPH (Services laYers over Light PHysical devices), a Service Oriented Architecture (SOA) based architecture [2] [6] for Wireless Sensor Networks (WSNs) [5] [9] and explains how this architecture has been used to design a prototype system for a real case scenario. SYLPH is a novel architecture which integrates a SOA approach with WSNs for building systems based on the Ambient Intelligence paradigm [1]. The architecture focuses on distributing the systems'

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1106–1112, 2009.
© Springer-Verlag Berlin Heidelberg 2009

functionalities into services. This model provides a flexible distribution of resources and facilitates the inclusion of new functionalities in highly dynamic environments [4]. SYLPH is based on the Ambient Intelligence (AmI) paradigm [10]. The architecture formalizes the integration of services, communications and wireless technologies to automatically obtain information from users and the environment in an evenly distributed way, focusing on the characteristics of ubiquity, awareness, intelligence, mobility, personalization, participation, etc., all of which are concepts defined by Ambient Intelligence.

Next, the problem description is introduced and it is explained why there is a need for defining a new architecture. Then, the architecture is briefly described. Subsequently, a case study is presented, describing how SYLPH has been used to define a wireless infrastructure into a tele-monitoring home care scenario. Finally, the results and conclusions are presented, including future lines of work.

2 Ambient Intelligence in Health Care Environments

Ambient Intelligence is an emergent multidisciplinary area [7] [8] based on ubiquitous computing [11]. Ambient Intelligence has an influence on the design of protocols, communications, systems integration and devices [13]. Ambient Intelligence proposes new ways of interaction between people and technology, making the latter to be adapted to the people's necessities and the environment where they are [1] [4]. This kind of interaction is achieved by means of technology that is embedded, pervasive, non-invasive and transparent for users, with the aim of facilitating their daily necessities [1]. Thus, it is necessary to provide efficient solutions that allow building Ambient Intelligence environments. These environments require context-aware technologies to acquire data and intelligent systems to transform the data into knowledge.

One of the key aspects for the construction of Ambient Intelligence environments is obtaining context information through sensor networks. There are different technologies for implementing wireless sensor networks, such as ZigBee, Bluetooth or Wi-Fi [12] [14]. However, their main problem is the difficulty for integrating devices from different technologies in a single network [3]. Another key aspect is the knowledge management, and this can be done by means of distributed systems and architectures, as for example SOA.

There are several attempts to integrate WSNs and a SOA approach [5]. In SYLPH, unlike those approaches, services are directly embedded on the WSN nodes (i.e. an electronic device attached to a network) and can be invoked from other nodes in the same network or other network connected to the former one. Moreover, in those proposals it is not enough considered the necessity of minimize the overload of the services architecture on the devices. SYLPH focuses specially on devices with small resources in order to save CPU time, memory size and energy consumption. Furthermore, as said above, SYLPH considers the possibility of connecting multiple WSNs based on different radio and link technologies [13], whilst other approaches do not.

One of the priorities of the Ambient Intelligence is health care. There are several health care developments for tele-monitoring based on WSNs [9]. However, these developments do not take into account their integration with other systems and technologies, so they are difficult to be adapted to new situations. SYLPH has been designed in order to face some of these issues. As indicated in the next section, SYLPH is an architecture that enables the integration of WSNs focusing on providing a greater simplicity of deployment and optimizing the reutilization of the available resources in such networks.

3 SYLPH Architecture Description

SYLPH is a distributed architecture which integrates a SOA approach over WSNs for building systems based on the Ambient Intelligence paradigm. The main objective is to distribute resources over multiple WSNs by modeling the functionalities as independent services. A service oriented approach has been chosen because such architectures are asynchronous and non-dependent on context (i.e. previous states of the system). It is important to mention that the main objective of this paper is not to make an extensive description of SYLPH1[1] but its use for Ambient Intelligence environments, such as the home care scenario described in the next section.

SYLPH can be executed over multiple wireless devices independently of their microcontroller or the programming language they use. SYLPH works in a distributed way so that the most application code does not have to reside only on a central node. SYLPH allows the interconnection of several networks from different wireless technologies, such as ZigBee or Bluetooth. Thus, a node designed over a specific technology can be connected to a node from a different technology. In this case, both WSNs are interconnected by means of a set of intermediate gateways connected simultaneously to several wireless interfaces. SYLPH allows applications to work in a distributed way and independently of the lower layers related to the wireless sensor networks formation and the radio transmission amongst the nodes that conform them.

Figure 1 shows the different layers of SYLPH. These layers are added over the application layer of each WSN stack. The SYLPH Message Layer (SML) offers the possibility of sending asynchronous messages between two wireless devices through the SYLPH Services Protocol (SSP). The messages specify the origin and target nodes and the service invocation in a SYLPH Services Definition Language (SSDL) format. SSDL describes the service itself and its parameters to be invoked. Applications can communicate directly amongst devices using the SML layer or by means of the SYLPH Services Directory (SSD), that uses, in turn, the mentioned SML layer. The SSD is used as a directory of the services offered by the network nodes.

Figure 2 shows the basic schema of the communication between two ZigBee devices using SYLPH. In such figure can be seen the protocols that allow the peer-to-peer interaction between layers of the same level at the two wireless devices.

---

[1] For further information about SYLPH, please visit the BISITE Research Group web page (http://bisite.usal.es).

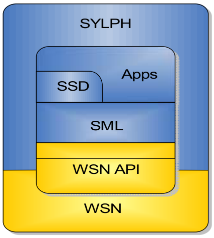


Fig. 1. SYLPH architecture layers

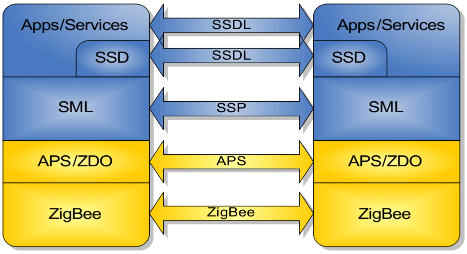


Fig. 2. Communication between two ZigBee devices using SYLPH

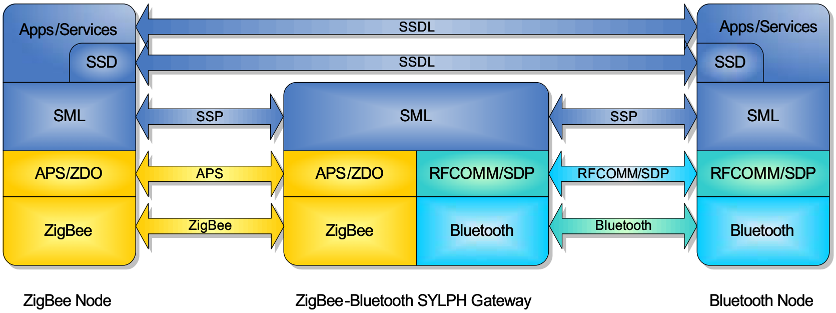


Fig. 3. SYLPH over ZigBee and Bluetooth networks

Several heterogeneous WSNs can be connected using a SYLPH Gateway. Figure 3 shows a ZigBee network and a Bluetooth network working together using SYLPH over them. The SYLPH Gateway is connected to several sensor networks through different hardware interfaces. Thus, it can forward messages amongst the different networks to which it belongs. From the application layers' point of view, there is no difference between invoking a service stored in a node in the same sensor network or in a different network. The interconnection of the different WSNs is transparent to the service invoking. Figure 3 is an example of this approach. If a ZigBee node invokes a service in a Bluetooth node, the ZigBee node will look for the service in a SSD belonging to the ZigBee network. The entry stored in the services table of the SSD points, in fact, to the SSP address of the SYLPH Gateway. When the ZigBee node invokes the service in the Bluetooth node, the SYLPH Gateway forwards the call message to the Bluetooth node through its Bluetooth hardware interface.

4 Case Study: Home Care Patients Monitoring

A tele-monitoring prototype system has been designed for improving health care of dependent people at their homes using the SYLPH architecture. The system makes use of several wireless sensor networks for obtaining context information in an automatic and ubiquitous way. A network of ZigBee devices are installed all over the home of each dependent patient to be monitored. Amongst the deployed ZigBee sensors there is a remote control carried by the monitored person that also includes a button which can be pressed in case of remote assistance or urgent help. Moreover, there is a set of ZigBee sensors for controlling the environment (light, smoke, temperature, etc.) in which the patient lives. Furthermore, there are several ZigBee actuators that physically respond to the changes in the environment (e.g. light dimmers and fire alarms). Finally, biomedical sensors [13] will allow the system to acquire continuously data with the vital signs of the patient. In this case, each patient carries three different sensors: an ECG (Electrocardiogram) monitor, an air pressure sensor acting as respiration monitor and a MEMS (Micro-Electro-Mechanical Systems) triaxial accelerometer for detecting falls.

All devices work as SYLPH nodes and can offer/invoke services all over the network. There is also a computer connected to Internet and to a remote health care centre. This computer also acts as a ZigBee master node through a physical wireless interface. On the other hand, the computer is the master node of a Bluetooth piconet (i.e. an ad-hoc network of Bluetooth devices) created by biomedical sensors acting as slave nodes. At the SYLPH level, the computer works as a SYLPH gateway so that it connects both wireless sensor networks each to other. Furthermore, it acts as a SSD and the rest of the nodes can register their own services.

Figure 4 shows a basic example of the system's communication and infrastructure. In this case, a smoke sensor detects a smoke level higher than the specified threshold. Then, the sensor invokes a service offered by the node which carries the fire alarm, making it to ring. At the same time, it also invokes a service offered by the master node, which sends an alarm through the Internet towards the remote health care centre. The caregivers can establish a communication over VoIP or by means of a webcam at home in order to check the incidence. The patient can also ask for assistance by pressing the alert button of the remote control or making a call through the VoIP terminal.

Furthermore, each patient is not only monitored at home, but also at its medical centre. The ZigBee remote control carried by each patient has a unique electronic label that identifies him or her. So, there are ZigBee and Bluetooth networks all over the medical centre so that the patient's ZigBee identification label and its Bluetooth biomedical sensors can automatically connect to the SYLPH network. Thus, when the patient goes into the centre the system realizes that he or she is inside. In fact, the system knows at every moment which patients and caregivers are inside the centre and at which point of the building.

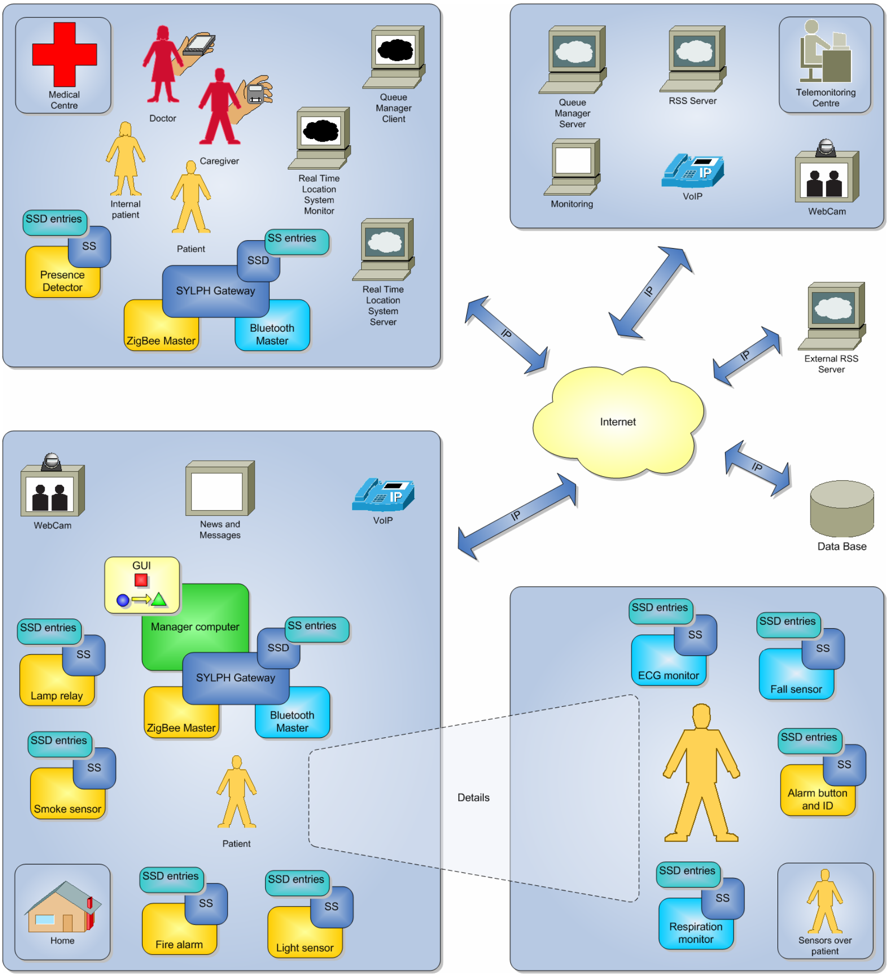


Fig. 4. Schema of the tele-monitoring system using SYLPH

5 Results and Conclusions

As described above, SYLPH allows wireless sensor devices from different technologies to work together in a distributed way. In addition, these devices do not require large memory chips or fast microprocessors in order to work with SYLPH.

SYLPH allows the possibility of adding new components at execution time. In this sense, SYLPH proposes a model that goes a step ahead in designing systems for Ambient Intelligence environments, such as home care, providing features that make it easily adaptable to several developments. SYLPH also allows the creation of WSNs that facilitates intelligent and distributed services. On the other hand, the automation of tasks and patient monitoring improves the system security and efficiency for care services to dependent patients.

Future works on SYLPH include the improvement of the SYLPH Gateways performance, support for other WSNs different from ZigBee or Bluetooth and the development of better tools for generating stubs and skeletons from the human-readable SSDL language. Regarding the tele-monitoring system described, the next steps consist on developing this proposal and implement it in a real scenario.

Acknowledgments. This work has been supported by the Spanish Ministry of Science and Technology project TIN2006-14630-C03-03.

References

1. Aarts, E.H., Encarnação, J.L.: True Visions: the Emergence of Ambient Intelligence. Springer, New York (2006)
2. Cerami, E.: Web Services Essentials Distributed Applications with XML-RPC, SOAP, UDDI & WSDL, 1st edn. O'Reilly & Associates, Inc., Sebastopol (2002)
3. Lei, S., Xu, H., Xiaoling, W., Lin, Z., Cho, J., Lee, S.: VIP bridge: Integrating several sensor networks into one virtual sensor network. In: International Conference on Internet Surveillance and Protection, ICISP (2006)
4. Matthew, G.: 802.11 Wireless Networks: The Definitive Guide. O'Reilly & Associates, Inc., Sebastopol (2005)
5. Moeller, R., Sleman, A.: Wireless networking services for implementation of ambient intelligence at home. In: Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems, ICCDCS (2008)
6. Ricci, A., Buda, C., Zaghini, N.: An agent-oriented programming model for SOA & web services. In: 5th IEEE International Conference on Industrial Informatics (INDIN 2007), Vienna, Austria, pp. 1059–1064 (2007)
7. Tapia, D.I., De Paz, Y., Bajo, J.: Ambient Intelligence Based Architecture for Automated Dynamic Environments. In: Borrajo, D., Castillo, L., Corchado, J.M. (eds.) CAEPIA 2007. LNCS, vol. 4788, pp. 171–180. Springer, Heidelberg (2007)
8. Tapia, D.I., Bajo, J., Corchado, J.M., Rodríguez, S., De Paz, J.F., Sánchez, J.M., et al.: Arquitectura Multiagente para Entornos Dinámicos: Tecnología e Inteligencia Aplicadas. In: Ubiquitous Computing and Ambient Intelligence, Congreso Español de Informática (CEDI-UCAMI 2007), Thompson-Paraninfo, Zaragoza, España (2007)
9. Varshney, U.: Improving Wireless Health Monitoring Using Incentive-Based Router Cooperation. Computer 41(5), 56–62 (2008)
10. Weber, W., Rabaey, J.M., Aarts, E.: Ambient Intelligence. Springer, New York (2005)
11. Weiser, M.: The computer for the 21st century. In: Baecker, R.M., Grudin, J., Buxton, W.A., Greenberg, S. (eds.) Human-computer interaction: toward the year 2000, pp. 933–944. Morgan Kaufmann Publishers, Inc, San Francisco, CA, USA (1995)
12. Hewlett-Packard: Understanding Wi-Fi. Hewlett-Packard Development Company (2002)
13. Yeow, T.W.: Biomedical Sensors. Encyclopedia of Biomedical Engineering. John Wiley & Sons, Chichester (2005)
14. ZigBee: ZigBee Specification Document 053474r13. ZigBee Standards Organization. ZigBee Alliance (2006)

Indoor Localization Based on Neural Networks for Non-dedicated ZigBee Networks in AAL

Rubén Blasco, Álvaro Marco, Roberto Casas, Alejandro Ibarz, Victorián Coarasa, and Ángel Asensio

Tecnodiscap group, University of Zaragoza, María de Luna 3, 50018, Zaragoza, Spain
`{rblasco,amarco,rcasas,aibarz,vcoarasa,aasensio}@unizar.es`

Abstract. Indoor localization is one of the most appealing technologies in Ambient Assisted Living (AAL) applications, providing support for diverse services such as personal security, guidance or innovative interfaces. Dedicated systems can be deployed to provide that information, but it is possible to gain advantage of available elements to compute a location without requiring additional hardware. In this paper, a ZigBee network designed for a home control application is improved with a localization functionality based on neural networks, achieving room-level accuracy, and non introducing additional infrastructure constraints to the original application.

Keywords: Indoor localization, Neural networks, ZigBee, Ambient Assisted Living.

1 Introduction

There is a wide variety of systems that can provide the location of mobile indoors. Aside from those based on vision [1]—which do not require from a specific device to be located—or inertial ones [2]—on which the device measures its acceleration to integrate its position—, localization systems operate by making mobile device interact with some fixed devices. Usually, the mobile device receives some signal emitted by the fixed devices or vice versa, that can be ultrasonic [3][4][5], light [6][7][8] or RF [9][10][11].

These signals can be used in many different ways, like measuring the angle of arrival (AOA) and obtaining a location by triangulation, or estimating the distance between the mobile devices and the fixed devices by measuring time of arrival (TOA), time difference of arrival (TDOA) or received signal strength (RSSI), and compute location by multilateration or just provide a reference location by proximity to the fixed devices.

Higher accuracy is achieved with those systems that can measure these signals precisely, and can use analytic models to compute location (intersections of beams in AOA, circles in TOA or hyperbolic in TDOA), which normally require from specific deployments. An example of these systems is the one from Ubisense [12], which uses ultra wide band (UWB) signals and combines AOA and TDOA measurements to achieve an accuracy of a few centimeters. However, the cost of this system is in accordance with its accuracy, and can be prohibitive for many applications.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1113–1120, 2009.
© Springer-Verlag Berlin Heidelberg 2009

An interesting approach is to use Wi-Fi to compute location, like WHEREMOPS [9] WiPS [10] and Ekahau [13] systems do. That way, it is possible to reuse our WLAN existing infrastructure, but only Wi-Fi enabled devices can be located. These systems usually do not use analytic models to estimate location, but often compute a RSSI fingerprint for every possible location in the application scenario. The accuracy of these systems is about 3 meters—going to 1 meter in the best cases—, which can be enough for many applications.

Although Wi-Fi networks are available in many environments, in AAL applications it is common that wireless sensor networks (WSN) based on lighter technologies than Wi-Fi—such as ZigBee—are used. ZigBee [14] is a wireless standard designed to ease deployment of WSN, providing auto routing, self-healing mechanisms, very low power consumption, and it is of big interest to use it to provide location of mobile devices. ZigBee manufacturers like Texas Instruments include location-ability in their ZigBee transceivers [15], but as with other localization systems, it is required the mobile device to receive data from a number of fixed—reference—devices. Computing the location of a mobile device using analytic models requires 3 measurements to achieve a solution and more are needed to ensure redundancy and robustness to errors [16]. The same happens with fingerprint-based location techniques, which provide better results when more measurements are available. This means that it is often necessary to oversize the network in order to enable localization.

In this paper, it is proposed to use neural networks (NN) in order to compute location of a mobile device within a ZigBee network, but using the minimum number of nodes, i.e. only those strictly needed to the targeted application and the network requirements, without deploying extra nodes. First, scenario application and neural network design is presented. Then, localization provided by the neural networks is evaluated and lastly, conclusions and future work are presented.

2 Setup Description

Is it possible to perform localization by using a non-dedicated ZigBee network? Within the MonAMI project [17] we have defined a smart AAL scenario where every sensor, actuator and HMI device uses ZigBee as communication network, with similar approaches in other related projects like EasyLine+ [18] or AmbienNet [19], and the objective is to provide these installations with localization services without increasing the cost, i.e., using the existing ZigBee infrastructure.

Obviously, we do not expect to achieve the accuracy of the localization systems presented in previous sections but just to provide room-level accuracy, which is enough for many home environments applications.

2.1 Scenario Setup

To probe that hypothesis, we have deployed a ZigBee network on 4th floor of Bldg. Ada Byron at University of Zaragoza. In this environment—similar to a hospital or a residence—a number of ZigBee outlet actuators (devices able to connect and disconnect mains powered devices) has been distributed. These devices represent a set of sensors and actuators commonly present in smart environments. Its distribution was not designed for localization and follows a pseudo-random distribution, intended only to provide communication coverage in the entire scenario. From now on, we will call these reference devices as beacons, whose location of is represented in figure 1 by caps pointing to lobe direction of biggest radiation.

A mobile ZigBee device consisting on a HMI with two buttons, accelerometer—fall detector—and temperature sensor will be used as the localizable device, which we will denote as tag, and its position can be seen also in figure 1.

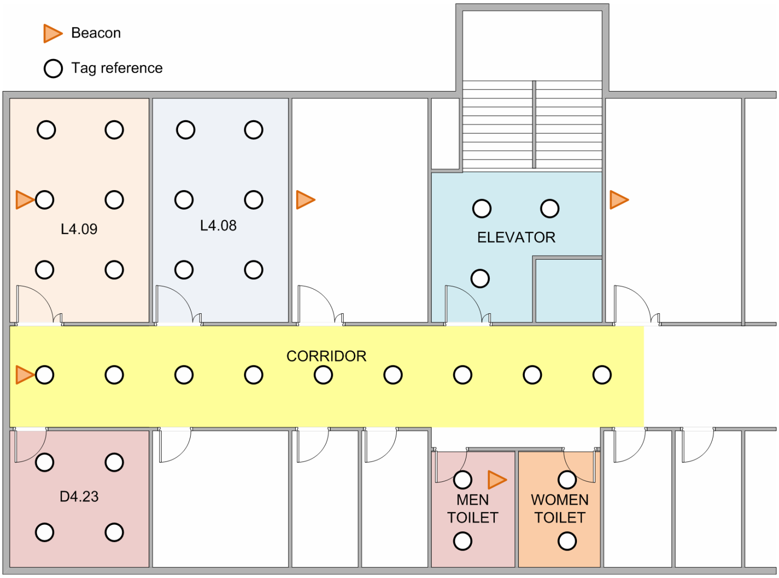


Fig. 1. Scenario map. Fixed devices are denoted by arrow caps, and tag reference locations are represented by circles.

2.2 Test Definition

With the scenario presented, we consider the following experiment to validate our hypothesis.

The objective is to achieve room-level accuracy, so seven interest areas in the scenario are defined as location targets. Then a user, carrying the tag connected to the ZigBee network, will take up several positions inside each of these localization areas.

For every of these positions, four measured will be collected, each one with a different user orientation (north, west, south, east). Each measurement consists on the received signal strength (RSSI) with which the tag receives its neighbors.

This sequence was repeated in 32 locations, generating a pattern database with 500 sets of RSSI measurements. Information about the location of the beacons was not stored, as only RSSI was used to estimate the location of the tag.

3 Neural Network Design

With the aim of discovering what NN drew best results, two methods were performed; a MultiLayer Perceptron (MLP) based on different architectures and a Self Organizing Map (SOM).

3.1 Data Adaptation

Our database were normalized among (-1, 1) to facilitate the training of both networks. Previously, like the RSSI range is from -100 to 0, when a beacon is not listened by the tag, we assign the -105 value.

Next step was filtering the database. Three groups were made from database depending on the rule used to filter:

- *Group 1*: points where the tag detects at least one beacon.
- *Group 2*: points where the tag detects at least two beacons.
- *Group 3*: points where the tag detects at least three beacons.

There was no need to continue filtering because only 5 beacons were deployed. Finally, for every single group, three subgroups were created randomly: train (80% of points), test (10%) and validation (10%).

3.2 MLP

MultiLayer Perceptron architecture is one of the best NN to classify patterns [20]. MLP is a feedforward NN, trained with the standard backpropagation algorithm. MLP learns to link input data to a desired response. As it is supervised, it requires a set of known patterns and their corresponding responses during training process. With one or two hidden layers, it can approximate virtually any input-output map.

First of all it was necessary to decide the most suitable architecture to face the problem. So, a 5×M×7 architecture was chosen; where the 5 inputs are the RSSI of each beacon, M represents the number of hidden neurons and 7 the number of outputs which correspond to the locating areas.

To analyze network outputs, two thresholds were defined: Activation Threshold (AT) and Selection Threshold (ST). An output must exceed AT to be considered active and must be, at least ST, far away from the next output to be declared as the winner. In this case, AT and ST were fixed at 0.1 and 0.4 respectively.

Some MLP were simulated changing the value of M. For each group described in the previous section, the train subgroup was used to train the network and the test subgroup was used to detect the best network according to the minimum error obtained. Finally, five more simulations were done using the chosen architecture and defining its error as the mean of the validation error of all the simulations (validation subgroup was neither used for training the network nor for selecting it).

3.3 SOM

SOM are a good tool for the visualization of high-dimensional data. They can convert n-dimensional data into a simple geometric, for example 2-dimensional. They can be used also to produce some kind of abstractions [21] and they are really useful grouping data with similar characteristics. Matlab and the SOM ToolBox [22] has been used in order to train and generate the SOM.

In order to find out the best organizing map, one simulation, which consisted of 30 repetitions, was done training some maps and using the train subgroup. The selected map was the one which threw the minimum test error. Finally, this simulation was repeated five times and the error of the map was defined as the mean of the validation error of all the simulations (as with the MLP, validation subgroup was neither used for training the map nor for selecting it).

To quantify the error of each map, two methods were used:

a) *One label*. Neurons of the map are labeled by vote, that is to say, a neuron is labeled with the name of the zone where it has been activated more times.
b) *Two labels*. Neurons of the map are labeled by frequency (e.g. if a neuron has been activated 10 times by zone A, 5 by zone B and 1 by zone C, the neuron will be labeled with two labels: zones A and B). It does not happen everywhere in the map, but in specific boundaries. As we will consider later in conclusions, this method can be applied because of the surrounding peculiarities of the problem.

In both methods an error will be assumed every time a neuron is activated by a zone which is not linked to.

4 Evaluation

4.1 MLP

Tests conducted with MLP have not provided clear results. There is not a relationship between the number of beacons that the tag listens to and the location error. Nevertheless, all the methods that are using RSSI for location agree that the more beacons the better accuracy. MLP can be considered as a universal functions estimator [23] but, in this case, we have not enough information to obtain an algorithm. Maybe it is the reason why this architecture cannot provide a good solution.

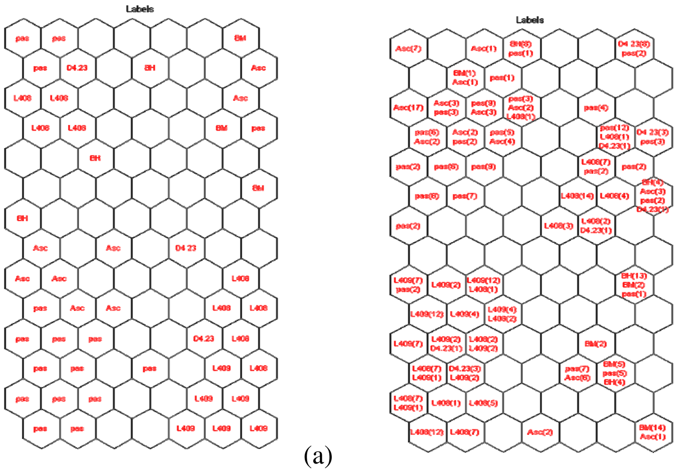
The best architecture found in this case has been 5×18×7 with an average validation error of 20.2%.

4.2 SOM: One Label Method

The first tests conducted with SOM showed a clear relationship between the number of beacons that the tag listens and location error (table 1). The average validation error decreases when the tag listens to more beacons. However, this error is still high, around 18.7 % in the best case (group 3).

Table 1. Classification error with SOM using only one label

Group number	Test error	Validation error
1	17.20%	24.00%
2	12.65%	19.59%
3	11.11%	18.66%



(a) (b)

Fig. 2. SOM map for one label (a) and two labels (b) methods

4.2 SOM: Two Labels Method

This fact leads to analyze the physical nature of the problem. The localization systems based on RSSI with ZigBee locates their resolution among 3m and 5m depending on the number of beacons (or fixed nodes). In this case, where a network with a minimum number of beacons has been forced, it would be logical to think that the accuracy in meters should be 5 meters or more.

Also, in the transition of localization areas without physical separation (door setting or the separation area without wall between the elevator hall and the corridor) the RSSI will be very similar. Therefore, if this system provided a location in meters it would not be difficult to provide coordinates and a radius of confidence. When this situation takes place, each time one user is placed less than 5 meters away from the contiguous area, the system will detect the user to be in one of the areas located in the radius of confidence within the two areas.

How can this idea be explained to SOM? If the activation of neurons is studied in detailed, it will be observed that there are areas in the map which are activated mainly by stimulus of two areas, i.e. “zone A” (7) and “zone B” (2). If the wining label is considered, in this case” zone A”, when points are set in the “zone B”, which are very close to this area, and the map activates the neuron, the situation is considered as an error. Nevertheless when using any other method, if it is within the radius of confidence, it is considered correct.

Therefore, if two labels are considered in those neurons responding to two specific boundaries, it won’t be know whether the user is located in the “zone A” or in “zone B”, but the area in which the user is located will be known, and probably even the boundary among them. Obviously, it can occur that the neuron is activated by stimulus in the two areas separated by 5 meters, being this unusual situation. In this case one only label will be considered.

Taking into account this situation the error in the localization is cut down drastically to 7.1% for the sample- 3. In contrast, in some situations the network feedback is "zone A"-"zone B" instead of one only area.

5 Conclusions

After analyzing the test results, we can conclude that it is possible to perform location at room-level accuracy using only the RSSI of a non-dedicated location ZigBee network as input. This system has a lower precision than other systems which are using a dedicated location ZigBee network with more fixed nodes. Nevertheless, it can provide an extra location service in smart environment which are using ZigBee at no cost.

If we can guarantee that the tag can listen to three or more beacons in every position, this system can locate with an average location error rate of 7.1 %. This result is important since it can be used to decide improving the coverage in an existing installation if localization services are wanted.

Besides that, a novel two labels classification method has been introduced, proving its relationship with the physical problem.

Future work points to reduce uncertainty in limited areas by using tracking algorithms that helps to discriminate between non-joined areas.

Acknowledgments. This work has been partially supported by the European Union, under the projects MonAmI (IST-5-0535147) and EasyLine+, and by the Spanish Ministry of Science and Technology, under the AmbienNET Project (TIN2006-15617-C03-02).

References

1. Julier, S., Bishop, G.: Tracking: how hard can it be? IEEE Computer Graphics and Applications 22(6), 22–23 (2002)
2. Jonker, P., Persa, S., Caarls, J., de Jong, F., Lagendijk, I.: Philosophies and technologies for ambient aware devices in wearable computing grids. Computer Communications 26, 1145–1158 (2003)
3. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket location-support system. In: 6th Annual international Conference on Mobile Computing and Networking, MobiCom 2000, pp. 32–43. ACM, New York (2000)
4. Ward, A., Jones, A., Hopper, A.: A New Location Technique for the Active Office. IEEE Personal Communications 4(5), 42–47 (1997)
5. Marco, A., Casas, R., Falco, J., Gracia, H., Artigas, J.I., Roy, A.: Location-based services for elderly and disabled people. Comput. Commun. 31(6), 1055–1066 (2008)
6. Want, R., Hopper, A., Falcão, V., Gibbons, J.: The active badge location system. ACM Trans. Inf. Syst. 10(1), 91–102 (1992)
7. Römer, K.: The Lighthouse Location System for Smart Dust. In: 1st international Conference on Mobile Systems, Applications and Services, MobiSys 2003, pp. 15–30. ACM, New York (2003)

8. Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., Colucci, D.: High-Performance Wide-Area Optical Tracking: The HiBall Tracking System. Presence: Teleoper. Virtual Environ. 10(1), 1–21 (2001)
9. Wallbaum, M.: WHEREMOPS: An Indoor Geolocation System. In: IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications, vol. 4, pp. 1967–1971. IEEE Press, New York (2002)
10. Kitasuka, T., Hisazumi, K., Nakanishi, T.: WiPS: Location and motion sensing Technique of IEEE 802.11 devices. In: Third International Conference on Devices Information and Applications, vol. 2, pp. 346–349. IEEE Press, New York (2005)
11. Anastasi, G., Bandelloni, R., Conti, M., Delmastro, F., Gregori, E., Mainetto, G.: Experimenting an Indoor Bluetooth-Based Positioning Service. In: 23rd International Conference on Distributed Computing Systems, pp. 480–483 (2003)
12. Cadman, J.: Deploying Commercial Location-Aware Systems. In: Fifth International Conference on Ubiquitous Computing, pp. 4–6 (2003)
13. Ekahau Real Time Location System (2009), `http://www.ekahau.com/`
14. ZigBee Alliance, `http://www.ZigBee.org`
15. Texas Instruments, Low power RF System-on-Chip for ZigBee Applications, `http://www.ti.com/corp/docs/landing/cc2431`
16. Casas, R., Cuartielles, D., Marco, A., Gracia, H.J., Falco, J.L.: Hidden Issues in Deploying an Indoor Location System. IEEE Pervasive Computing 6(2), 62–69 (2007)
17. Monami Project site (2009), `http://www.monami.info/`
18. EasyLine+ Project site (2009), `http://www.easylineplus.com/`
19. AmbienNet Project site (2009), `http://www.atc.us.es/ambiennet/`
20. Del Hoyo, R., Buldain, D., Marco, A.: Supervised classification with Associative SOM. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2687, pp. 334–341. Springer, Heidelberg (2003)
21. Kohonen, T.: The Self-Organizing Map (SOM), `http://www.cis.hut.fi/projects/somtoolbox/theory/somalgorithm.shtml`
22. SOM ToolBox. Helsinki University of Technology, `http://www.cis.hut.fi/projects/somtoolbox/SOMToolBox`
23. Funahashi, K.I.: On the approximate realization of continuous mappings by neural Networks. Neural Networks 2(3), 183–192 (1989)

Managing Ambient Intelligence Sensor Network Systems, an Agent Based Approach

Guillermo Bosch and Cristian Barrué

Technical University of Catalunya

Abstract. Smart homes and ubiquitous computing solutions are getting special interest in healthcare context as they can help extending prevalence in prefered enviroments. This paper presents an agent based implementation approach for heterogeneus flexible sensor networks in the ambient intelligence context.

1 Introduction

Average life expectancy in the EU is one of the highest in the world and is continuing to rise. The demographic shift leads to an epidemiologic shift. The increase in longevity, the number of elderly people and in the survival to acute accidents and diseases implies an increased prevalence of chronic morbidity and disability. Clearly, societal resources will not be sufficient to assist all elderly or people with disabilities, so IST are expected to play a key role in this respect. The overall objective of the european Ambient Assisted Living research programme is to enhance the quality of life of older people through the use of Information and Communication Technologies (ICT). ICT can help people to remain active in their preferred environment by increasing their autonomy, self-confidence and mobility, supporting carers, families and care organisations. Assistive Technologies (AT) play an important role in the elder quality of life enhancement, including mobility devices such as walkers and wheelchairs, as well as hardware, software, and peripherals that assist people with disabilities in accessing computers or other information technologies. Ambient Intelligence (AmI) can contribute to the prevalence in the preferred environment offering monitoring and security services among others. In this paper we describe a multi-agent system approach to implement a dynamic and flexible intelligent sensor network to support the assistive technologies involved in the EU funded project *SHARE-it*[1].

1.1 Plan of Paper

The rest of this paper is organized as follows: In section §2 a state-of-the-art in AmI context is presented. In section §3 we introduce the *SHARE-it* project and

---

[1] Authors would like to acknowledge support from the EC funded project *SHARE-it*: Supported Human Autonomy for Recovery and Enhancement of cognitive and motor abilities using information technologies (FP6-IST-045088). The views expressed in this paper are not necessarily those of the *SHARE-it* consortium.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1121–1128, 2009.
© Springer-Verlag Berlin Heidelberg 2009

the multi-agent system which our system is a part of. The sensor network system is explained in detail in §4. Finally in §5 some experiences and conclusions are discussed.

2 State of the Art

Ensuring that the Assistive Technology works not isolateeterod but integrated in a given intelligent ambiance have brought new research about Ambient Intelligence (AmI) (13)(7). According to the European IST program 1999 (ISTAG), AmI focuses on creating intelligent homes to properly help the inhabitants with their Activities of Daily Living (ADL). Europe, Japan, Singapore and USA are paying special attention to this area in houses, hospitals and even outdoors. People will live easily in digital environments in which the electronics are sensitive to people's needs, personalized to their requirements, anticipatory of their behaviour and responsive to their presence. Pollack (12) proposes three goals for intelligent assistive technologies addressing cognitive problems: (1) assure that the elder is safe and performing necessary activities, (2) assess the cognitive status in the normal living environment, and (3) compensate the cognitive impairment. AmI possibilities in environment monitorization, data gathering and feedback on user activities can contribute to fulfill these goals.

AmI implies three relatively new technologies: Ubiquitous Computation, Ubiquitous Communication and Intelligent User Interfaces. Ubiquitous Computation consists of embedding microprocessors, sensors, and actuators in everyday objects, so that they acquire some processing capability; it can be perceived how they are used and they can act in an autonomous way. The ubiquitous computing paradigm envisioned by Mark Weiser (15) is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. Some examples of this idea are the several projects in progress involving smart homes like House_n (8), Aware Home (9) or OXYGEN (3), with experimental *ad-hoc* sensorized environments to track human activity and open new living research environments in many life areas like health tracking or work spaces.

One of the pillars of AmI is Ubiquitous Communications, as information needs to be processed and shared among devices to achieve deliberative decisions. Data processing and distribution and task performing in potentially unstructured dynamic environments is a quite complex problem to solve. A next step in *always on* communications, new ubiquitous technologies (such as RFID) promise a world of networked and interconnected devices (e.g. fridge, television, vehicle, garage door, etc) that provide relevant content and information whatever the location of the user. In particular, the convergence (and future scalability) of broadband Internet and ubiquitous networks with current mobile services may emerge as the key means for providing communication and monitoring capabilities to users (1). Multiple technological solutions can be distinguished, presenting differences in aspects like power consumption, range or immunity to electromagnetic interferences. In the concrete case of radiofrequency, the systems exploit technologies widely spread like GSM, Bluetooth, WiFi, Zigbee or RFID, as well as others emerging like Ultra-WideBand ([4]) or Wireless Sensor Networks ([11]).

3 SHARE-it

The goal of this EU FP6 funded project is to develop a scalable, adaptive system of add-ons to sensor and assistive technology so that they can be modularly integrated into an intelligent home environment to enhance the individuals autonomy. The system is designed to inform and assist the user and his/her caregivers through monitoring and mobility help. Thus, this project plans to contribute to the development of the next generation of assistive devices for older persons or people with disabilities (both cognitive and/or motorial) so that they can be self-dependent as long as possible. There is a special focus on add-ons to be compatible with existing technologies and to achieve an easier integration into existing systems. *SHARE-it* also aims at adaptive systems as transparent, and consequently, easy to use to the person as possible. Scalability is meant to include or remove devices from the system in a simple, intuitive way.

The specific objectives of the project are: (1) to explore the benefits of the concept of situated intelligence to build agent-based artefacts that will enhance the autonomy of the target user group in their daily life, (2) to investigate and implement innovative forms of shared autonomy through cognitive processing of goals, monitoring, environment models and perception of unexpected events, (3) to build a mobile platform for complex real world domains, (4) to build adaptive interfaces for the target population, and (5) to target appropriately the caretaking services.

SHARE-it can obtain these objectives trough the use and/or development of mobile platforms (wheelchair and walker): (1) the semi-autonomous wheelchair Rolland III ([10]) , which is able to function safely indoors and outdoors in appropriately set-up environments, (2) Spherik ([5]), the semi-automated guided platform to around a house that uses a new kind of spherical wheels, and (3) the i-Walker platform ([6]) , the robotically augmented walker to reduce fall risk and confusion and to increase walker convenience and enjoyment.

3.1 The Multi-Agent System

Agent-based systems technology has generated lots of excitement in recent years because of its promise as a new paradigm for conceptualizing, designing, and implementing software systems. This promise is particularly attractive for creating software that operates in environments that are distributed and open.

Distributed Agents Technology is focussed on support for (the development of) large-scale, secure, heterogeneous, agent systems. Research in this area includes scalable and secure agent platforms, location services, directory services, and systems management. Support systems are not only expected to support heterogeneous agents (both architecture and programming language), but also different operating systems and hardware.

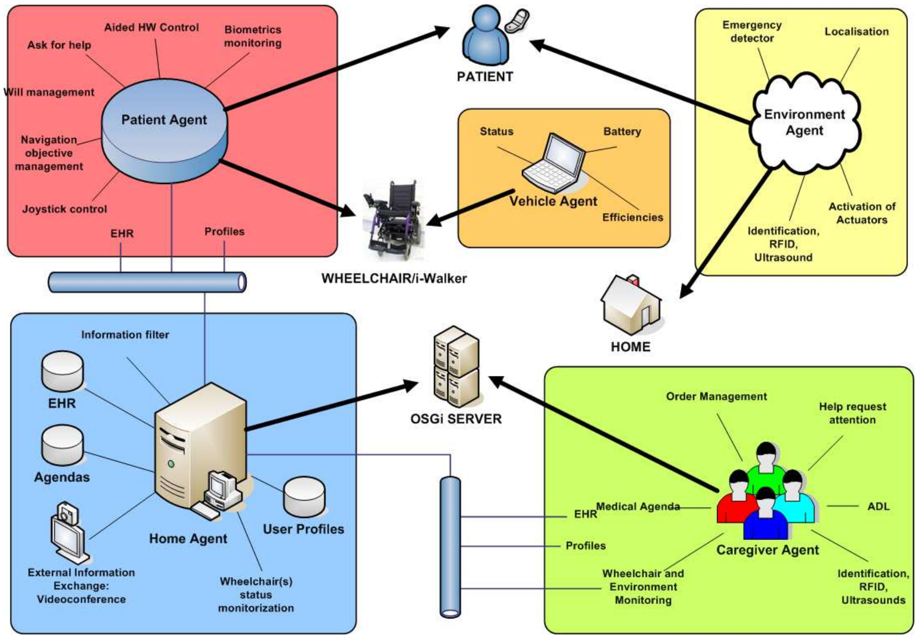


Fig. 1. *SHARE-it* MAS architecture

The research areas of multi-agent systems and distributed systems coincide, and form the research area of distributed agent computing. In short, multi-agent systems are often distributed systems, and distributed systems are platforms to support multi-agent systems.

The agent layer in *SHARE-it* (see Figure 1) is built by different agents that are in charge of specific tasks in the global system:

Patient Agents, which run in PDAs or Ultra-Mobile PCs. An instantiation of this agent provides all the available and permitted services to each user for instance security, mobility, monitoring and help services.

The Vehicle Agents are allocated in the assistive hardware devices related to the project, be it the different wheelchair (Rolland, Spherik) or in the i-Walker. Most services are related to mobility, monitoring and resource management.

The Caregiver Agents are situated in the computers belonging to the caregivers of *SHARE-it* target population as well as in their individual PDA. These agents are in charge of managing all the user's help request messages, so they can be attended properly. Also, it will be notified if any anomalies are detected in the user's biometric signals.

The Environment Agent runs in a network computer connected to all the environment network of sensors (i.e. AmI sensors). Its basic target is to distribute the information from all available sensors to all the agents interested.

4 Environment Sensor Network

In this section the mechanism that is used to manage the Ambient Intelligence sensor network is described. The goal is to perform the autonomous data gathering and information extraction from the environment, as well as the interaction with the Ambient Intelligence devices that are included in the habitual environment of the users. These sensors provide data, which has to be conveniently stored, filtered and delivered to the Intelligent Agent System, which will provide the autonomous decision and control tasks into *SHARE-it*.

The final system ought to be intelligent enough to: (1) distinguish which data from all the available is significant for each one of the components of the system, and deliver them to those components, and (2) extract and infer information from the raw data retrieved from the environment and transform it into meaningful ontology concepts, so a better knowledge of the environment is acquired.

Both capabilities will be achieved by means of an Intelligent Multi-Agent System, which constitutes the Agent Based Control (ABC) System of *SHARE-it*. Regarding to the environment, the ABC will provide software components to:

- Proactively take decisions about the rooms conditioning
- Provide assistive services to the mobile entities (e.g. , localization)
- Process the sensor signals in order to extract meaningful qualitative information related to services tied to those sensors
- Deploy a base to perform user activity tracking, in order to being able to record or recall user activities of daily living for adaptation and personalization.

Though some of the data presented by the environmental sensors is stored into the general database, the extraction and inference of information from the data provided by these environmental sensors is performed by the Environment Agent, which delivers it to the other Agents in the system when required.

4.1 Sensors and Technologies

SHARE-it uses a large number of sensors, encompassing the requirements of the correct functioning of the platform(s) and the purpose of monitoring and localizing the user. The sensors are both wearable and located on the platform(s) and/or in the environment. The sensor deployment for *SHARE-it* as experimental environment has been in Casa Agevole ([14]), a house designed specially to be friendly for impaired people. The technology selection for sensors has been a heterogeneous set for localization and presence (RFiD, Ultrasounds, Zigbee triangulation), domotics (Zigbee motes and KONNEX) and non-invasive biometrics wearable devices like a pulsioxymeter wrist.

Specific use of data gathering in *SHARE-it* is finalized at: (1) localizing the platform, which is an indispensable issue related to safe navigation, (2) localizing the user, (3) collecting vital data to implement and support the definition of users profiles, (4) collecting performance data derived from the analysis of the use of the platform(s) and (5) modifying the behaviour of the platform.

To this extent, information coming from the domotic sensors is considered of interest for the *SHARE-it* project. For example, it is crucial to warn the caregiver in case of certain previously defined alarms, such as fire detection, absence of movement within a defined time slot, a door left open when not expected, etc. Such information is not only useful in the case of alarm detection, but also to infer knowledge from the set of common activities of daily living of the person under the supervision of the caregiver. To that purpose, data should be gathered from a set of home automation sensors (smoke, movement, door open/closed, light sensors).

4.2 Realization: A Jadex Implementation

Environment Agent (EA) is responsible to perceive what is happening in the environment. As mentioned in the previous section, it is easy to achieve through sensors information. Sensors in a house can be easily installed in order to process their data.

According to the adaptation and dynamic requirements of the sensor network the capabilities mechanism that the agent development framework Jadex ([2]) offers is used. This reasoning engine that follows the Belief Desire Intention (BDI) model uses capabilities for the modularization of agents, whereby capabilities contain ready to use functionalities. Basically a capability is a set of beliefs, goals and plans that can be dynamically included in the agents reasoning engine.

Capabilities, from the Environment Agent point of view, will reach a slightly different significance than the original Jadex proposal, which is to reuse blocs of code that are needed in diverse agents. Here, each single model of capability has to do with a specific environmental sensing/acting framework.

Through this subsection a hypothetical, but representative and easy to understand, example is described. In this case a module named *FireCap* corresponds to a capability with a set of beliefs, goals and plans referring to all of the environmental things related to fire events in a house. Once this capability is introduced into the Environment Agent, it becomes, for instance, able to cope with fire emergencies (i.e. sending alarm messages when there is fire in the house).

Every single capability module is meaningful by itself. That means that all the beliefs, goals and plans of a sensing framework in the environment work in an independent way. Nevertheless, to have an intermodal perception of the environment the system needs the different capabilities to interact. This way the system is able to join different skills from individual capabilities. For instance: separately, the before mentioned *FireCap* can send an alarm message of fire when needed. In the same way, there could be a capability related to light functionalities in a house, *LightCap*, which would be aware of the light state (on/off) and could also change it. Interaction between capabilities lead the system to more environmental awareness and power for acting and adapting to users needs. When the system incorporates both *FireCap* and *LightCap* features in the specific hypothetical example which is analysed, the following would occur: provided that a fire situation was to come to pass in the house, the *LightCap* would switch the lights on if the situation happened at the end of the day. This would be very helpful for the user to evacuate the premises (since they would be too nervous about the situation to find light switches).

Hence, the Environment Agent is meant to be able to increase its perception functionalities. Jadex plans can dynamically load and unload capabilities. This offers the possibility to change the underlying agent and capability models at runtime. It rightly ensembles with the adaptive system of add-ons modularly integrated.

Moreover, Environment Agent implementation relies on the plug-and-play idea. Therefore, it allows the addition of a new sensor (or set of sensors) without physical reconfiguration or user intervention. EA by itself only contains a few basic features. The core of the agent enables:

- The awareness of new physical sensor addings/removals.
- The ability to add/remove capabilities.
- The ability to add/remove interaction among capabilities.
- The ability to add/remove sensors.
- The ability to check sensor readings.

To firstly create the connection when a sensor is plugged into the system, EA implements a SensorServer using Sockets. Every new physical sensor that is connected to the system acts, from the other side, as a SensorClient communicating all the necessary data to the Environment Agent.

Every type of sensor has been ontologically represented in the system. Once the physical sensor is connected to the EA, semantic information is added according to the ontological sensor representation. Thus, the agent starts to reason at the highest level of abstraction where different sensor hierarchy pertinence leads to specific treatment.

5 Conclusions

After reporting the general motivation and giving a short introduction to the domain of the *SHARE-it* project, the theoretical view on the approach has been outlined. A Multi-Agent System, especially focused on the Environment Agent, has been chosen in order to distinguish which data from all the available is significant for each one of the components of the system and to extract and infer information from the raw data retrieved from the environment. Finally, the implementation of a practical realization to manage the sensor network using the Jadex reasoning engine technology has been explained. Jadex offered the BDI agent approach, an integrated reasoning engine, allowed us standard agent ACL messaging and granted us the *capability* tool to bring dynamic flexiblity to the agents structure and behaviour.

In summary, it can be concluded that the approach has the potential of managing a sensor network that relies on the ambient intelligence concept. In the implementation section §4 a way to develop a scalable, adaptive system of add-ons to sensor and assistive technology so that they can be modularly integrated into an intelligent home is depicted. Consequently, ambient intelligence in the system is characterized by offering flexibility, handling changes in the sensor network, and maintenance, permitting the modification of agent skills to improve performance. The tests with real sensors in the *Casa Agevole* environment gave encouraging results to keep this dynamic approach and leads to more complex and stressing tests for the future.

References

[1] Itu/spu ubiquitous network societies, `http://www.itu.int/osg/spu/ni/ubiquitous/`

[2] Jadex, `http://jadex.informatik.uni-hamburg.de`

[3] Oxygen, `http://oxygen.lcs.mit.edu/Overview.html`

[4] Ultra-wideband (uwb) technology, `http://www.intel.com/technology/comms/uwb/`

[5] Agulló, J.: Les rodes omnidireccionals, el darer pas evolutiu de la roda. Memorias de la Real Academia de Ciencias y Artes de Barcelona LXI(7) (2004)

[6] Annicchiarico, R., Barrué, C., Benedico, T., Campana, F., Cortés, U., Martínez, A.B.: The i-walker: an intelligent pedestrian mobility aid. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 708–712 (2008)

[7] Augusto, J.C., Shapiro, D. (eds.): Advances in Ambient Intelligence. IOS Press, Amsterdam (2007)

[8] Intille, S.S.: Designing a home of the future. IEEE Pervasive Computing, 80–86 (April-June 2002)

[9] Kidd, C.D., Orr, R.J., Abowd, G.D., Atkeson, C.G., Essa, I.A., MacIntyre, B., Mynatt, E., Starner, T.E., Newstetter, W.: The aware home: A living laboratory for ubiquitous computing research. In: Streitz, N.A., Hartkopf, V. (eds.) CoBuild 1999. LNCS, vol. 1670. Springer, Heidelberg (1999)

[10] Mandel, C., Huebner, K., Vierhuff, T.: Towards an Autonomous Wheelchair: Cognitive Aspects in Service Robotics. In: Proceedings of Towards Autonomous Robotic Systems (TAROS 2005), pp. 165–172 (2005)

[11] Marin, I., Arceredillo, E., Zuloaga, A., Arias, J.: Wireless sensor networks: A survey on ultra-low power-aware. Transactions on Engineering, Computing and Technology 8, 44–49 (2005)

[12] Pollack, M.E.: Intelligent Technology for an Aging Population: The use of AI to assist elders with cognitive impairment. AI Magazine 26(2), 9–24 (2005)

[13] Ramos, C., Augusto, J.C., Shapiro, D.: Ambient intelligence—the next step for artificial intelligence. IEEE Intelligent Systems 23(2), 15–18 (2008)

[14] Vescovo, F.: Casa Agevole (2005), `http://www.progettarepertutti.org/progettazione/casa-agevole-fondazione/index.html`

[15] Weiser, M.: The computer for the twenty-first century. In: Scientific American, pp. 94–104 (September 1991)

Ambulatory Mobility Characterization Using Body Inertial Systems: An Application to Fall Detection

Marc Torrent[1], Alan Bourke[2], Xavier Parra[1], and Andreu Català[1]

[1] Technical Research Centre for Dependency Care and Autonomous Living – CETpD
Technical University of Catalonia - UPC
Edifici Neàpolis – Rambla de l'Exposició, 59-69
08800 Vilanova i la Geltrú – Catalonia – Spain
{marc.torrent-ponch,xavier.parra,andreu.catala}@upc.edu
[2] Wireless Access Research Centre – Biomedical Electronics Lab
Department of Electronic & Computer Engineering – University of Limerick
Limerick – Ireland
alan.bourke@ul.ie

Abstract. The aim of this paper is to study the use of a prototype of wearable device for long term monitoring of gait and balance using inertial sensors. First, it is focused on the design of the device that can be used all day during the patient daily life activities, because it is small, usable and non invasive. Secondly, we present the system calibration to ensure the quality of the sensors data. Afterwodrs, we focus in the experimental methodology for data harvest from extensive types of falls. Finally a statistical analysis allows us to determine the discriminant information to detect falls.

1 Introduction

In recent years, the progressive aging in our society has produced an increase in the number of dependents or disabled people and at least a growing number of persons that need special attention. In Spain, for example, there are 7.7 million of people older than 65 years. The 32.2 % of them presents some sort of disability, and this percentage is greater as the age range grows [1], [2]. The new requirements arising from this situation require social efforts, but also efforts in science and technology to design together new services and systems in order to improve people's wellbeing and quality of life.

It is a fact that the third part of the population older than 65 years suffers falls each year doing their daily life activities. This falls are one of the major causes of disability and dependency as they can cause major injuries. For instance, the 90% of hip fractures in the elderly are caused by a fall [3]. This is a dramatic factor in disability and mortality as the 33 % of the old people that suffers from a hip fracture dies in the next 12 months, and a 60 % develops some sort of disability that difficult their daily activities and makes their quality of life worse [4].

The early detection of fall risk persons is the most efficient way to not only prevent hip fractures, but also in dependency, institutionalization, and mortality in the elderly. The existent methods for the falling risk estimation are based in interviews with the patient about his fall historical [5], or in an exploration of gait and equilibrium at the

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1129–1136, 2009.
© Springer-Verlag Berlin Heidelberg 2009

laboratory [6]. These methods have numerous disadvantages as they take into account only punctual determinations of the state of the patient and include subjectivity. Thus monitoring the body movements for long periods of time during the patients daily life activities could be a good source of information for a fall risk diagnosis. Moreover, the reliable detection of falls could be crucial in emergency situations.

Inertial sensors such as accelerometers and gyroscopes have proven to be useful in the analysis of human motion [7] with the advantage that they can be integrated in some sort of wearable device. Although many works and research groups related to the area have reached good results, as for example, in the classification of activities [8] , detection of falls [9], [10] and fall risk prediction [11], this is still is an open research subject which needs more robust algorithms, technological improvement with user experience and adaptation and integration in complete ambient intelligent systems.[12]

The aim of this paper is to study the use of a prototype of wearable device for long term monitoring of gait and balance using inertial sensors. First, it is focused on the design of the device that can be used all day during the patient daily life activities, because it is small usable and non invasive. Secondly, we present the system calibration to ensure the quality of the sensors data. Third, we focus in the experimental methodology for data harvest from a extensive types of falls. Finally an statistical analysis allows us to determine the discriminant information to detect falls.

2 Inertial Sensor System

The wearable device, as seen in fig.1) is composed by a central Intelligent Hardware Unit (IHU) and three different inertial sensor technologies. The IHU is the central part of the system, and its function is handling the sensor data. It features a microcontroller based process unit, a communications module, external memory, and interfacing circuitry. The core of the IHU is a dsPIC (Microchip Technology Inc.), a hybrid microcontroller with basic digital signal processor (DSP) features that enables the system with mathematic capabilities. The microcontroller operates at 20 MIPS (Mega Iterations Per Second) and have an internal program memory of 24Kbytes which allows a wide range of programming possibilities. The dsPIC uses different digital interfaces to communicate with the sensors in order to provide robustness to the readings. Another interesting feature of the dsPIC is his low power consumption along with its low power operation modes which enhances battery life.

The device integrates three different MEMS inertial sensors which can measure 3D linear acceleration, angular rate and compass orientation. The accelerometer is a three-axis sensor model LIS3LV02 (ST Microelectronics Inc.) with a measuring range of ±6 g and a sensitivity of 323 bit/g. The sensors have a built-in digital circuitry that uses to apply a first low pass filter to the data captured and to transmit it using a digital I^2C interface. The three-axis gyroscope is constructed by assembling two different MEMS gyroscopes, the one-axis ADIS16100 (Analog Devices Inc.), and the two-axis IDG300 (Invensense Inc.). The two sensors have similar characteristics and work well in conjunction, with a typical range of ±300 deg./s. and a typical sensitivity of 4 bit/deg/s. The analog output signals from the IDG300, are redirected to the auxiliary inputs of the ADIS16100 which has a digital interface used to send the three-axis signals digitally to the IHU. The magnetometer or electronic compass is a three-axis sensor which can determine the absolute orientation of the device using the earth magnetic field as reference. The sensor has a resolution of 0.2 degrees of deviation and is very sensible to tiny variations in the magnetic field in its surroundings. For this reason its application is reduced to controlled scenarios.

The device can operate in two different modes: online or offline. In online mode the sensors readings are transmitted in near real time via Bluetooth radio. It integrates a Bluetooth transceiver Parani- ESD200 (Sena Technologies Inc.) capable to establish a wireless communication with another device using the RFCOMM protocol (virtual serial RS232 connection). This is a much extended protocol, so it enables the system to communicate with a large range of devices including laptops, handheld devices and mobile phones. The alternative is the offline mode where the sensors readings together with a time stamp are stored in an external memory. The memory used is a micro SD flash card of 1 or 2 Gbytes capacity. The main features of this memory are its reduced size, high capacity, low power consumption and the possibility to use a standard SPI interface to communicate with de microcontroller.

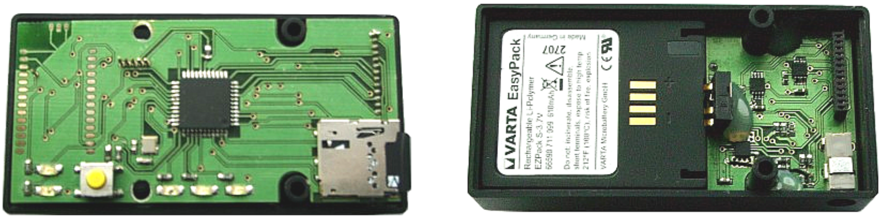


Fig. 1. Inertial system

The device, including the sensors, is powered by a single Lithium Polymer rechargeable battery with a capacity of 610 mAh. With this battery the system can run for about 5 hours without interruption. All the electrical interface circuits including the battery charger, battery monitor, battery protection and voltage regulation are integrated in the device. The size of the sensor system with the battery and the casing included is 78x40x22 mm. and has an approximate weight of 30 gr.

3 Inertial System Calibration

In order to extract useful information from the designed system it is necessary to calibrate the inertial sensors, to translate the output electrical signals to understandable physical measurements. This calibration process has to be done with each single sensor, correcting its main sources of error such as drift, sensitivity error, misalignment of the axes and other errors due to environmental factors like temperature. This calibration process in inertial sensors is complex and normally requires the use of expensive precision instruments. An alternative is the simple but systematic procedure designed by Ferrari et al. [13], for accelerometers and gyroscopes and the one designed by O'Donovan [14] for the calibration of the magnetometer.

These procedures provide correction for the major sources of error in inertial sensors: these are the drift, the sensitivity and the axis misalignment. They also have the advantage of not needing any special equipment to carry them out, as they use physical phenomenon's (i.e. gravity, earth magnetic field) as references for the calibration.

3.1 Combined Sensor Calibration

The calibration processes for the three sensors can be joined in a single process that involves putting the sensor in 12 predefined static positions and 3 full revolution rotations, reducing the calibration time. This calibration process, once developed, is also suitable for its integration in the on board device controller, as it only involves sum and matrix multiplication for each sensor.

3.2 Temperature Compensation

The calibration of the sensors is hampered by one main issue; it only takes into account the internal sensor factors but not the external ones. The main external factor that affects inertial sensors is the temperature. Only a few degrees of temperature variation can modify the sensor parameters introducing errors in the readings that cannot be neglected.

In applications such as subject movement monitoring, great temperature gradients are not to be expected, and as the sensor would be directly attached to the patient's body, one can expect constant temperatures between 20°C and 40°C. For this reason, a linear approximation can be a good solution for this kind of systems. The effectiveness of this approximation has been proved experimentally with all the sensors and with a large number of trials, as can be seen in figure 2.

This method provides a long term calibration valid for a range of temperatures, so it is not longer necessary to carry out a calibration each time the sensor has to be used. This procedure can be also embedded in the on-board controller in devices that have a temperature sensor.

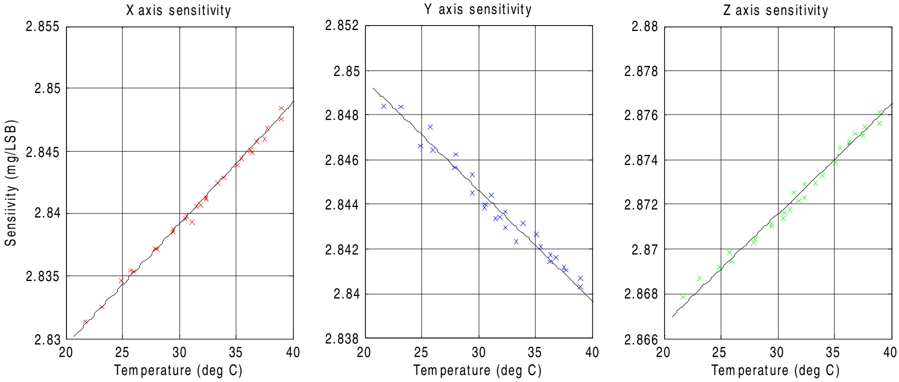


Fig. 2. Accelerometer sensitivity variation over temperature

4 The Simulated Fall Study

The fall study involved 16 healthy volunteers performing falls over large crash mats for two different scenarios and performing three repetitions each exercise. Signals from the three sensors in the device were recorded during each fall-event with the device located at the chest. These subjects ranged in age from 24-35 years old, with a body mass from 68 to 111 kg, and a height from 1.65 to 1.96m. The University of Limerick Research Ethics Committee (ULREC) approved the falls protocol.

The fall types used during testing for the current study were selected to simulate common fall types in elderly people The falls performed were similar to those performed in the study of Bourke et al. [15]. The different types of falls are the following:

Forward falls

1. Faint with flexed knees.
2. Step down from a platform and fall forward. Thick (>15cm) soft mat on floor.
3. Walking and trip yourself up. Thick (>15cm) soft mat on floor.

Backward falls

4. Faint with Round back and flexed knees.
5. Backward Sitting down on empty (on the floor without using your arms or taking a step back). Thick (>15cm) soft mat on floor.
6. Backward to the base of a padded wall. Thick (>15cm) soft mat on wall and floor.

Side falls left

7. Faint with flexed knees.
8. Fall backward and turn to the side.
9. Side fall to the base of a padded wall. Thick (>15cm) soft mat on wall.

Side falls right

10. Faint with flexed knees.
11. Fall backward and turn to the side .
12. Side fall to the base of a padded wall. Thick (>15cm) soft mat on wall.

Free falls

13. Falling off a chair.
14. Unrestricted activities (fall-over as you like).

In order to compare the falls data with other movements we inquire to the same group of young healthy subjects performing Activities of Daily Living while fitted with the same sensor as the simulated-fall study, performing three repetitions. Each subject started and finished each ADL in a standing position. The ADL chosen were those that may have produced impacts or abrupt changes in a person's movement (and thus possibly results in false triggering of a threshold-based fall-detection algorithm) resembling activities carried out during the normal course of an elderly person's daily life.

Activity tests

15. Walking 10 meters.
16. Standing position to a sitting position in a normal chair. (height 43cm).

17. Standing position to a sitting position in a bank. (height 44.5cm).
18. Standing position to a lying on a mat or bed. (height 40cm).

In one month we achieved to compile a signal database of 636 data sets, 468 of which are falls in a large diversity of scenarios. This rich database is used now to train different algorithms for the movement/fall identification problem. Also using two different sensor technologies during the same experiments can give us the opportunity to compare the differences between sensors on different body locations.

5 Fall Detection Algorithm

The data recorded from the three-axis accelerometer was derived by taking the equation norm to calculate the module. With this signals and looking at the exact moment of the fall, an extraction of characteristics was done. The characteristics analyzed were the following:

a) *Upper fall threshold*. This threshold is related to the peak of acceleration produced during the impact with the floor in a fall. This acceleration peak can be very high in direct falls to the floor, but it is lower in faints or falls to a wall. So it is not possible to distinguish completely a fall with only this characteristic because of the acceleration peaks produced in some ADLs.
b) *Lower fall threshold*. There is always a minimum peak related to the bounce occurred after the fall
c) *Time between upper and lower fall threshold.*
d) *Difference between upper and lower threshold.*
e) *Maximum angular velocity peak.*

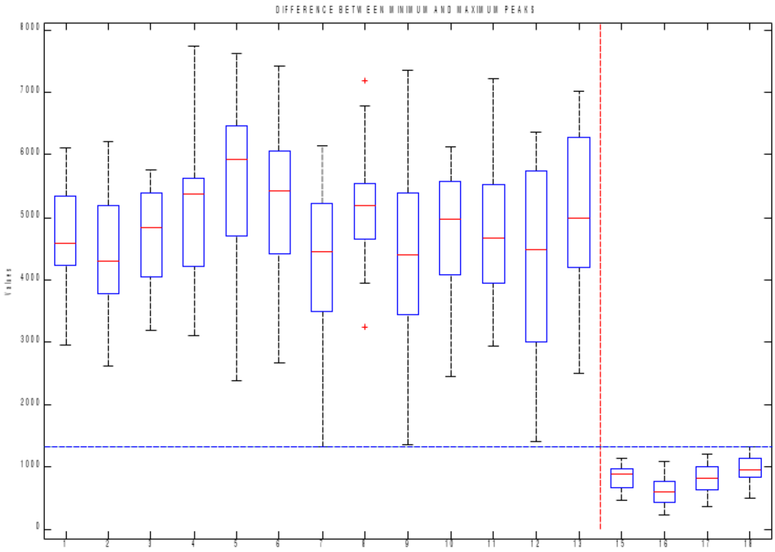


Fig. 3. Box Plot of the difference between max. and min. acceleration peaks

The analysis of these characteristics with a Box Plot with Median, 25 and 75 Percentile and outlayers is ploted in Fig 3 for the difference between upper and lower threshold. This characteristic is the most discriminant and allows us to define a threshold between falls (1-13) and DAL (15-18)

6 Conclusions and Future Work

An ambulatory inertial system for mobility analysis is presented. The first results show that the system could be useful for fall detection. Further work is needed to develop algorithms for activity identification and to embed it into the on-board controller of the device.

Acknowledgment

This work is partially supported by the Spanish MEC project ADA (DPI2006-15630-C02-01) and the European Community project CAALYX (IST-STREP-045215).

References

1. Las personas mayores en España. Informe 2004, IMSERSO, Madrid (2004)
2. Libro blanco de la dependencia. Ministerio de trabajo y asuntos sociales, Madrid, Ref Type: Catalog (2005)
3. Tinetti, M.E.: Clinical Practice, Preventing falls in elderly persons. Journal of Medicine 348(1), 42–49 (2003)
4. Hannan, E.L., Magaziner, J., Wang, J.J., Eastwood, E.A., Silberzweig, S.B., Gilbert, M., et al.: Mortality and locomotion 6 months after hospitalization for hip fracture: risk factors and risk-adjusted hospital outcomes. J. of the American Medical Association 285(21), 2736–2742 (2001)
5. Ganz, D.A., Higashi, T., Rubenstein, L.Z.: Monitoring falls in cohort studies of community-dwelling older people: effect of the recall interval. J. of the American Geriatrics Society 53(12), 2190–2194 (2005)
6. Brauer, S.G., Burns, Y.R., Galley, P.: A prospective study of laboratory and clinical measures of postural stability to predict community-dwelling fallers. J. of Gerontology: Medical Sciences 55(8), 469–476 (2000)
7. Mathie, M., Coster, A.C.F., Lovell, N.H., Celler, B.G.: Accelerometry: providing an integrated, practical method for longterm, ambulatory monitoring of human movement. Physiol. Measure. 25(2), R1–R20 (2004)
8. Pärkkä, J., Ermes, M., Korpipää, P., Mäntyjärvi, J., Peltola, J., Korhonen, I.: Activity classification using realistic data from wearable sensors. IEEE Trans. on Information Technology in Biomedicine 10(1) (January 2006)
9. Lindemann, U., Hock, A., Stuber, M., Keck, W., Becker, C.: Evaluation of a fall detector based on accelerometers: a pilot study. Med. Biol. Eng. Comput. 43, 548–551 (2005)
10. Noury, N., Fleury, A., Rumeau, P., Bourke, A.K., Laighin, G.Ó., Rialle, V., Lundy, J.E.: Fall detection – Principles and Methods. In: Proceedings of the 29th Ann. Int. Conf. of the IEEE EMBS, August 2007, pp. 1663–1666 (2007)

11. Giansanti, D.: Investigation of fall-risk using a wearable device with accelerometers and rate gyroscopes. Physiol. Meas. 27(11), 1081–1090 (2006)
12. Molinero, A.R., Catala, A., Díaz, M., Caalyx: Evidence-based selection of helth sensors for eldery telemonitoring. International Society of Gerontechnology (2008)
13. Ferraris, F., Grimaldi, U., Parvis, M.: Procedure for effortless in-field calibration of three-axis rate gyros and accelerometers. Sens. Mater. 7, 311–330 (1995)
14. O'Donovan, K.J., O'Keeffe, D.T., Lyons, G.M.: A Technique for the Calibration of a Miniature Tri-Axial Magnetometer for Kinematic Analysis Applications, Ph.D. Thesis, ch. 5
15. Bourke, A.K., O'brien, J.V., Lyons, G.M.: Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait and Posture (2006)

User Daily Activity Classification from Accelerometry Using Feature Selection and SVM

Jordi Parera[1], Cecilio Angulo[1], A. Rodríguez-Molinero[2],
and Joan Cabestany[1,⋆]

[1] UPC. Technical University of Catalonia - CETpD. Technical Research
Centre for Dependency Care and Autonomous Living
Neàpolis. Rambla de l'Exposició, 59-69. 08800 Vilanova i la Geltrú - Spain
[2] FHCSAA. Sant Antoni Abad Hospital - CETpD. Mobility and Gait Lab
Sant Josep, 21-23. 08800 Vilanova i la Geltrú - Spain

Abstract. User daily activity monitoring is useful for physicians in geriatrics and rehabilitation as a indicator of user health and mobility. Real time activities recognition by means of a processing node including a triaxial accelerometer sensor situated in the user's chest is the main goal for the presented experimental work. A two-phases procedure implementing features extraction from the raw signal and SVM-based classification has been designed for real time monitoring. The designed procedure showed an overall accuracy of 92% when recogninzing experimentation performed in daily conditions.

1 Introduction

One of the consequences of chronic diseases and strokes is the limitation of the motion capacity and a straightforward lack of physical activity of the people, having a direct impact on their quality of life. By analyzing user daily activity, medical treatments would count with valuable additional information, allowing a better diagnose and treatment assessment. Usual instruments to supervise patient's mobility are based on the subjective perceptions of the observer, collected throught individual tests. However acceleration-based activity monitoring wearable systems are currently being investigated as a right direction to overcome subjectiveness and to provide a compactne and robust technical solution.

Previous studies on ambulatory activity monitoring rely either, in multiple body fixed sensor configurations or/and off line data processing, when only one sensor is used for activity identification. The purpose of this presented research is to implement real time events' classification of user's daily life activity by means of a Support Vector Machine based on features processed from signals provided by one small-sized low power consumtion wearable sensor module, so the system can be used everywhere during user daily activity without any external infrastructure

---

[⋆] This work has been partly supported by the Spanish MEC project ADA-EXODUS (DPI2006-15630-C02-01) and the FP6 EU project CAALYX (IST-2005-045215). First author is supported by an UPC research grant. The authors would like thank Carlos Pérez-López, and Marc Torrent-Poch at CETpD for their technical support.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1137–1144, 2009.
© Springer-Verlag Berlin Heidelberg 2009

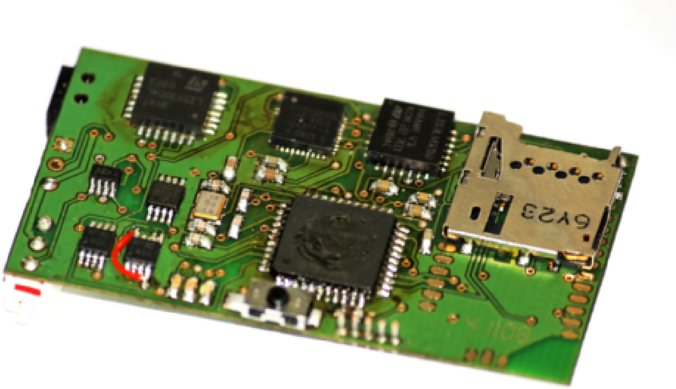


Fig. 1. 3D axis accelerometer sensor

needed. The device holds a battery, a triaxial accelerometer, all the necessary electronic components and a low power DSPic microprocessor, in Fig. 1 there is a photo of this measuring system. By implementing SVM-based real time activity labeling instead of simple acceleration logging, a possible interaction with the user is enabled, for drug taking alerts for instace, and the ammount of data used to store the user activity will also be reduced which will increase the device's operating time. The rest of this paper is organized as follows: In section 2 the problem is formulated. Section 3 presents the proposed feature selection strategy based on discrimiant power for generated features. Section 4 is devoted to the experiment design for validating the performance of the procedure. Finally, conclusions and further research are given in Section 5.

2 Problem Formulation

2.1 Input Signals

Input signals generating raw data are obtained from the triaxial accelerometer of the sensor device located on the user's chest sampled at 50 Hz. Activities to be classified last from 0.5 seconds, for walk activity (steps), to 2.4 seconds, for the stand up activity, hence the input data at the processing block is windowed into a 3-column matrix with 120 samples. Data will be processed every half time window, so the classifier will identify which activity is being performed with a delay of 1.2 seconds.

2.2 User Daily Activities

The final objective in our research is to identify a set of five usual daily activities performed by the user while wearing the sensor device. The activities to be classified are:

1. Standing up from a sit state. This action lasts for 1-2.5 seconds with differentiated phases: forward bending, active raising, passive raising and downward bending. The timing and magnitude between these phases de termine many pathologic characteristics.

2. Sitting down. This action also ranges 1-2.5 seconds. It is similar as an inverse of standing up, these pairs of signals being the most similar in the group of activities.
3. Transition action. Since the sit down and stand up actions are very similar, a control variable has been defined that improves the final classification by avoiding signals that are apart of the stand or sit signals. It is defined as an action occurring just before (after) a sit down (stand up) action.
4. Walk activity. The duration of a step is variable between 0.5 and 1.5 seconds, hence the classifier is trained to search for a complete walking action instead of focusing on individual steps.
5. Steady stand activity. This signal has no significant peaks or timing, but is not constant. The sitting steady state gives the same signal than the steady stand state, so we must detect before the sit-stand transition to correctly label the activity.

2.3 Signal Processing

Preprocessing raw data provided by accelerometer will help to avoid overfitting and generalizing the solution. Reducing the input dimension of data from a 3-column matrix with 120 samples to a vector of 12 features, this reduction will also speed up the training and classification procedures, but it will add the cost of preprocessing. Main features extracted from the triaxial accelerometer data are the following,

- Computing the modulus of the vector components is a simple solution to reduce the data dimension by 3 and also it makes the data orientation independent [1]:

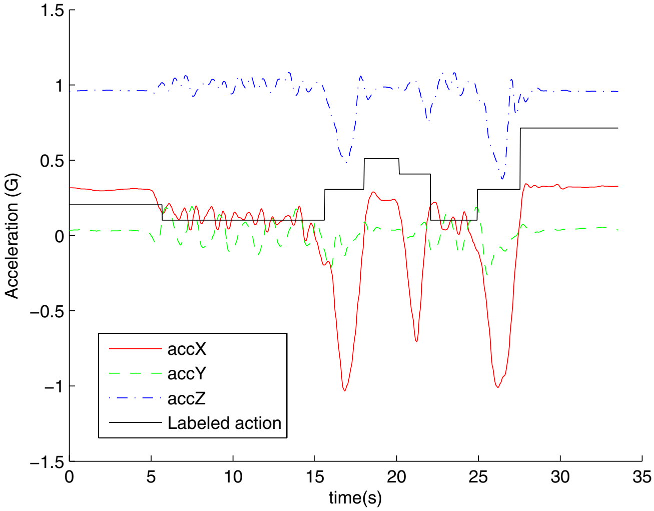


Fig. 2. 3 axis Acceleration signal and labeled action

$$AM = \sqrt{AX^2 + AY^2 + AZ^2} \tag{1}$$

- Orientation angles θ and ϕ based on the earth gravity allows to calculate the orientation of the sensor device. This method works fine in static movement conditions but also low centripetal accelerations don't affect the result [2]. Impacts or large accelerated movements will, however, add error:

$$\theta = \arctan 2(X, \sqrt{Y^2 + Z^2}) \tag{2}$$

$$\phi = \arctan 2(AY, AZ) \tag{3}$$

- Vertical (AV) and forward (AF) acceleration: once the orientation angles are known, the accelerations in the earth fixed reference frame can be computed from the mobile reference frame, that is, the wearable accelerometer, by applying the rotation matrix of the $X-$ and $Y-$ axis. These features are interesting because they give a value that is robust to the orientations of the measuring device:

$$AV = -\cos(\varPhi) \cdot \sin(\varTheta) \cdot AX + \sin(\varPhi) \cdot AY + \cos(\varTheta) \cdot (AZ + G) \tag{4}$$

$$AF = \cos(\varTheta) \cdot AX + \sin(\varTheta) \cdot AZ \tag{5}$$

- Energy expenditure indicators using the acceleration signals are used to calculate the integral of absolute value (IAA) and the integral of magnitude (IAV)[1]:

$$IAA = \int |AX| + |AY| + |AZ| dt \tag{6}$$

$$IAV = \int \sqrt{AX^2 + AY^2 + AZ^2} dt \tag{7}$$

- The increments of the acceleration module.
- Frequency features can be obtained by performing the Fast Fourier Transform (FFT) on the accelerometer signals either, on each axis or on the acceleration magnitude, returning the main frequency components and magnitudes. However, calculating these features demand a high processing time, so they should be used only if proven effective.

3 Feature Selection

Combining both, precedent and standard statitiscal features (mean, max, min, range, standard deviation, entropy...), a huge set of possible features can be obtained that have been proved useful in other similar SVM-based articles [4]. These features gives relevant information about the signal as well as they enable

---

[1] IAV has been identified as less accurate than IAA [3].

to process the input stream of data in a batch process every half-size input window and not every sampled acceleration.

The method used to incrementally select features is the discriminant power measured as the distance between the group of data obtained from observations when performing a certain activity and the other groups of observations when doing other activities. If different activities are represented by non overlapping ranges of values then this feature can classify a number of activities by itself. In order to add robustness, we will select several features that can identify different activities. Since the features have many different units they are normalized so the relevance and performance of each feature can be compared [5].

The grouping method for data obtained from different observations of the same activity can be the mean value or an interval of values. The indicators used to define intervals of data are percentiles generating boxplots and whiskers (see Fig. 3. The bounds for the boxplot will be the 25% and the 75% percentile. Whiskers used to bound the intervals are defined as the most extreme data value within $1.5 \cdot IQR$, where IQR is the interquartile range of the sample.

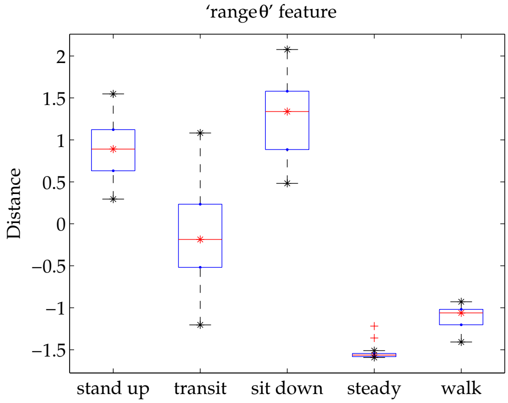


Fig. 3. Intervals for the 5 activities based on 'range θ' feature

Once the intervals and boxplots of each activity on each designed feature are known, the distance between intervals is measured. In the case of not overlapped intervals the distance is positive and it means that this feature can be used to classify the pair of activities. The distance between the different activities is computed and stored in a matrix that records the distance from one activity from the others. The number of positive distances, or the number of distances that are greater than a certain threshold in each row is the number of classes that are different from the other classes in that particular feature. We tested 3 indicators to select the best discriminant set of features:

1. The number of classes that a single feature classifies, i.e. the number of positive values of the distance matrix. Since 5 classes are considered, the

5×5 distance matrix maximum value is 10[1]. For our collected data set the maximum value obtained for a feature was 9, so not a single feature exist that can classify all the 5 classes.

2. The sum of distances between classes: this is a value that represents how a feature is different for each class. Since the features are normalized and considering negative values for overlapping activities, values from -20 to $+20$ can be expected.
3. The sum of positive distances between classes: this is a value that represents how a feature is different in each class when intervals do not overlap. We can expect values from 0 to $+20$.

4 Experimentation and Results

The exercise performed by the test subjects to collect data was: to stay steady in vertical position, then walk about 4 meters, sit down on a chair, stay sitted for a few seconds, next stand up, walk, and finally sitting down again and staying steady again. Each test subject repeats the exercise 3 times. At the figure 2 there is the raw acceleration signal(converted to G unit) of a single ecercise, thre is also the labeled action as a solid black line. Experiment were video recorded to enable the labeling of activities. The test group consists on 6 healthy subjects with no mobility limitations with mean age 38.17 and standard deviation 12.6.

Using the three previously defined indicators, features were ordered but the length of the feature vector was left variable in order to check the accuracy of the method with different lengths of the feature vector. Five one-versus-rest SVM were trained for the multiclassification problem with Gaussian kernel and inputs were assigned to the class with a higher number of votes. When no positive votes exist, then no label is assigned. A 10-fold cross-validation method with 3 repetitions was performed for each length of the feature vector, ranging from 1 to 30 features, and accuracy was defined as the best value between the 3 defined indicators for each input vector's length. The Fig.4 shows the best performance result (the mean of the 3 repetitions of the 10-fold test) using any of the 3 methods of feature selection versus the feature vector length. In the table 1 there is the result of the classification in the form of a confusion matrix, where the rows represent the activity data and the columns represent wich percentage the current activity has been labeled. The sum of the elements of a row doesn't have to sum 100% because the resulting vote of the 5 SVM can be negative so no activity is identified. A 10% of stand up actions are labeled as transition but this wrong labeling is not a hard mistake because transition is a state that happens just before or after stand up or sit down actions, hence this label still provides information whether we know which was the action before a false transition is labeled.

---

[1] th distance matrix is simetric and diagonal of zeros so it can be cosidered only the lower triangle, hence the max distance value is 10.

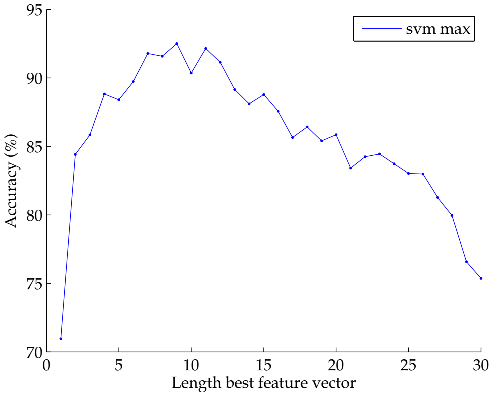


Fig. 4. Accuracy results versus number of features used

Table 1. Confusion Matrix (%)

Label	stand up	sit down	steady	walk	trancision
stand up	75.56	14.44	0	0	10
sit down	6.06	89.09	0	0	4.85
steady	0	0	93.33	6.67	0
walk	0	0	2.22	95.56	2.22
trancision	1.39	1.39	0	3.06	94.17

The best accuracy, 91.06%, is achieved when the length of the feature vector is 7 and the selected features are: ‘STD AX’,‘‘min Ax’, ‘STD AV’, ‘STD AF’, ‘min AF’,‘STD θ’ and ‘range θ’. Calculating standard deviation features (STD) requires a relative hard processing time effort. Lower, but not bad, accuracy results can be achieved using features easier to be calculated. A 90.03% accuracy is obtained with a vector having length 13.

5 Conclusions and Further Work

A two-phases procedure implementing (i) features extraction from the raw signal of a triaxial accelerometer sensor situated in the user chest based on a distances matrix and (ii) SVM classification has been designed for a real time monitoring problem. The designed algorithm showed an overall accuracy of 91% when recogninzing five different activities during experimentation performed in daily conditions.

Further research includes verifying the performance of the classifier on very long time experimentations and increase the number of daily activities to be identified by the system.

References

1. Bidargaddi, N., Sarela, A., Klingbeil, L., Karunanithi, M.: Detecting walking activity in cardiac rehabilitation by using accelerometer, December 2007, pp. 555–560 (2007)
2. Giansanti, D.: Does centripetal acceleration affect trunk flexion monitoring by means of accelerometers? Physiological Measurement 27(10), 999–1008 (2006)
3. Bouten, C.V.C., Koekkoek, K.T.M., Verduin, M., Kodde, R., Janssen, J.D.: A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. IEEE Transactions on Biomedical Engineering 44(3), 136–147 (1997)
4. Begg, R.K., Palaniswami, M., Owen, B.: Support vector machines for automated gait classification. IEEE Transactions on Biomedical Engineering 52(5), 828–838 (2005)
5. Isabelle, G., André, E.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

A Metrics Review for Performance Evaluation on Assisted Wheelchair Navigation

Cristina Urdiales[1], Jose M. Peula[1], Ulises Cortés[2], Christian Barrué[2],
Blanca Fernández-Espejo[1], Roberta Annichiaricco[3], Francisco Sandoval[1],
and Carlo Caltagirone[3]

[1] Dpto. Tecnología Electrónica, University of Málaga, Málaga, Spain
`acurdiales@uma.es`
[2] Software Department, Technical University of Catalonia, Barcelona, Spain
[3] Fondazione Santa Lucia, Rome, Italia

Abstract. In nowadays aging society, many people require assistance for non-pedestrian mobility. In some cases, assistive devices require a certain degree of autonomy when the persons' disabilities difficult manual control. In this field, it is important to rate user performance to check how much help he/she needs. This paper presents an overview of common metrics for wheelchair navigation, plus some proposed by authors to take into account new approaches to wheelchair control. We present an example of proposed metrics on a robotized Meyra wheelchair at Santa Lucia Hospedale in Rome to prove their meaning and adequacy.

1 Introduction

Autonomy in an agent can be defined as the ability to make independent choices. A mobile is considered to be autonomous when it can perform a given task in a dynamic environment without continuous human guidance. Many elder individuals are unable to safely and independently operate a standard power wheelchair due to physical perceptual or cognitive deficits. In these cases, robotics provide the meaning to assist these persons in navigation.

Depending on how much autonomy is given to the wheelchair, collaborative wheelchair assistance can be categorized into i) safeguarded operation; and ii) shared control.

In the first case, wheelchairs are totally controlled by humans, but in some cases the robot makes some decisions to avoid imminent danger when communication interruption and delays are frequent [1] or when human control is not adequate [2] [3].

A second group of approaches [4] [5] rely on using a basic set of primitive actions like AvoidObstacle, FollowWall and PassDoorway to assist the person in difficult maneuvers, either by manual selection or automatic triggering. Hence, the operator may guide the robot directly, or switch among various autonomous behaviors to deal with complex situations.

In an extreme case, as commented before, the human operator might only point the target and the machine would be in charge of motion planning and

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1145–1152, 2009.
© Springer-Verlag Berlin Heidelberg 2009

path tracking on its own [6] [7]. In most cases though, at any point, the user may override automatic control and take over.

This paper reviews the different metrics that have been applied to wheelchair navigation in section 2. To prove the validity of these metrics, some of them are applied for assisted power wheelchair navigation in a real experiment in Fondazione Santa Lucia (FSL) in section 3. Finally, conclusions and results are provided in section 4.

2 Assisted Navigation Metrics

There is no established standard to measure wheelchair performance, but there is a large number of proposals that are more or less coincident regarding parameters of interest in assisted navigation. This section presents an overview of such parameters and has been organized as follows. First of all, performance is interdependent on the wheelchair user condition, which is, indeed, hard to rate. Nevertheless, there are some studies to measure the degree of disability of a person (subsection 2.1). Person skills while driving any wheelchair -electric or not- can also be evaluated, usually through obstacle courses (subsection 2.2). Finally, subsection 2.3 overviews the different metrics that can be applied to conventional and/or electrical wheelchair navigation.

2.1 User Condition Metrics

In order to further evaluate the target population, it is necessary to note that disability is usually defined as the degree of difficulty or inability to independently perform basic activities of daily living (ADL) [8] without assistance. Several different disability scales have been used to evaluate the cognitive and physical state and condition of the in-patients, namely the Mini-Mental State Examination (MMSE), Geriatric Depression Scale (GDS) and Barthel Index.

The mini-mental state examination (MMSE) or Folstein test [9] is a brief 30-point questionnaire test that provides a quantitative measure of cognitive status in adults. It can be used to screen for cognitive impairment. Although results may change slightly depending on the number of question, 24 is usually the accepted threshold for dementia.

The Geriatric Depression Scale (GDS) [10] is a self-report inventory, constructed to assess depression and general well-being in the elderly. It is a brief questionnaire of 30 yes-no questions about how the user feels on the day of administration. Scores of 0 - 9 are considered normal, 10 - 19 indicate mild depression and 20 - 30 indicate severe depression.

The Barthel Index [11] consists of 10 items that measure a person's daily functioning specifically the activities of daily living and mobility. The items can be divided into a group that is related to self-care (feeding, grooming, bathing, dressing...) and a group related to mobility (ambulation, transfers...). The maximal score is 100 if 5-point increments are used, indicating that the patient is fully independent in physical functioning. The lowest score is 0, representing a totally dependent bedridden state.

2.2 Test Courses

In [12] Kilkens et al. present a review of a wide array of wheelchair skill tests, aimed to asses the ability to propel and maneuver a wheelchair under standardized and/or simulated conditions of daily living. The skills included in the 24 tests, related to factors like wheelchair propulsion, negotiation of kerbs, etc. The most common outcome measure of the different tests is *task performance time. Physical strain*, *independence in performance*, *distance covered in propulsion*, and *endurance.* These metrics can be applied mostly to obstacle courses but usually not to daily living, as they are measured over a given time and trajectory.

2.3 Assisted Navigation Metrics

After a test course is established, there are well established metrics for manual wheelchair driving (e.g. [13]). However, there are not so many standards for power wheelchair navigation. There seems to be an agreement[14], though, to distinguish between task metrics (goal achievement features, like error rate with respect to an unique, prefixed trajectory) and psycho metrics (measure parameters like fatigue and attention from the user point of view).

2.4 Task Metrics

Task metrics include all quantitative measures that can be objectively related to the performance of the user with the wheelchair during a given task. The most common and easiest metric concerning navigation is *task success*, meaning that the goal has been achieved. Other popular metrics include *task completion time*, *number of collisions* and *distance travelled* [15] . When tests consist of following a prefixed path a typical task metric is the *deviation with respect to the canonical trajectory.* Such a deviation can be measured in a straight way or, more often, quantified into a few bins as “ok”, “acceptable” and “wrong”.

Local metrics representing performance at a given time instant are also feasible: driving efficiency can basically be measured in terms of *safety*, *directiveness* and *smoothness*, as proposed in [16]. If the aforementioned parameters are averaged all through a trajectory, they yield global information instead of local one.

If dynamic obstacles may appear on the way, four difference metrics have been established: T_s, D_s, T_c and D_c, which correspond to *Time* and *Distance to Stop*, when the obstacle is not hit and *Time* and *Distance to Collision*, otherwise, all referred to the instant the obstacle is introduced [15].

2.5 Psycho Metrics

Psycho metrics are related to the state of the user rather than to his/her performance. From this point of view, psycho metrics could be estimated via biometric sensors. From this point of view, typical metrics related to stress include *heart rate*, *sweat* and *blood pressure.* If no biometric sensors are used psycho metrics are typically related to the user interface. One of the first was the *interaction number* or number of times that the user interacts with the interface [15]. When interaction was not easy to measure externally, some authors used the *amount of information exchanged* between user and machine instead. It is interesting to measure workload instead of usability, because it is more coupled with the user condition than with the interface. Some authors [17] consider workload to be related with a very simple parameter: *Joystick Variation*, which measures a change of more than 10% in the position of the stick. It is important to note, though, that in users operating a wheelchair, this variation may be also related to spasmodic joystick movements. Similarly, given the target population of wheelchair navigation, it is also important to take into account *Consistency*, defined as the variation of the user output when facing similar situations. A high Consistency is expected to be related to users with good cognitive capabilities, whereas a low one is related to random joystick motion. Thus, consistency provides an idea about the cognitive driving capabilities of the user on-line. Another typical parameter is *Intervention level*, defined as the portion of time that the user moves a joystick [14]. In shared control, it is also interesting to check if machine and person agree via a parameter named *Disagreement*, as proposed in [16], which represents the difference between the human output and the robot output. A high Disagreement is expected to be related with effort and frustration.

2.6 Subjective Questionnaires

It has been reported that subjective ratings may come closest to tapping the essence of mental workload and provide the most generally valid, sensitive and practically useful indicator.

In order to measure workload, the NASA proposed the known as *Task Load Index (TLX)* [18]. TLX was developed under the assumption that workload emerges from the interaction between the requirements of a task, the circumstances under which it is performed, and the skills, behaviors, and perceptions of the operator rather than from interaction with an interface alone. TLX is generic for any task and can be applied to supervised navigation and, in our case, shared control. TLX consists of different component scales. An average of these scales, weighted to reflect the contribution of each factor to the workload of a specific activity from the perspective of the rater, is proposed as an integrated measure of overall workload. The most salient factors in TLX include overall workload (low-high), task difficulty (low-high), time pressure (none-rushed), performance (failure-perfect), mental/sensory effort (none-impossible), physical effort (none-impossible), frustration level (fulfilled-exasperated), stress level (relaxed-tense), fatigue (exhausted-alert), activity type (skill based/rule based/ knowledge based). These factors are ranged within a 12 cm line which is later numbered from 0 to 100 and subjectively marked by each user.

There are more specific questionnaires for assistive systems, though. One of the best known ones is *PIADS (Psychosocial Impact of Assistive Devices Scale)*[19], which is available free of charge for use in clinical, research or educational projects. PIADS can be filled in 5 to 10 minutes and includes ratings for self-esteem, well-being, quality of life, embarrassment, eagerness to try new things and up to 26 other subjective factors scoring from -3 to 3. PIADS is reported to be very sensitive to differences among device options (e.g., tilt, recline) within a device category (e.g., powered wheelchairs).

The Matching Person and Technology (MPT) [20] sheet is also a well known tool to assess how assistive technology is perceived from the user's point of view. MPT evaluates people's condition (hearing, sight, mobility ...), activities of daily living (writing, reading, domestic tasks...) and cognition/learning. All these factors are evaluated regarding previously used technology, currently used technology and required technology. After this, MPT moves to the person's opinion on technology, experience with it, frequent activity and such, which are labelled as positive/indifferent/negative. There is a large part of the questionnaire that must be filled not by the user, but by the person supervising the process, instead.

Functioning everyday with a wheelchair (FEW) [21] is meant to be a dynamic indicator or profile of perceived user function related to wheelchair/scooter use. FEW is mostly concerned with the size, fit, postural support and functional features of wheelchairs (or scooters). FEW provides 10 assessments that the user may score into 6 bins ranging from *Completely agree* to *Completely Disagree*, plus one for *Does not Apply.*

Another similar questionnaire is *QUEST (Quebec User Evaluation of Satisfaction with assistive Technology)* [22]. QUEST focuses on the satisfaction of the user with the provided assistive technology and/or services. It tests psychometric properties with respect to test-retest stability, alternate-form equivalence, internal consistency, factorial composition and nomological validity. QUEST presents 12 items that score from 1(not satisfied at all) to 5 (very satisfied), going from physical considerations like weight or dimensions of the device to service delivery or repair and servicing. It also includes subjective considerations about the ease of use or effectiveness.

3 Tests and Experiments

Some of the presented metrics were applied to a set of experiments performed during July, 2007 in Santa Lucia Hospedale, highly specialized in rehabilitation. All tests were performed using a shared control approach presented by the authors in [23], which calculates the commands sent to the wheelchair as a weighted average between the user's command and a potential field approach. The weight in this combination depends on the local efficiency of each of them.

Specifically, we are going to compare the performance of two in-patients 19 and 31 when crossing a door 90 cms. wide in front of them and then turning right to face a corridor, a typical activity in an indoor environment. They only know the goal to reach. Only a selection of the presented metrics were used for several reasons: i) some are redundant or similar (for example, PIADS, FEW or QUEST measure the impact of technology in the user); ii) some are not applicable to the problem; and iii) some were not approved for these in-patients by the ethical committee of FSL (for example collisions and biometric sensing).

The metrics we have chosen are: i) Smoothness, safety and directiveness efficiency, and disagreement: These metrics estimate the efficiency of robot and human movements and how much help the chair gives to the user, plus how much they agree on the output command (measure of frustration). ii) Curvature and time to Destination: estimates the user time reduction when repeating the test and the smoothness of the path. iii) Intervention level and joystick variation: measure how much interaction has the user with the wheelchair. iv) Inconsistency: let us to estimate if the user was erratic or not while moving the wheelchair in similar situations. v) PIADS questionnaire. The most popular one of the questionaires.

In-patients 19 and 31 have been chosen because their disability indexes were quite different . According to MMSE, in-patient 19 scores for dementia, whereas in-patient 31 does not (28 and 19 respectively). Regarding Barthel, in-patient 19 has better motory abilities than in-patient 31 (98 and 76 respectively), even though both in-patients are autonomous according to questionnaire PIADS. In brief, and always regarding these indexes, in-patient 19 should score better for physical activity and worse for cognitive indexes.

Table 1 show the task and psycho metrics for in patients 19 and 31. Despite their condition differences, their task metrics are quite similar given the type of shared control in the experiments, where lacks in motion control tend to be compensated in real time by the machine. In detail, though, we could observe that in-patient 31 is slightly better in average than in-patient 19, though also presents a slightly larger deviation in performance. This is expectable in a patient presenting physical, but not significant cognitive problems when navigation assistance is provided. The time to destination parameter provides more interesting information, though. In-patient 19 moved from 47.1 to 40.8 seconds to achieve the task in the second try, whereas in-patient 31 reduced 10 seconds in a try. This tendency was detected in more in-patients, so perhaps further studies could establish that persons with full cognitive capabilities learn faster than challenged ones.

Regarding psycho metrics, differences between the in-patients are more relevant. It is important to note that Intervention Level is forcefully high because the proposed shared control approach does not work without the input of

Table 1. Metrics for in-patient 19 and 31

	IN-PATIENT 19				IN-PATIENT 31			
METRIC	TYPE	MEAN	STD	UNITS	TYPE	MEAN	STD	UNITS
Smoothness	Task	83.08	25.50	(per 100)	Task	85.07	22.50	(per 100)
Safety	Task	96.36	15.42	(per 100)	Task	96.54	15.28	(per 100)
Directiveness	Task	71.58	25.08	(per 100)	Task	69.92	22.18	(per 100)
Curvature	Task	0.11	0.36	(degrees)	Task	0.12	0.35	(degrees)
Time to Destination	Task	43.95	3.3	(secs)	Task	39.77	4.7	(secs)
Intervention Level	Psycho	85.13	—	(per 100)	Psycho	90.97	—	(per 100)
Disagreement	Psycho	17.29	19.80	(per 100)	Psycho	10.62	11.36	(per 100)
Inconsistency	Psycho	14.13	16.92	(per 100)	Psycho	9.37	9.89	(per 100)
Joystick Variation	Psycho	2.04	7.44	(per 100)	Psycho	2.00	7.27	(per 100)

human. Despite this, it can be observed that it is clearly larger in in-patient 31, meaning that he collaborates more with the machine. Disagreement is larger in this case for in-patient 19, even though the corrections of this person are lower than in in-patient 31. This could be attributed to the cognitive disabilities of in-patient 19, an idea that is reinforced by a poorer consistency as well. Finally, it is interesting to note that workload is low in both in-patients as desired.

4 Conclusions

Assistive navigation is a very active area of research, but standardization and proper evaluation metrics will be of key importance for testing and replicability. This paper has presented a review of most common metrics in the field, ranging from user condition medical evaluation to personal questionnaires. This review is expected to be useful for doctors and caregivers, engineers and end users all alike. Part of the proposed metrics have been tested in a real experiment to prove that they are significant and check their variation all together. Future work will focus on determining a minimum set of metrics for the different types of experiments in the field.

Acknowledgment

Authors would like to acknowledge support from the EC funded project *SHARE-it*: Supported Human Autonomy for Recovery and Enhancement of cognitive and motor abilities using information technologies (FP6-IST-045088). The views expressed in this paper are not necessarily those of the *SHARE-it* consortium.

References

1. Morris, A., Donamukkala, R., Kapuria, A., Steinfeld, A., Matthews, J.T., Dunbar-Jacob, J., Thrun, S.: A robotic walker that provides guidance. In: Proc. of the 2003 IEEE Int. Conf. on Robotics and Automation, Taipei, Taiwan, September 2003, pp. 25–30 (2003)
2. Parikh, S.P., Grassi, V., Kumar, V., Okamoto, J.: Usability study of a control framework for an intelligent wheelchair. In: Proc. of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005, pp. 4745–4750 (2005)
3. McLachlan, S., Arblaster, J., Liu, D.K., Valls, J., Chenoweth, L.: A multi-stage shared control method for an intelligent mobility assistant. In: Proc. of the 2005 IEEE 9th Int. Conf. on Rehabilitation Robotics, Chicago, USA, July 2005, pp. 426–429 (2005)
4. Simpson, R., Levine, S.P.: NavChair: An Assistive Wheelchair Navigation System with Automatic Adaptation, pp. 235–255. Springer, Heidelberg (1998)
5. Horiguchi, Y., Sawaragi, T.: Effects of probing to adapt machine autonomy in shared control systems. In: Proc. International Conference on Systems, Man and Cybernetics, Hawaii, USA, October 2005, vol. 1, pp. 317–323 (2005)
6. Crisman, J., Cleary, M.: Progress on the deictic controlled whellchair, pp. 137–149. Springer, Heidelberg (1998)

7. Frese, U., Larsson, P., Duckett, T.: A multigrid algorithm for simultaneous localization and mapping. IEEE Transactions on Robotics 21(2), 1–12 (2005)
8. Lawton, M., Brody, E.: Assessment of older people: self-maintaining and instrumental activities of daily living. Gerontologist (9), 179–185 (1969)
9. Crum, R., Anthony, J., Bassett, S., Folstein, M.: Population-based norms for the mini-mental state examination by age and educational level. Journal of the American Medical Association 269(18), 2386–2390 (1993)
10. Yesavage, J., Brink, T., Rose, T., Lum, O., Huang, V., Adey, M., Leirer, V.: Development and validation of a geriatric depression screening scale: A preliminary report. Journal of Psychiatric Research 17, 37–49 (1983)
11. Mahoney, F., Barthel, D.: Functional evaluation: the barthel index. Maryland State Medical Journal (14), 56–61 (1965)
12. Kilkens, O., Post, M., Dallmeijer, A., Seelen, H., van der Woude, L.: Wheelchair skills tests: a systematic review. Clinical Rehabilitation 17(4), 418–430 (2003)
13. Webster, J., Cottam, G., Gouvier, W., Blanton, P., Beissel, G., Wofford, J.: Wheelchair obstacle course performance in right cerebral vascular accident victims. J Clin Exp Neuropsychol. (11), 295–310 (1988)
14. Cooperstock, J., Pineau, J., Precup, D., Atrash, A., Jaulmes, R., Kaplow, R., Lin, N., Prahacs, C., Villemure, J., Yamani, H.: Smartwheeler: A robotic wheelchair test-bed for investigating new models of human-robot interaction. In: Proc. of the IEEE Conference on Intell. Robots and Systems (IROS), San Diego, USA (2007)
15. Parikh, S., Rao, R., Sang-Hack, J., Vijay, K., Ostrowski, J., Taylor, C.: Human robot interaction and usability studies for a smart wheelchair. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 2003 (IROS 2003), October 27-31, 2003, vol. 4(3), pp. 3206–3211 (2003)
16. Urdiales, C., Peula, J.M., Barrue, C., Cortes, U., Sandoval, F., Caltagirone, C., Annichiaricco, R.: An adaptive scheme for wheelchair navigation collaborative control. In: Asociation for the Advancement of Artificial Intelligence (AAAI 2008), Chicago, USA (2008)
17. Khoury, G.J., Kondraske, G.V.: Measurement and continuous monitoring of humanworkload associated with manual control devices. NASA JSC, Houston, TX, Tech. Rep. 91-011R (1991)
18. Hart, S., Staveland, L.: Development of NASA-TLX (Task Load Index): results of empirical and theoretical research, pp. 5–39. Elsevier, New York (1988)
19. Scherer, M., Jutai, J., Fuhrer, M., Demers, L., DeRuyter, F.: A framework for modeling the selection of assistive technology devices (atds). Disability and Rehabilitation: Assistive Technology 2(1), 1–8 (2007)
20. Gatti, N., Matteucci, M., Sbattella, L.: An adaptive and predictive environment to support augmentative and alternative communication. In: Miesenberger, K., Klaus, J., Zagler, W.L., Burger, D. (eds.) ICCHP 2004. LNCS, vol. 3118, pp. 983–990. Springer, Heidelberg (2004)
21. Mills, T., Holm, M., Trefler, E., Schmeler, M., Fitzgerald, S., Boninger, M.: Development and consumer validation of the functional evaluation in a wheelchair (few) instrument. Disability and Rehabilitation 24, 38–46 (2002)
22. Demers, L., Weiss-Lambrou, R., Ska, B.: The Quebec User Evaluation of Satisfaction with Assistive Technology (QUEST 2.0): An Overview and Recent Progress, pp. 101–105. IOS Press, Amsterdam (2002)
23. Urdiales, C., Poncela, A., Sanchez-Tato, I., Sandoval, F.: Efficiency based reactive shared control for collaborative human/robot navigation. In: Proc. of the IEEE Conference on Intell. Robots and Systems (IROS 2007), San Diego, USA (2007)

Conventional Joystick vs. *Wiimote* for Holonomic Wheelchair Control

L. Duran, M. Fernandez-Carmona, C. Urdiales, J.M Peula, and F. Sandoval

Dpto. Tecnología Electrónica, University of Málaga, Málaga, Spain
`lauradrosas@uma.es`

Abstract. In nowadays aging society, power wheelchairs provide assistance for non-pedestrian mobility, but inside narrow indoor spaces, holonomic ones are required. While they adapt well to complex environments, it is harder to control them via a conventional joystick. Thus, extra buttons and/or knobs are included to decide what to do. To make control more intuitive, we propose to use a *Wiimote* for holonomic wheelchair control. Experiments in a narrow environment have been succesful and prove that *Wiimote* requires less interaction to achieve the same results that a conventional joystick. This has been reported to reduce mental workload and, hence, allow more relaxed interaction with the wheelchair.

1 Introduction

Autonomy in an agent can be defined as the ability to make activities independently. However, a physically and/or cognitive challenged person may require some assistance to achieve autonomous navigation. Lack of human resources to assist elder people leads naturally to create systems to do it in an autonomous way (eg. [1]).

Specifically, the most typical mobility assistive devices are power wheelchairs, typically controlled via a conventional joystick. Unfortunately, houses nowadays are not large enough in most cases to allow easy wheelchair navigation as turning in narrow spaces and corners is quite challenging with a typically wide vehicle. In order to solve this problem, some research has focused on holonomic wheelchairs [2], that can move in any direction and turn in place smoothly (Fig.1.a). However, controlling these wheelchairs is far from intuitive; for example, it is necessary to distinguish between a turn or lateral displacement (Fig.1.b). Indeed, not every person may have the required skills to control these power wheelchairs, specially if they have a significant physical disability. Poor use of technology can result in systems that are difficult to learn or use and even may lead to catastrophic errors [3]. This may occur because, while there are typically reductions in physical workload, mental workload has increased [4]. Furthermore, some persons may be unable to control the chair on their own. It is consequently of extreme importance to create a new generation of tools to assist people with disabilities, so that their independence and autonomy is improved [5].

Some authors have recently employed a *Wiimote* for robot control [6]. The main advantage of the *Wiimote* is that it includes an accelerometer that allows

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1153–1160, 2009.
© Springer-Verlag Berlin Heidelberg 2009

motion-based commands along with traditional direction-based joystick control. This paper presents a system to control an holonomic wheelchair using a *Wiimote*. We are targeting people with physical disabilities that make it difficult to control complex joysticks. Persons with cognitive disabilities could use it as well with some assistance from an onboard computer [7]. Hardware and low level control is described in section 2. The system implementation and commands used to control the wheelchair are commented in section 3 and experiments appear in section 4. Finally, section 5 shows conclusions and future work.

2 Spherik and the *Wiimote*

In this work, we use Spherik as holonomic wheelchair. Spherik is manufactured by ESCLATEC[1]. Its main advantage is that it can adapt to the complexity of an environment potentially unprepared for wheelchair navigation. As most wheelchairs, it presents 2 Degrees of Freedom (DOF) to move ahead with variable direction, but also with 3 DOF, as it moves freely with the three degrees of freedom of the rigid body plane movement. This is achieved via 3 sets of omnidirectional wheels with freely rotating spherical rollers. 2 of these sets are driven by DC electrical motors, while the third one is passive. Battery Autonomy ranges from 4 to 5 hours in normal working conditions. Spherik includes a conventional joystick with two knobs and two buttoms. Combining all these, we choose different navigation modes: conventional, on-site turn or sideways motion (Fig. 1.b). All control variables in Spherik are published in *modbus*, a well known protocol for industrial control[2]. Changing these variables, we can send commands to the motors, establishing the platform speed and direction. Originally, this was done via the joystick, but in this work we access the Spherik's inner microcontroller to access *modbus* via TCP (standard *modbus* port 502 via socket). Naturally, this gives us access to motors from any PC, so controllers connected to that PC also control Spherik motors.

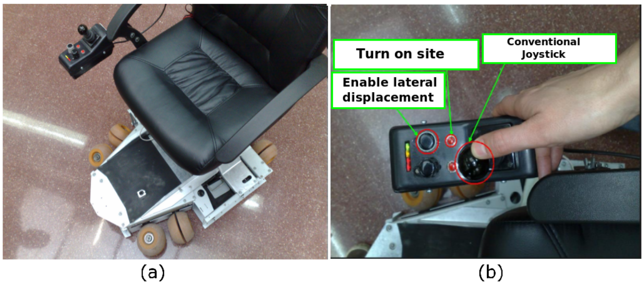


Fig. 1. (a) Holonomic wheelchair Spherik, by ESCLATEC. (b) Spherik joystick and controls to turn.

---

[1] http://www.esclatec.com/

[2] http://www.modbus.org/

The Wii Remote commonly known as *Wiimote*, is the primary controller for the Nintendo Wii console. Its motion sensing capability allows the user to interact with and manipulate items on screen via movement and pointing through the use of accelerometer and optical sensor technology.

Instead of the traditional controllers paradigm, the *Wiimote* assumes a one-handed remote control-based design. To make motion sensitivity more intuitive, it is designed as a remote control. That makes it fit perfectly for pointing and converts the *Wiimote* into an special device because of its peculiar way of interaction with the user.

The *Wiimote* consists of three hardware components: The ADXL330, a 3-axis accelerometer [8][9], which is contained within the *Wiimote*; the communications link, provided by the Bluetooth units in the *Wiimote* and the gateway processor; and the PC itself to perform the data processing and fall detection. The Wii Remote Bluetooth protocol can be implemented on other devices including cell phones, which often have poor usability with games.

Concerning software, the *Wiimote* is compatible with most of development suites as Borlan Delphi, C++, C, Java, and VB. Specifically, our system relies on *CWiid*, a collection of Linux tools written in C for interfacing to the Nintendo *Wiimote*, including an event-based API, an event/mouse/joystick driver, and GUI/control panel.

3 Proposed Architecture

Our system uses the *Wiimote* because it adapts perfectly to our requirements. First, we need enough degrees of freedom to control an holonomic wheelchair. A traditional joystick works with a plane and addicional control can be only provided by adding button/knobs, so there is no continuity in manipulation. Furthermore the *Wiimote* is a commercial and very affordable device so it is easy to get by any person. It was designed with the purpose of symplifying the interaction between a person and a machine. Similarly, the proposed control architecture makes use of its ADXL330 accelerometer. It captures displacements and rotations in the 3D reference system of the *Wiimote*, so interfacing with the wheelchair relies on hand manipulation of the remote control. Hand movements will be translated in displacements of the wheelchair to reach the desired final position.

Our complete system consists of a Wii remote control, a laptop or a PC, a Bluetooth dongle and an holonomic robotic wheelchair. With the three axes of the accelerometer, we can control three different movements of the wheelchair (Fig. 2). We have chosen three uncorrelated movements to map to the WiiMote: the pitch axis to move forward and to stop, the yaw axis to turn right/left and the roll axis for the steering movement (right/left). Naturally, combinations of these movements allow navigation in any direction. These variables are obtained via the CWiid library, a GPL software to program the *Wiimote* available on the Internet[3].

---

[3] http://abstrakraft.org/cwiid/

As soon as a person moves the *Wiimote* with the hand, the accelerometers readings are sent to the laptop through Bluetooth. Our software application translates the values of the accelerometers into longitudinal, transversal and rotational velocities for the microcontrollers of the wheelchair. Then, we communicate the laptop and the wheelchair through a modbus by sending these velocities.

It can be observed in Fig. 2 that we only work with the x and z axes, as motion in y in the WiiMote is not reflected in the CWiid accelerometer variables and we can not use the IRbar (extern landmark for *Wiimote* localization) while freely moving around. Nevertheless, these two directions fit correctly with the usual hand movements of a person, as a wrist turn or arm elevation. As we are looking for an easy and intuitive interface, we chose as basic motion commands the common hand actions: arm up/down and right/left. Sideways displacements have been related to roll movements/wrist turning, as these are not as common as the others. In order to capture left and right turns without the IRbar, we just calculate the magnitude of the roll motion: major roll values are related to sideways displacements, whereas low values are related to natural, joint induced driftings while trying to keep the control in the XY plane. We expect this simple interface to enable a natural and simple interaction.

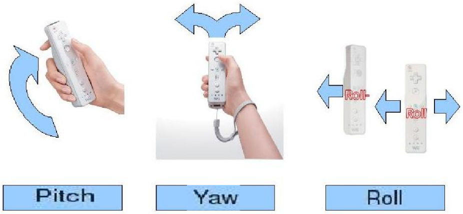


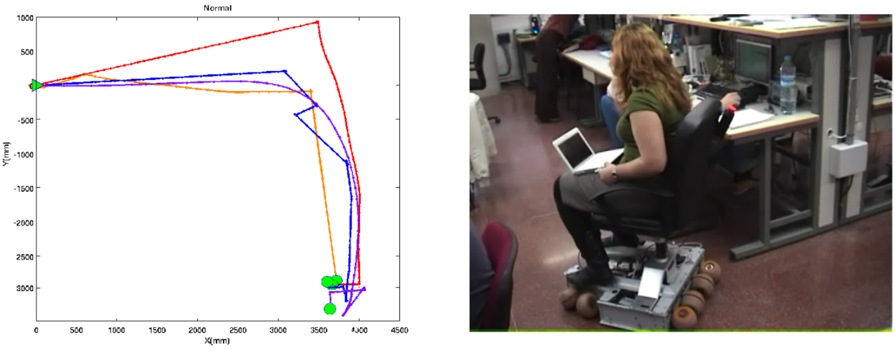
Fig. 2. *Wiimote* motion directions

4 Tests and Experiments

Before testing the system on users from the target population (persons with physical disabilities, but neglectable cognitive ones), it is necessary to undertake tests on a number of able-bodied subjects usually involving a test course. Metrics to be collected typically include time required to travel the course, the number of safety violations and the amount of effort required to drive the course [10]. In some wheelchair experiments with HCI (Human-Computer Interaction), effort was measured in terms of number of interactions with the control device. Yanko proposes for a single switch system to use: i) the number of clicks required to navigate the course; ii) the amount of time spent scanning to get to the necessary commands; iii) the amount of time spent moving or executing the given commands; and iv) the total amount of time spent on the course. As we have no clicks, but joystick commands instead, we will evaluate test courses via: i) the Intervention Level, defined as the portion of time that the user moves a joystick [11]; ii) Joystick

Variation, which measures a change of more than 10% in the position of the stick and has been used as an indirect measure of workload [12]; iii) time spent stopped; and iv) global trajectory time. We assume that it is desirable to keep a low Joystick Variation and to minimize time stopped as well as global trajectory time. It must be noted that button operation does not have an impact on Intervention Level because it is a punctual operation in our experiments, unlike in *Single Switch Controllers*. Furthermore, choosing a joystick mode requires consideration, which is reflected on stop time. Hence, it is reasonable that button operation decreases Intervention Level as it implies time spent in decisions rather than intuitive navigation. Low joystick variation is recommended in navigation because it means a firm hand movement, except in cases of tremors where they are filtered. High Intervention Level means more time moving the joystick and less trying to figure out how to operate it and a low global time spent indicates efficiency of these commands. Using these metrics, we compare the performance of users driving via the wheelchair joystick (Fig.1) and the *Wiimote*. The aim of these metrics is to show that for holonomic wheelchairs, the *Wiimote* is more flexible and intuitive. It provides more time to move or make decisions and speed to react to objects too. Our preliminary tests have been performed by 5 people. Each one followed the same trajectory twice (test and experiment). Some of these users had experience with the *Wiimote* but only as a game device. We did not notice any difference between them because no one had experimented with wheelchairs before, although users of wheelchairs are not very used to holonomic wheelchairs.

Fig. 3 shows our trajectory designed to benefit from the advantages of an holonomic wheelchair, as there are turns and maneouvres that could be complicated with a conventional one. Blue lines belongs to an experienced user and red lines to a novel one. Both of them with only a little experience with the *Wiimote* but as a game device. The main difference between them are in the narrower areas of the enviroment, where navigation is more difficult for the novel user and it provokes peaks on the trajectory.



(a) (b)

Fig. 3. (a) Paths by joystick and *Wiimote* by two different users: experienced one (orange-red) and non-expert (violet-blue). (b) Detail of the first turn in the path.

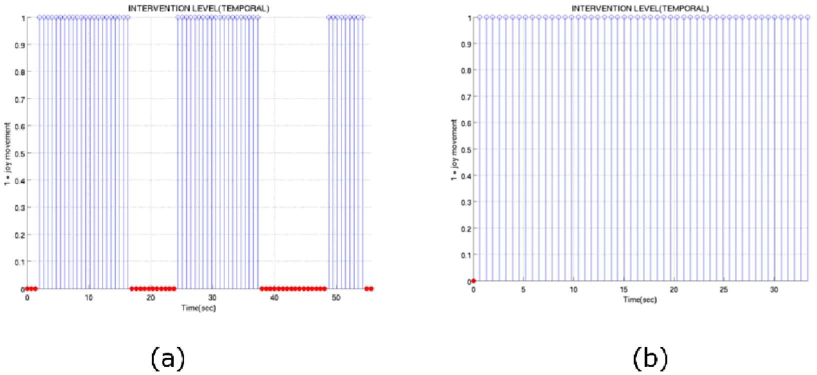
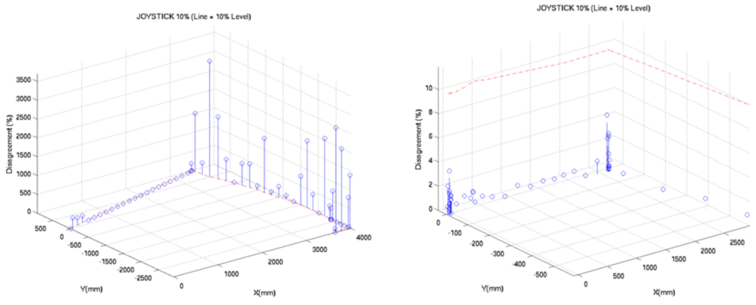


Fig. 4. (a) Intervention Level during joystick test made by the experienced user. (b) Intervention Level during *Wiimote* test made by the experienced user.

Fig. 4.a is related to the joystick. As we see, the Intervention Level is practically constant in the whole trajectory. We do not use the joystick while turning the wheelchair to change direction, as we have to manually choose between sideways motion or turning. Basically, using the Spherik joystick, persons simply move ahead and decide when to turn via buttons/knobs. We spent a little more than 50 seconds to reach the final position, being approximately more than 30 seconds invested in using the joystick. We almost never stop the wheelchair. 20 seconds were spent pushing buttons, as presented in Fig. 4.a. It needs to be noted that the Intervention Level is related to joystick usage, not necessarily to active commands. Hence, when the wheelchair is turning over itself in the same position, it is equal to zero.

Fig. 4.b is related to the equivalent *Wiimote* test. The trajectory is almost the same, but it can be observed that Intervention Level is clearly more homogeous, basically because the *Wiimote* does not need to be kept in a standby position. This means that the user can rest the *Wiimote* on his/her lap while moving without actually sending a command to the chair. This is also reflected in the Joystick Variation parameter in Fig.5.b, where peaks are related to actual commands. It can be observed, as expected, that commands are fed to the wheelchair only when necessary: to change directions. Still, as observed in Fig.4.b, there are no zero-Intervention Level areas, meaning that we do not need to change from one controller type to another during the trajectory. Fig.5.a shows the joystick variation for the Spherik joystick test. It can be observed that variation is not very large either: as comented, most turns involved buttons instead of joystick. However, it can be also observed that variations are more spreaded over the main curve, whereas in the *Wiimote* case, they are punctual. This means that the user keeps modifying the trajectory in the first case but not in the second. All in all, the *Wiimote* test took around 34 seconds and the whole time was invested in manipulating the remote, but only two peaks were significant, while the rest was related to involuntary manipulation due to body motion.



(a) (b)

Fig. 5. (a) Joystick variation. (b) *Wiimote* variation.

5 Conclusions

This work has presented a system to manoeuvre an holonomic wheelchair via a *Wiimote* controller. Our target is to provide a simple, intuitive method to benefit from all possible wheelchair movements without actually having to use buttons or knobs, so that persons with cognitive or physical disabilities find it easier to interact with it. In order to do so, the *Wiimote* is connected to a PC via Bluetooth and the PC connects to the wheelchair via socket-modbus to interface with the motors. We have defined a set of very simple gestures to choose movements: forwards, turn left, turn right and sideways motion left and right. Backwards motion is not allowed by medical advice. The proposed HCI has been compared to a conventional joystick one in an indoor environment in terms of number of joystick movements, as proposed by Yanko [10]. In both cases, an holonomic wheelchair is used, as the chain of commands for a non-holonomic wheelchair to achieve the same results would be far longer. However, evidence proves that *Wiimote* requires less movements and, hence, seems more intuitive than conventional controllers. Further work will focus on implementing a robotic control architecture on Spherik to assists users that can not complete trajectories on their own.

Acknowledgment

Authors would like to acknowledge support from the EC funded project *SHARE-it*: Supported Human Autonomy for Recovery and Enhancement of cognitive and motor abilities using information technologies (FP6-IST-045088). The views expressed in this paper are not necessarily those of the *SHARE-it* consortium. Authors will also like to thank ESCLATEC and Antonio B. Martinez-Velasco, creators of Spherik.

References

1. Prassler, E., Scholz, J., Fiorini, P.: Navigating a robotic wheelchair in a railway station during rush hour. Int. Journal on Robotics Research 18(7), 760–772 (1999)
2. Takayuki, O.: Study on a holonomic ominidirectional power wheelchair mechanism and basic operation. Nippon Robotto Gakkai Gakujutsu Koenkai Yokoshu 17, 1145–1146 (1999)
3. Norman, D.: Design rules based on analyses of human error. Communications of the ACM 26, 254–258 (1983)
4. Weiner, E.: Human factors of advanced technology (glass cockpit) transport aircraft. NASA TR 117528 (1989)
5. Cabrera, M., Burgelmen, J., Boden, M., da Costa, O., Rodriguez, C.: Ehealth in 2010: Realising a knowledge-based approach to healthcare in the eu. Technical Report EUR EN 21486 (2004)
6. Lapping-Carr, M., Jenkins, O.C., Grollman, D.H., Schwertfeger, J.N., Hinkle, T.R.: Wiimote interfaces for lifelong robot learning smart wheeler. In: Proc. of the AAAI Spring Symposium, California, USA (2008)
7. Urdiales, C., Peula, J.M., Barru, C., Corts, U., Sandoval, F., Caltagirone, C., Annichiaricco, R.: An adaptive scheme for wheelchair navigation collaborative control. In: Proc. of 2008 Fall Symposium of the Asociation for the Advancement of Artificial Intelligence (AAAI 2008), Arlington, USA (2008)
8. Castaneda, K.: Nintendo and pixart team-up. Nintendo World Report (2007)
9. Wisniowski, H.: Analog devices and nintendo collaboration drive videogame innovation with imems motion signal processing technology. Analog Devices (2009)
10. Yanco, H.A.: Evaluating the performance of assistive robotic systems. In: Proc. of Workshop on Performance Metrics for Intelligent Systems, Gaithersburg, MD, USA, pp. 417–419 (2002)
11. Cooperstock, J., Pineau, J., Precup, D., Atrash, A., Jaulmes, R., Kaplow, R., Lin, N., Prahacs, C., Villemure, J., Yamani, H.: Smartwheeler: A robotic wheelchair test-bed for investigating new models of human-robot interaction. In: Proc. of the IEEE Conference on Intell. Robots and Systems (IROS), San Diego, USA (2007)
12. Clarke, D., Yen, S., Kondraske, G.V., Khoury, G.J., Maxwelle, K.J.: Telerobotic network workstation for system performance and operator workload monitoring. NASA JSC, Houston, TX, Tech. Rep. 91-013R (1991)

Normal versus Pathological Cognitive Aging: Variability as a Constraint of Patients Profiling for AmI Design

Luiza Spiru[1], Camelia Ghita[1], Ileana Turcu[1], Lucian Stefan[1], Ioana Ioancio[1],
Costin Nuta[1], Mona Blaciotti[1], Mariana Martin[1], Ulises Cortes[2],
and Roberta Annicchiarico[3]

[1] Ana Aslan International Academy of Anti-Aging,
Piata Kogalniceanu Nr.1Bucharest, Romania
[2] Universidad Politecnica de Catalunia, Jordi Giona 1, 08034 Barcelona, Spain
[3] Fondazione Santa Lucia, Via Ardeatina 354, Rome, Italy
`lsaslan@brainaging.ro`

Abstract. The development of supportive environments and Assistive Technology (AT) is a priority recommended by the International Plan of Action on Ageing (Madrid 2002). The first essential matter related to the creation and implementation of a certain Ambient Intelligence (AmI) system is to know for whom it have to be designed, in order to meet user's profile and needs with the functional capabilities of the intelligent and semi-autonomous assistive device, and to identify the best ways of human-machine interaction. The multiple dimensions of cognitive variability in the elderly make these tasks very difficult. A comparative overview of the cognitive changes in normal and pathological aging could be useful in order to deal with the cognitive variability constraint in the establishment of user profiles and target populations.

Keywords: cognitive variability, assistive technology, user profile.

1 Background

One of the most important answers to the economical and social challenges issued by global population aging is keeping old people as much as possible independent, safe at their home and at the same time socially inserted. An answer equally reflected in their life quality improvement, the delay of their institutionalization and the lowering of health care costs and burden for the society. In this respect, the International Plan of Action on Ageing (Madrid 2002) recommends the development of supportive environments and Assistive Technology (AT) as a priority direction. Besides assisting the disabled or the elderly in performing activities of daily living, assistive technology devices also allow a better control of costs, as most equipment rentals are monthly far less expensive than recruiting the services of a nurse or home health aide. There are also limitations, as some assistive devices are not provided as rental equipment or may require structural modifications to a home. They may or may not add value to a home depending primary on their nature and functionality [1]. By combining the ongoing technological developments with user-centered designs, an extended multidisciplinary effort would be able to enhance the acceptance of the intelligent assistive systems and their capacity of improving user's life quality in a non-intrusive way.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1161–1167, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Ambient Intelligence (AmI) is "a vision for future communication and human-machine interaction systems" [2], whose development is a complex and difficult process dealing with two main challenges [3]. The first is to define and implement a mechanism for the installation of custom-made AmI applications on portable devices according to their specific hardware constraints. The second deals with change and upgrades: hardware and software technologies for AmI are in permanent evolution. The environment where AmI applications have to act is dynamic, so that they must be able to react in a more or less automatic and unassisted way to the environmental changes, i.e. they must include mechanisms that allow their dynamic adaptation to newly issued conditions. Finally, any possible change of user demands could require the more or less automatic reaction of AmI devices [4].

The first essential matter related to the creation and implementation of a certain AmI system is to know for whom it has to be designed [5], as the elderly are an exceptionally heterogeneous population segment regarding their physical and cognitive functions. This was a matter in our SHARE-it project too, another actual endeavor dedicated to the development of the next generation of intelligent and semi-autonomous assistive devices for elderly and people with cognitive and/or motor disabilities [6]. The establishment of SHARE-it target population, user's profile, the impairments and needs to be supported, was the main step in the assessment of the SHARE-it AmI platforms' suitability and design. The task was complicated by the fact that, in contrast to the large majority of existing AmI platforms designed to support elderly in their activities of daily living that are predominantly centered on physical disabilities, those envisaged by the SHARE-it project aime to equally attempt the cognitive ones, being well known that in the elderly the two kinds of disability are usually mixed [7]. An overview regarding the cognitive particularities in normal and pathological aging proved necessary and useful for the establishment of SHARE-it user profiles and target populations. The main aspect it revealed was that the critical marker of normal decline due to aging - the huge intra-individual and inter-individual variability – is a capital constraint in user profiling /target group definition that requires particular solutions to solve it. Additional variability induced by the incidence and development of age-specific cognitive pathology constitutes an additional burden.

2 Variability of Cognitive Changes in Normal Aging

As people age, biological and psychological changes turn up in multiple ways far from being uniform across the whole brain, across all cognitive domains, or across all the older individuals [8]. Moreover, the aspect related to their span that divides them into sort-term and long-term changes is also to be taken into account. Cross sectional and longitudinal studies document that the two major types of cognitive abilities, the 'crystallized' one (related to long term memory) and the 'fluid' one (related to short term memory storage while processing information), evolve differently across the lifespan [9]. Crystallized abilities' increment through education, occupational and cultural experience is less affected by aging and disease, often remaining intact in the early stages of dementia or even after brain injury. Their rate of increase slows in late adulthood and goes slowly from the ninth decade onward [9]. By contrast, fluid abilities decline gradually after peaking in the mid 20s until the 60s, when their decline accelerates, and they are susceptible to neurological insults and effects of biological aging. Normal cognitive aging is also marked by a general behavioral slowing, a more difficult selective attendance to information and inhibition of irrelevant environmental input [10], mainly related to frontal lobe atrophy [11].

How much cognitive decline is normal in aging? – Considering the average, an excellent, synthetic answer is provided by K.J. Anstey and L-F. Low (2004) [12]: the mean decline is of 1 to 2 standard deviations (15–30 IQ points) in fluid cognitive abilities between 20–70.7 years of age, followed by an acceleration of about 0.5 of a standard deviation per decade. On the commonly used cognitive tests, on a 33 item vocabulary test no difference in the average score was detected for 25 versus 70 years of age. On test of recall of digits in the order of their presentation, a 25 year old will recall about seven, whereas a 65 year old will recall about 5.9. On a test of coding speed limited to 90 seconds, a 25 year old will correctly code about 78 items, whereas a 70 year old will code about 51 items correctly.

3 Intraindividual Variability

It is a common component of aging-related cognitive decline and the behavioral changes associated with neurodegenerative and other brain-related disorders. Despite the large number of reports on intra-individual variability there is little synthesis, and no direct examination of the neural underpinnings [13, 14]. Intra-individual variability of physical status and affect/beliefs, as well as their relations with cognition, examined in older adults (healthy elderly, individuals with arthritis as a non-neurological health-related disturbance and people with dementia as neurological compromise), showed that increased inconsistency in non-cognitive domains was associated with poorer cognitive function [15]. The investigation of within-person variability across cognitive domains (termed dispersion) in 304 non-demented older adults aged 75-92 realized in a Canadian study that pursued the relationship between aging and the quantitative (overall levels) and qualitative (patterns) dispersion of cognitive functions revealed higher levels of dispersion in the old-old adults and those identified as having some cognitive decline, suggesting that dispersion level may serve as a marker of cognitive integrity Qualitatively, three distinct dispersion profiles were identified using clustering methods, closely related to demographic, health and performance characteristics of the individuals, suggesting that patterns of dispersion may be meaningful indicators of individual differences [16].

Interest in the study of intra-individual variability is growing rapidly. A promising direction is toward developing a more systematic and intensive conception of the meaning and significance of the wide variety of manifestations of intra-individual variability. Interesting data issued in EU Share-it Project (FP6-IST-045088) regarding the assessment of intra-individual short-term changes in cognitive functions (including individual emotional state) and their impact on the activities of daily living and human machine interaction (HMI) was recently published [17].

4 Interindividual Variability

Enormous variability exists across the individuals. The retention of unaltered cognitive function could be present in 70- 80 years old people and they could perform as well or better than younger ones, while others show declining signs at 60s. Some older adults have excellent episodic memory function but impaired executive functions, and vice versa. Certain aspects of attention and memory hold up well with age, while others significantly decline, as well as perception mainly due to the decline of sensory capacities [8]. Deficits at these early processing stages could affect cognitive functions later in the processing stream (higher-level cognitive functions such as language processing or decision making). Complex cognitive tasks may also depend on a set of executive functions that manage various components of the tasks. Impairment of executive functions as a key component of age-related decline in a range of cognitive tasks that overlap and interact in complex ways is supported by evidence [8]. Selective attention of older adults appear to be slower than younger, mainly due to a general slowing of information processing. The findings differ across studies and across tasks, older people appearing to be slower in responding to the targets, but similar with younger regarding the impairment of attention by distraction [18]. Older adults exhibit significant deficits in tasks that involve active manipulation, reorganization and integration of information, such as decision making, problem-solving, planning goal directed behaviors (working memory).

The basic cognitive functions most affected by age are attention and memory. The task to discern elderly people with normal, age-related memory complaints from those really having memory disorders or at risk to develop them is a difficult one. As self referrals for memory complaints are not a good indicator of cognitive impairment because in older adults they are often associated with depression, their paralleling with specific questionnaires fulfilled by relatives or carers is recommended. Anstey and Low (2004) [12] provided a pertinent characterization of memory complaints depending on normal aging or disease. In normal aging they are kept at a low level, do not interfere with daily functioning, do not alter cognition tests, are not paralleled by changes of mood and their onset is often unclear. By contrast, depression more likely includes memory complaints, whose onset is sudden; many of them interfere with daily functioning and are paralleled by depressed or anxious mood, as well mild cognitive impairments that remit on recovery. In dementia, memory complaints have gradual onset, are variable and very general, interfere with daily functioning, memory loss awareness could be absent, cognition impairments worsen in time, and labile or blunted mood could be associated.

The episodic memory problems experienced by older adults may involve deficient encoding, storage, or retrieval processes [9]. What may be particularly critical for episodic memory impairment in the elderly is the extent to which an event is bound to its special and temporal context. Considerable evidence points to retrieval as a source of episodic memory problems in aging. Older adults do not have significant impairments in semantic memory (storage of general knowledge about the world) during normal aging. Although access to information may be somewhat slower (particularly for words and names), the organization of the knowledge system seems unchanged with age [8]. The evidence suggests that autobiographical memory is also largely preserved with age. Oder adults remember as much as the young ones about the details and the circumstances surrounding highly emotional public events. Regarding procedural memory, the highly skilled activities are acquired more slowly than episodic memories to extensive practice. Once acquired, procedural memories are expressed rather automatically and are not amendable to description.

Inter-individual variability is likely attributable to a range of factors and mechanisms - biological, psychological, health-related, environmental and life style. One possibility is that variability is related to differential internal compensatory mechanisms. Recent functional neuroimaging studies have found different patterns of brain activation in older and younger adults while performing identical memory or working memory tasks. One such pattern involves greater bilateral activation in older adults for tasks that activate only unilateral brain regions in young adults [9]. This activation observed particularly in a sub-group of high-functioning older people and has been interpreted as compensatory activity, expression of some reorganization of the aging brain. The suggestion that bilateral activation represents inefficient of less selective cognitive processing in older adults was also advanced. Lifestyle variables have also been attempted as factors accounting for differential cognitive aging. Active lifestyles are generally associated with better outcomes (especially aerobic exercise producing substantial benefits particularly for tasks requiring executive control [19]. Performance on these same kinds of non-automatic tasks is also particularly sensitive to circadian rhythms [8]. Older people perform better at their peak time of the day (usually in the morning), on tasks requiring inhibitory control.

The impact of premorbid cognition is also largely studied. The decline in cognitive abilities is partially determined by the individual's previous level of cognitive function, as assessed by measures of crystallized abilities (verbal reasoning and word knowledge). Information about previous occupational level, interests, educational level gathered from the individual and an informant could additionally help. Any assessment of cognitive impairment by must include the psychometric and interview assessment of premorbid cognitive function [12].

5 Differentiating Normal Changes from Cognitive Pathology

There are no rules to easily distinguish between normal aging and dementia as the most invalidating cognitive pathology. Differentiating normal cognitive aging from MCI and early stage dementias requires a complex neuro- and psycho-geriatric assessment. Screening instruments are available but they only yield a score that is associated with an established risk in population based studies [12]. MMSE scores under 24 are generally considered as indicating cognitive impairment, but highly educated patients are less likely to be diagnosed. Key considerations for diagnosis of a cognitive disorder include a history of cognitive abilities decline, evidence that the decline interferes with everyday activities, and impairment evident in more than one cognitive domain. Interviewing an informant is the best way of obtaining an accurate history. Patients who present a diagnostic challenge include those with low education, premorbid intellectual disability, those currently suffering another mental health condition such as depression or psychosis.

6 Cognitive Variability Constraint in User Profiling

The elderly represent the population segment for which the defining patterns of disability and functional impairment are the most difficult [20]. The above described, multiple dimensions of cognitive variability in the segment of old and especially cognitively disabled old people makes very difficult the task of establishing the most circumscribed profile of the user for whom a given assistive technology application has to be designed. It's about the establishment of suitable batteries of assessment tests able to select a user whose profile, a particular "cluster" of well established disabilities and needs could maximally benefit from the appropriately chosen intelligent devices embedded in the assistive system. It's about to meet suitable user's needs with the functional capabilities, and equally limitations, of the intelligent and semi-autonomous assistive devices, and to identify the best ways of interaction between the user and the tool. A fruitful solution, adopted in the SHARE-it project in order to minimize the cognitive variability constraint was to individuate a group of individuals as large as possible that can achieve benefits as effectively as possible, in terms of functional, social, and clinical features.

In our opinion, the main cognitive characteristics and impairments of an "average" user, thought to benefit from an intelligent assistive device designed for mild to moderate impaired old people should respect the scores specified in the above mentioned scales: Grober-Buscke evaluation score or equal with or higher than 10-12, Rey Figure Test score of 20-25 (from 36), Yessavage Geriatric Scale lesser or equal 10 (from 15), MMSE score over or equal with 18 (from 30), Clock Drawing Test score must over or equal to 8 (from 10), Tinetti static score lesser or equal to 22, Tinetti Dynamic score must lesser or equal to 20, IADL score must be equal to or over than 4 (from 8) and ADL score must be equal to or over than 3 (from 6).

References

1. Cadena, C.: Assistive Technology and the Impact on Senior Adults, Associated Content (2006), `http://www.associatedcontent.com/article/81295/assistive_technology_and_the_impact.html?cat=12`
2. Riva, G., Loreti, P., Lunghi, M., Vatalaro, F., Davide, F.: Presence 2010: The Emergence of Ambient Intelligence. In: Riva, G., Davide, F., Jssel-steijn, W.A.I. (eds.) Being There: Concepts, effects and measurement of user presence in synthetic environments, pp. 59–82. Ios Press, Amsterdam (2003)
3. Fuentes, L., Jimenez, D.: An Aspect-Oriented Ambient Intelligence Middleware Platform. In: Proceedings of 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC 2005), Grenoble, France (2005), `http://portal.acm.org/citation.cfm?id=1101482`
4. Fuentes, L., Jiménez, D.: Combining Components, Aspects, Domain Specific Languages and Product Lines for Ambient Intelligent Application Development. In: Pfeifer, T., et al. (eds.) Advances in Pervasive Computing 2006, Adjunct Proceedings of Pervasive 2006, Dublin (2006), http://www.pervasive2006.org/ap/pervasive2006_adjunct_4I.pdf
5. Casas, R., Blasco Marín, R., Robinet, A., Delgado, A.R., Yarza, A.R., McGinn, J., Picking, R., Grout, V.: User modelling in ambient intelligence for elderly and disabled people. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2008. LNCS, vol. 5105, pp. 114–122. Springer, Heidelberg (2008)
6. SHARE-it (Supported Human Autonomy for Recovery and Enhancement of cognitive and motor abilities using information technologies), FP6-2006-IST-6 project Nr. 045088, `http://www.ist-shareit.eu/shareit`

7. Horn, J.L.: A context for understanding information processing studies of human abilities. In: Vernon, P.A. (ed.) Speed of information processing and intelligence, pp. 201–238. Ablex, Norwood, NJ (1987)
8. Glisky, E.L.: Changes in cognitive function in human aging. In: Riddle, D.R. (ed.) Brain aging. Models, methods, and mechanisms. Frontiers in Neurosciences series, CRC Press, Boca Raton (2006), `http://www.neurosciencenetbase.com/ejournals/books`
9. Christensen, H.: What cognitive changes can be expected with normal aging? J. Psychiatry 35, 768–775 (2001)
10. Salthouse, T.A.: Interrelations of aging, knowledge, and cognitive performance. In: Staudinger, U.M., Lindenberger, U. (eds.) Understanding human development: dialogues with lifespan psychology, Kluwer Academic Publishers, Dordrecht (2003)
11. Greenwood, P.M.: The frontal aging hypothesis evaluated. J. Int. Neuropsychol. Soc. 6, 705–726 (2000)
12. Anstey, K.J., Low, L.-F.: Normal cognitive changes in aging. Australian Family Physician 33(10), 783–787 (2004)
13. MacDonald, S.W.S., Nyberg, L., Bäckman, L.: Intra-individual variability in behavior: links to brain structure, neurotransmission and neuronal activity. Trends in Neurosciences 29(8), 474–480 (2006)
14. Li, S.-C., Aggen, S.H., Nesselroade, J.R., Baltes, P.B.: Short-Term Fluctuations in Elderly People's Sensorimotor Functioning Predict Text and Spatial Memory Performance: The MacArthur Successful Aging Studies. Gerontology 47, 100–116 (2001)
15. Strauss, E., MacDonald, S.W.S., Hunter, M., Moll, A., Hultsch, D.F.: Intraindividual variability in cognitive performance in three groups of older adults: Cross-domain links to physical status and self-perceived affect and beliefs. Journal of the International Neuropsychological Society 8(7), 893–906 (2002)
16. Hilborn, J.V., Strauss, E., Hultsch, D.F., Hunter, M.A.: Intraindividual variability across cognitive domains: Investigation of dispersion levels and performance profiles in older adults. Journal of Clinical and Experimental Neuropsychology (2008), `http://www.informaworld.com/smpp/content~content=a901492159~db=all`
17. Budde, S.H., Barkowsky, T.: A Framework for Individual Cognitive Performance Assessment in Real-time for Elderly Users. In: AAAI 2008, Chicago, USA (2008), `http://www.ist-share-it.eu/shareit/Members/admin/posterAAAI2008_BuddeBarkowsky.pdf`
18. McDonald, J.M., Craik, F.I.M.: Effects of aging and task difficulty on divided attention performance. J. Exp. Psychol. Hum. Perc. Perf. 14, 267 (1988)
19. Colcombe, S., Kramer, A.F.: Fitness effects on cognitive function of older adults: a meta-analytic study. Psychol. Sci. 14, 125 (2003)
20. Myers, G.C., Lamb, V.L., Agree, E.M.: Patterns of Disability Change Associated with the Epidemiologic Transition. In: Robine, J.M., Jagger, C., Mathers, C.D., Crimmins, E.M., Suzman, R.M. (eds.) Determining Health Expectancies, John Wiley & Sons, Ltd., Chichester (2003)

Estimating the Embedding Dimension Distribution of Time Series with SOMOS

Pedro J. Zufiria[1] and Pascual Campoy[2]

[1] Grupo de Sistemas Dinámicos, Aprendizaje y Control. Departamento de
Matemática Aplicada a las TT. de la I. ETSI Telecomunicación.
Universidad Politécnica de Madrid, Spain
`pzz@mat.upm.es`
[2] Computer Vision Group. DISAM. ETSI Industriales.
Universidad Politécnica de Madrid, Spain
`campoy@etsii.upm.es`

Abstract. The paper proposes a new method to estimate the distribution of the embedding dimension associated with a time series, using the *Self Organizing Map decision taken in Output Space* (SOMOS) dimensionality reduction neural network. It is shown that SOMOS, besides estimating the embedding dimension, it also provides an approximation of the overall distribution of such dimension for the set where the time series evolves. Such estimation can be employed to select a proper window size in different predictor schemes; also, it can provide a measure of the future predictability at a given instant of time. The results are illustrated via the analysis of time series generated from both chaotic Hénon map and Lorenz system.

Keywords: embedding dimension, time series, window size, SOMOS architecture.

1 Introduction

Time series prediction is a ubiquitous problem in many different disciplines of science, engineering, economy, etc. Given a time series sample $\{x_0, x_1, \dots, x_{t-1}, x_t\}$, the prediction of future unknown values $\{x_{t+1}, \dots\}$ is usually based on the assumption that such following values depend in some way on a subset of the available previous ones. Hence, standard prediction models define a window of dependence (of size r) which slides over time to compute consecutive values of the time series:

$$x_{t+1} = G(x_t, x_{t-1}, \dots, x_{t-r+1}, \eta_t) \qquad (1)$$

where η_t may gather noise or disturbances associated with the process. In this context, recurrent neural architectures trained via supervised methods such as MLP and RBFs have been widely employed to approximate G, for a given window size. The estimation of the required window size r has received much attention in the past: a value smaller than required does not allow the predictor to work properly; on the other hand, a larger value than required (over-dimensioning the

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1168–1175, 2009.
© Springer-Verlag Berlin Heidelberg 2009

architecture) leads also to serious problems of parameter adjustment, efficiency and generalization capabilities.

The use of equation (1) is implicitly based on the assumption that there exists some underlying dynamical system

$$\frac{dy(\tau)}{d\tau} = F(y(\tau)), \qquad F : \mathbb{R}^n \to \mathbb{R}^n \tag{2}$$

of (a priori) unknown dimension n, so that the time series values may correspond, for instance, to the sampled measurements at time instants $\{\tau_0, \tau_1, \ldots, \tau_t, \ldots\}$ of some ith component of the system trajectory ($x_t = y^i(\tau_t)$). Hence, the predictor design could be grounded on the determination of (2); fortunately this computation needs not be made explicitly.

1.1 The Embedding Dimension

A fundamental result relating the time series and system (2) states that, under some regularity conditions on F, and assuming that the dynamics of (2) evolve within a manifold M of dimension $m \leq n$, there exist a value $d_e \in [m, 2m+1]$ (called *embedding dimension*) such that the space of values

$$Y(\tau) = (y^i(\tau), y^i(\tau - \tau_e), \ldots, y^i(\tau - (d_e - 1)\tau_e))$$

describes the dynamics of (2). In fact, the exists a diffeomorphism which embeds M into R^{d_e}, so that topological properties of the trajectories in (2) are reflected in R^{d_e}. The value τ_e is called the *delay or embedding time.*

The embedding dimension fundamental results were first formalized in [8]. There, it is proved that selecting the value $d_e = 2m+1$ guarantees the embedding for fairly general regular cases; nevertheless, such value is higher than could be expected a priori. In fact, it may be possible to obtain reasonable results with a smaller value for $d_e < 2m + 1$, as numerically proved in [6].

It is important to note that results in [8] were developed in the context of continuous-time dynamical systems. In order to translate such results to the discrete-time arena, the relationship between delay and sampling times must be taken into account. The delay time τ_e reflects the time gap between the different trajectory values selected to construct samples quantities in R^{d_e}. Such value may be larger than the sampling time τ_s (so that is defined as multiple of it, $\tau_e = \kappa\tau_s, \kappa \in \mathbb{N}$), and in general it should be estimated together with d_e [7].

As a consequence, equation (1) could be used to characterize the time series, provided the time dependencies gather previous values $x_t, x_{t-\kappa}, \ldots, x_{t-(d_e-1)\kappa}$. In case that time series does not have a too high correlation between successive values (i.e., “samples”), one can assume delay and sampling time values to be the same ($\kappa = 1$). This implies that the appropriate window size r in (1) can be bounded by $m \leq r \leq 2m + 1$.

On the other hand, the existence of (1) with finite r implies that there is a discrete-time underlying system

$$z_{t+1} = H(z_t), \qquad H : \mathbb{R}^r \to \mathbb{R}^r \tag{3}$$

so that the time series values correspond to some ith component of the discrete system trajectory ($x_t = z_t^i$). In fact, this straightforward result can be interpreted as a (simpler) discrete version of the embedding theorem so that $X_t = (x_t, x_{t-1}, \ldots, x_{t-d_e+1})$ with $d_e = r$ characterizes the dynamics of z_t. (Note that the dynamics in (3) will evolve within a manifold M' of dimension $m' \leq r$.)

1.2 Procedures to Estimate Embedding Dimension

There are basically three families of methods employed to estimate the embedding dimension associated with a given time series:

- Methods that exploit the smoothness property of the map, such as False Near Neighbors (FNN), the wavering product or Aleksic method [3,5].
- Methods that compute second order statistic measures such as Principal Components (PCA), Singular Values (SVD) or Singular-Spectrum (SSA) associated with the data [1,3].
- Methods which evaluate some invariant measures such as the correlation dimension [1,4].

In this work, a new method for estimating the embedding dimension is proposed. It is based on the use of a neural network architecture initially designed to address the problem of *dimensionality reduction* [2]. Here, we employ such architecture to process some data derived from the time series in order to approximate the overall distribution of its embedding dimension.

2 The SOMOS Neural Network

The *Self Organizing Map decision taken in Output Space* (SOMOS) neural network architecture has been proposed in [2] to address the problem of dimensionality reduction. Different capabilities of the network are outlined in such context, and some applications on standard problems as well as some novel applications on image processing are also illustrated in [2].

2.1 The Standard SOM

The standard Kohonen Self Organizing Topological Map (SOM) defines a finite set of neurons physically distributed on a set S which usually can be interpreted as embedded in $\mathbb{R}^m$ with coordinates $y \in \mathbb{N}^m$ (usually $m = 2$). The fixed relative location among the neurons defines the concept of neighborhood which leads to a topology in S. In addition, each neuron y has an associated weight vector $w_y \in \mathbb{R}^n$, which determines its response to the input data. Given an input vector $x \in \mathbb{R}^n$, the neuron y^* whose w_y^* is the closest to x (among all neurons $y \in S$) is selected and activated so that the value y^* can be interpreted as the output associated with x. Then, the weight vector w_y^* of such winning neuron is updated according to the input data; in addition, the weight vectors of the neurons located in a given neighborhood of y^* are also updated.

It is important to note that in the standard SOM each neuron is clearly identified either by its fixed physical location y or by its corresponding weight vector w_y.

2.2 SOMOS Architecture

The SOMOS net defines a modified SOM where each neuron is identified by a pair (y, w_y) but only w_y is defined in an initial phase. The training algorithm will determine the associated value y which will represent the output value associated with the neuron. During such training procedure, both (y, w_y) are updated; it is important to note that although y does represent a physical location (as in the standard SOM) it needs not be fixed for the given neuron all over the training procedure. Hence, the number of resulting values of y for all neurons is not predefined, as opposed to standard SOM. This will allow for the theoretical definition of infinite possible values: in fact, the coordinates y of each neuron belong to $\mathbb{Z}^n$ and are not a priori restricted to a set S. Finally, the map will define a function $f : \mathbb{R}^n \to \mathbb{Z}^n$.

As in standard SOM, an explicit discrete metric or distance is defined in the output space of neurons, which generates also a topology in such space. But in this case, the concept of neighborhood is not employed as in standard SOM training procedure; the metric will directly be employed for defining the output value y. The computation of y is grounded on the idea that distances between output values must preserve some proportionality with respect to distances of the corresponding data in the input space. That is why the Map is called SOMOS (Self-Organizing Map decision taken in Output Space).

2.3 SOMOS Training Scheme

The training algorithm can be summarized as follows. Once a new datum x is presented to the network, the winning neuron (y^*, w_y^*) is selected as the one whose weight w_y^* is the closest to x. Hence, the weight value of such neuron (an only such neuron) is updated following the standard SOM scheme

$$w_y^*(t+1) = w_y^*(t) + \alpha(t)[x(t) - w_y^*(t)],$$

where $\alpha(t)$ decreases upon the number of performed updatings of such weight. The output value y^* is computed via

$$y^* = \arg\min_{y \in \mathbb{Z}^n} \left\{ \sum_{i \in \Omega} (\|y - y_i\| - d_i/R)^2 \right\}, \tag{4}$$

where Ω represents a selected set of previously created neurons with output y_i and with input distance d_i from w_y^* (divided by a desired ratio R, see [2]). For instance, Ω may contain the previously computed neurons up to a fixed maximum number, and the y value for the first neuron can be arbitrarily assigned.

A priori, this discrete minimization problem can be computationally cumbersome. Nevertheless, the fact that $\mathbb{Z}^n$ can be embedded into $\mathbb{R}^n$ allows for the use of several standard minimization techniques to obtain suboptimal solutions.

2.4 SOMOS Results

Let us consider a set of input data $x \in \mathbb{R}^n$ which lie on an unknown space of unknown reduced dimension r; the SOMOS algorithm is first initialized so that output values lie in $\mathbb{Z}^n$. After the training process, the resulting neuron outputs y, in the average, will only have r nonzero components. As a consequence, r can be easily estimated and the resulting function $f : \mathbb{R}^n \to \mathbb{Z}^n$ can be employed to estimate a dimensionality reduction mapping from $\mathbb{R}^n$ to $\mathbb{R}^r$.

When data lie on a set of varying local density or dimensionality, the SOMOS neuron outputs are distributed according to such local density and dimensionality. As shown later, such information can be employed to assess the quality of the prediction steps.

3 Estimating Embedding Dimensions with SOMOS

Given the time series $\{x_0, x_1, \ldots, x_{t-1}, x_t\}$, let us assume that its evolution is characterized by an equation of the form (1) where noise η_t is not too large when compared with distances between pairs of neighbor points.

In that case, if we construct a set of vectors of the form

$$(x_t, x_{t-1}, \ldots, x_{t-d+1}) \in \mathbb{R}^d, \tag{5}$$

they are expected to lie all over $\mathbb{R}^d$, provided $1 < d \leq r$. On the other hand, if we select $d = r + 1$, the vectors of the form $(x_t, x_{t-1}, \ldots, x_{t-r}) \in \mathbb{R}^{r+1}$ should lie “around” the r-dimensional manifold implicitly defined by

$$x_t - G(x_{t-1}, x_{t-2}, \ldots, x_{t-r}, 0) = 0,$$

provided G is smooth enough. The same would happen for a set of vectors of the form $(x_t, x_{t-1}, \ldots, x_{t-r}, x_{t-r-1}) \in \mathbb{R}^{r+2}$ which also should remain around the r-dimensional manifold implicitly defined by

$$x_t - G(G(x_{t-2}, \ldots, x_{t-r-1}, 0), x_{t-2}, \ldots, x_{t-r}, 0) = 0,$$

and so on.

Therefore, the estimation of r can be carried out via the above proposed SOMOS architecture, where different values of d can be employed to construct data vectors of the form (5). The value of r will be estimated as the larger value of d, such that training SOMOS with (5) will result in output values y where in general all d components are different from zero.

The procedure is based on estimating the dimension of a set the samples of the time series are supposed to lie within. There are many different definitions concerning the dimension of a set. Besides the topological dimension D_T (also called covering dimension, due to Brouwer, Urysohn and Menger) and the Haussdorf dimension D_H (also called Haussdorf-Besicovitch dimension), there is a whole family of Box dimensions (also called Box counting dimensions) associated with a measure: they include the Lower box dimension D_{Lb} (also called Lower

Minkovski) and the Upper box dimension D_{Ub} (also called Upper Minkovski or Kolmogorov capacity). If both Lower and Upper box dimensions coincide, they define the Minkovski-Bouligand dimension D_0. On the other hand, information dimension D_1, also called entropy dimension and logarithmic density, is based on the probability measure on the set; and correlation dimension D_2 is based on distances and probabilities associated with pairs of points.

Box dimension, information dimension and correlation dimension are concrete cases of the infinite family of Rényi generalized dimensions (derived from Rényi's q-entropies, $q = 0, 1, \ldots$) with $q = 0$, 1 and 2 respectively, and satisfy $D_0 \geq D_1 \geq D_2$. Among them, the correlation dimension D_2 is the most frequently employed index, since it can be efficiently estimated from recurrence times [4]. Given a set of points N, defining $g(\epsilon)$ as the number of pairs of points whose distance is below ϵ, it is satisfied that

$$C(\epsilon) = \lim_{N \to \infty} \frac{g(\epsilon)}{N^2} \overset{\epsilon \to 0}{\sim} \epsilon^{D_2}.$$

On the other hand, the SOMOS output equation (4), provided Ω increases with N, forces the output y to take nonzero values on a space of integer dimension closest to D_2. This way, the SOMOS architecture provides a sort of local integer approximation of the correlation dimension.

The obtained neural spacial distribution (with the corresponding number of connections for each neuron) can be employed to size the window (r) of a recurrent predictor of the type of (1). In addition, the local number of connections associated with the activated neuron for the input vector $(x_t, x_{t-1}, \ldots, x_{t-r+1})$ provides an idea of the uncertainty in the predicted value $\hat{x}_{t+1} = G(x_t, x_{t-1}, \ldots, x_{t-r+1}, 0)$: a high local connectivity indicates a local dimension higher than r (suggesting that more than r values should be employed for a proper prediction); hence, the predicted value must be considered with higher uncertainty.

4 Simulation Examples

4.1 Series Based on Hénon Map

Let us consider the time series $\{x_0, \ldots, x_t\}$ generated by measuring the first component of the underlying Hénon map:

$$x_{t+1} = y_t + 1 - a x_t^2, \quad y_{t+1} = b x_t, \tag{6}$$

which happens to be chaotic for $a = 1.4$ and $b = 0.3$. In this case, since (3) is defined by (6) we have that $r = 2$; in fact, it is known that the map evolves in a set whose correlation dimension is $D_2 \approx 1.42$.

The time series values $\{x_1, \ldots, x_{10000}\}$ have been generated according to (6) with $x_0 = y_0 = 0.5$. Then, sets of the form (5) (with almost 10000 vector values) have been constructed for different values of d; then, they have been employed to train a SOMOS architecture. Note that the number of connections N_c of each neuron with neighbor ones gives an approximation of the local dimensionality of

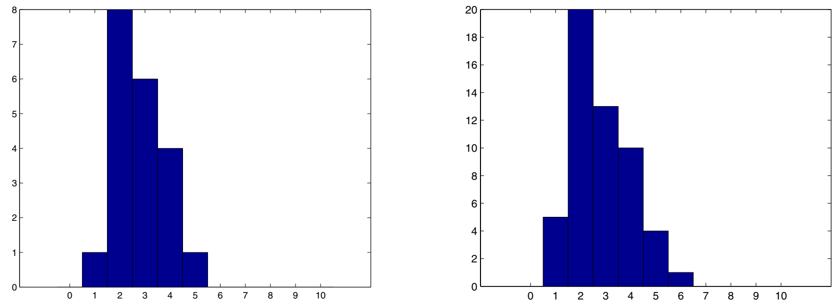


Fig. 1. Distribution of number of connections for Hénon map with $d = 3$ and $d = 5$

the attractor ($d_e = \frac{N_c}{2}$), so that average embedding dimension can be estimated as half the average of connections.

Figure 1 shows the distribution of the number of connections associated with all the generated neurons for $d = 3, 4$ and 5, the average number of connections being 2.8, 2.94 and 2.83 respectively. This suggests $d_e \approx 1.428$, which is a very good approximation of the previously mentioned known correlation dimension.

4.2 Series Based on Lorenz System

We consider now the Lorenz system:

$$\frac{dx}{dt} = \sigma(y(t) - x(t)); \quad \frac{dy}{dt} = x(t)(a - z(t)) - y(t); \quad \frac{dz}{dt} = x(t)y(t) - bz(t); \tag{7}$$

with $a = 45.92$, $b = 4$ and $\sigma = 16$, so that it shows chaotic behavior [3]. The discrete time series is obtained by numerically integrating (7) with the standard Runge-Kutta method in the interval $[0, 500]$ with initial conditions $(0.1, 1.1, 1)$.

Figure 2 shows the distribution of the number of connections associated with all the generated neurons for $d = 4$ and $d = 5$. The average number of connections

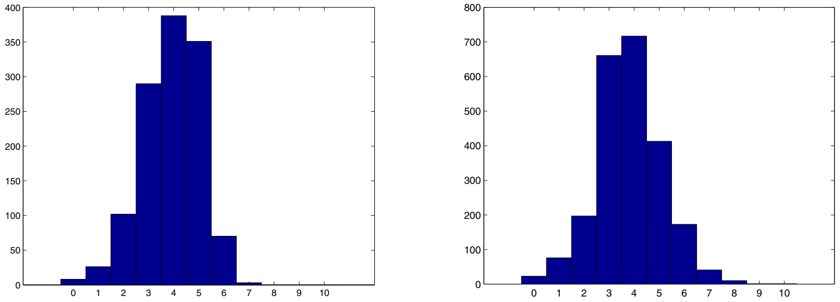


Fig. 2. Distribution of number of connections for Lorenz system with $d = 4$ and $d = 5$

is 3.92 and 3.81 respectively. Such values suggest $d_e \approx 1.94$, which is a good approximation of the known correlation dimension of the attractor ($D_2 \approx 2.02$).

5 Concluding Remarks

The SOMOS neural architecture, initially proposed in [2], has been successfully applied for approximating the distribution of the embedding dimension associated with a time series. The results provided by SOMOS can be used to size the window in predictor recurrent architectures; also, they can be employed to assess the quality of the prediction steps. The performance of the proposed scheme has been illustrated by analyzing two time series based on different chaotic systems.

Acknowledgments. This work has been partially supported by projects MTM2007-62064 and DPI2007-66156 of the Plan Nacional de I+D+i, MEyC, Spain, and by projects CCG07-UPM/000-3278 and Ayudas 2008 of the Universidad Politécnica de Madrid (UPM) and CAM, Spain.

References

1. Ataei, M., Lohmann, B., Khaki-Sedigh, A., Lucas, C.: Model based method for estimating an attractor dimension from uni/multivariate chaotic time series with application to Bremen climatic dynamics. Chaos, Solitons and Fractals 19, 1131–1139 (2004)
2. Campoy, P.: Dimensionality Reduction by Self Organizing Maps that Preserve Distances in Output Space. In: International Joint Conference on Neural Networks, Atlanta (in press, 2009)
3. Frank, R.J., Davey, N., Hunt, S.P.: Input Window Size and Neural Network Predictors. In: IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJNN 2000), vol. 2, pp. 2237–2242 (2000)
4. Grassberger, P.: Generalized dimensions of strange attractors. Physics Letters A 97(6), 227–230 (1983)
5. Jiayu, L., Zhiping, H., Yueke, W., Zhenken, S.: Selection of Proper Embedding Dimension in Phase Space Reconstruction of Speech Signals. Journal of Electronics 17(2), 161–169 (2000)
6. Packard, N.H., Crutchfield, J.P., Farmer, J.D., Shaw, R.S.: Geometry from a time series. Phys. Rev. Lett. 45, 712–716 (1980)
7. Small, M.: Applied Nonlinear Time Series Analysis: Applications in Physics, Physiology and Finance. World Scientific Series on Nonlinear Science, Series A, vol. 52 (2005)
8. Takens: Detecting strange attractors in turbulence. In: Rand, Young (eds.) Dynamical Systems and Turbulence, Warwich. Lecture Notes in Mathematics, vol. 898, pp. 366–381 (1981)

Training Methods and Analysis of Composite, Evolved, On-Line Networks for Time Series Prediction

Russell Y. Webb

University of Canterbury, Christchurch, New Zealand
`r.webb@elec.canterbury.ac.nz`
`http://www.elec.canterbury.ac.nz/people/webb.shtml`

Abstract. New results for online prediction using predictive networks composed of smaller prediction units are presented. Strategies for choosing training signals across a range of signal types are discussed. Composite networks are shown to generalise across a wide range of test signals. The best network found by a genetic evolution is present, simplified, and analysed.

1 Overview and Background

Merging insights from the fields of neuroevolution[1] and reservoir computing[2], previous work by Webb[3] introduced the use of echo state networks (ESN)[4] organized into and larger network to predect times sequences. The new contributions of the present work are as follows:

- Refinement of the training and testing methods.
- Unified output equation to produce a general time-series predictor.
- Demonstration of the difference between sequential and parallel update.
- Simplification and analysis of the best three-node network found.

2 Composite Predictive Networks

A composite predictive network (CPN) is a network composed of smaller predictive nodes sending signals amongst themselves and finally producing a prediction of a target time series. In this work, the smaller predictive units used are echo state networks, but that choice is arbitrary. The major requirement of the CPN structure is that each smaller predictive unit receive two input series and learn to predict two arbitrary target signals. The schematic in Figure 1 shows how these predictive nodes are drawn and roughly how an ESN is used to implement the node. A network of k nodes based on the architecture presented here will be called a k-net for brevity.

The ESN simulations used here are largely identical to those from the earlier work by the author [3,5] and utilises ESN simplifications similar to those found

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1176–1183, 2009.
© Springer-Verlag Berlin Heidelberg 2009

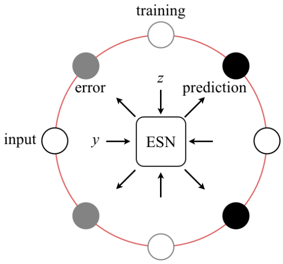


Fig. 1. Schematic form of a prediction node with two inputs and two predictions

in the literature [6,7,8]. Each ESN contains 40 reservoir neurons and uses a *tanh* activation function; other parameters are tuned by a genetic algorithm. The results presented here all involve 3-nets, composed of an input signal, x_t, network of nodes, zero signal (not shown) that allows inputs to be unused, and combining weights used to form the final prediction of x_{t+1}.

The most interesting side-effect of evolved or random network connectivity is that it is no longer clear how to form a prediction from the available signals. In earlier work, each output was a correction term relative to the previous, so a simple summation produced a viable prediction. For k-nets, a linear filter was implemented to combine all available signals into a prediction. This combining filter is learned on-line, in parallel with the predictor. The final prediction from the k-net is $\hat{x}_{t+1} = \sum_{i=1}^{4k} f_i s_i$ where f is a value in the output filter and s is a signal produced by one of the nodes. For the 3-net, there are 12 node outputs (4 from each node composed of a prediction and error for each training input).

3 Evolution of Solutions

A huge search space led to the use of a genetic algorithm (GA) for optimisation. It is hoped that solutions provided by the GA will provide insight into the composite prediction task and speed the search of the entire space by providing heuristic simplifications. A fairly conventional GA algorithm (using the typical, mutation and crossover operations on a population size of 70) was implemented and reported previously in detail [3]. Since then, the training method, testing approach, and fitness function have been refined as discussed in the following sections.

3.1 Signals for Training and Testing

The fitness function used to rank solutions in the evolving population is the total training error on a subset of available signals. In previous work, the training and test signal was always one term from the Mackey-Glass (MG) oscillator [9] with these parameters: $\alpha = 0.2, \beta = -0.1, d = 17$. The assumption was that the prediction network evolved to predict the MG signal would be suitable for general signals because of the wide spectral range and complexity of the signal. However, experiments showed that evolved predictors, while working well on MG prediction, did not generalise to other signals. Analysis showed that the combining weights at the output were primarily responsible for the over specialisation of the predictor.

In order to train the combining weights, the set of training signals shown below were introduced. Any subset of them can be used as training signals:

$$MG = Mackey - GlassPosition \tag{1}$$

$$MG_2 = tanh(Mackey - GlassVelocity) \tag{2}$$

$$Sin = 0.5sin(0.18t) \tag{3}$$

$$+Sin = 0.5sin(2\pi t/17) + 0.4sin(2\pi t/29) \tag{4}$$

$$Sin = 0.8sin(2\pi t/14) * sin(2\pi t/19) \tag{5}$$

$$LMap = LogisticsMap(k = 3.59) \tag{6}$$

$$Spike = 0.8sin(2\pi t/4.1) * sin(2\pi t/3.79). \tag{7}$$

The training method for $\boldsymbol{f}$ was also changed. To train and calculate the fitness, identical CPNs are created for each training signal. These CPNs are trained in parallel (4,800 training steps and 1,200 test steps) while sharing a common set of combining weights to form the output. This setup unites the wiring of the CPN with the combining weights representing the utility of each signal in the network. In addition, the GA was given control over spectral radius of the nodes and the learning rates; which means that the full specification of a CPN is given by the following: spectral radius and learning rate of each ESN, learning rate for the combining weights, and the connectivity between nodes.

4 Parallel versus Sequential Update

The other major change to the system was a move from parallel update to sequential update of the nodes in the CPN. Parallel update had been used in the past, but it was realised each node was getting information produced at the previous time step (the only *current* information was the target signal value). The previous study [3] compared the CPN implementation of a cascade error correction network to published results and found the cascade error corrector to perform better (however, a parallel update CPN was found that outperformed the cascade using a GA). The major reason for the poorer result was that parallel update prevented information from flowing from one node to another within a processing step.

In order to achieve the desired data-flow behaviour, sequential update of the nodes was introduced. The problem of deadlock between nodes that are each waiting for the other was solved by defining a fixed update order (N_1, N_2, N_3) and implementing buffering on all inputs. Each input records the last incoming value, so that an input from a node that has not been processed will simply be the value produced on the previous time step. This update strategy is capable of representing all pure data-flow networks as well as those using buffering of past values.

Figure 2 shows the result of running the GA with both update policies. Even though the sequential strategy has a much larger search space (since it matters what order the nodes are listed in), the sequential update is seen to produce better predictions (here trained on the MG signal). Parallel systems are usually thought of as being more powerful, but in this case the two policies are implementing different types of algorithms, and the sequential is more powerful because it is combinational within an update step.

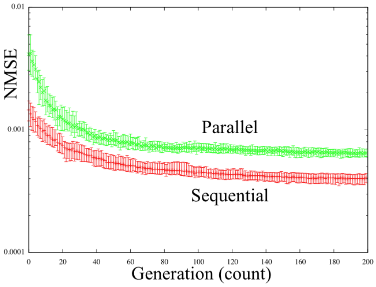


Fig. 2. Prediction performance achieved from evolving solutions using parallel and sequential update rules. The error bars show one standard deviation from the mean value.

5 Training Strategies

The goal of the GA is of course determined by its fitness function and does not yet represent the full aim of the study. The final goal of the study is to determine what makes a CPN capable of predicting a wide range of signals. Training the CPN on a wide range of signals is the obvious way to achieve this goal. However, some signals need to be reserved for testing so that specialisation to the training set can be distinguished from generalisation. Figure 3 shows a typical result of running a 200-generation GA with 11 different training sets (a - k) and testing the best solutions from the final population with all signals. Dots in the chart mark the signals used in training, and the testing performance against each signal is shown by the linear shading. Within a column, white represents the best predictor and black the worst. The NMSE used to measure prediction error was capped at 1 to prevent a few extremely poor performers from washing out the chart (i.e. black shows the worst solutions without allowing comparison the NMSE in those cells).

The first column (a) is the average across the rows. Expectedly, training on all the signals produces a predictor that works well for all signals (but might be specialised just to the training set), and the diagonal dots show that training on one signal always finds a good predictor for that signal. The block of slowly varying signals (b-f) show a high degree of similarity; in that, training on one seems to work well for all ($*Sin$ being a the least similar). Of these smooth training signals, Sin produces the best predictor of the $Sharp$ signals.

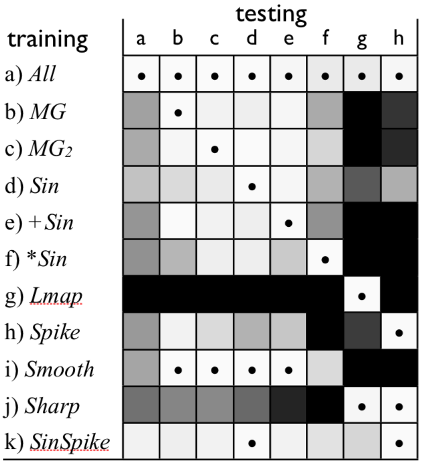


Fig. 3. Training and testing relation. Dots show signals used for training and columns show relative testing performance (white is best).

An inspection of results from single-signal training, led to several experiments shown in the final rows of the chart. First, all but one of the smooth signals were used in training and showed that smooth signals do not seem to selected predictors that can deal with abrupt changes. Second, selecting both *Sharp* signals did not produce CPNs capable of predicting smooth signals. Finally, based on the best results from the two groups, one from each was selected (*Sin* and *Spike*). This combination successfully produced predictors across the entire range of signals, indicating that the goal of finding general predictors is feasible.

6 Simplification of Solutions

In order to identify a good, simple k-net, the GA was used to build a library of solutions achieved after 200 generations. The total compute time for the library of solutions was on the order of weeks on a 3 GHz dual core desktop machine. Each collected solution was then tested over 100 trials and the solution which minimised the NMSE across all the signals was selected. Even though the best solution contained only error propogation signals (no predications were propogated, though many good solution in the library do propogate predictions), this network still seemed too complex to analyse.

A culling process was used to rank the contributions of each linkage in the network to the final prediction. At each step, the least important link is removed and the error of the simpler, resulting network is recorded. Two links are needed for the network to perform a useful predictions: an ESN-input and a training-input of the target signal. Figure 4 shows the recorded error curve as the connections are removed from the best 3-net found and labels the links in the original network by ranked importance (i.e. link 12 will be removed first and link 1 is the last to be removed).

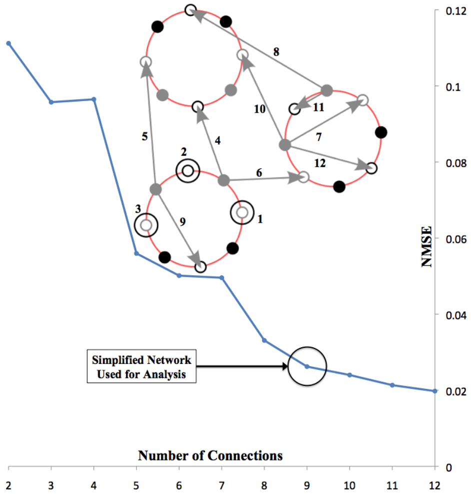


Fig. 4. The prediction error achieved when the least useful link is iteratively removed from the network. Links are numbered in order of decreasing significance.

7 Analysis of Solutions

The resulting simplified network is shown in Figure 5 along with the combining weights that minimise NMSE over all signals. The role of N_2 is clear; it receives input from the target signal and tries to predict the target (see circled inputs). The predictions it makes are then averaged together and the previous errors it made are partially added. The first node processed, N_1, oddly gets a training input from the second, so this input is buffered by the sequential update algorithm arriving at the next time step. N_1 has no inputs so its internal activation is zero which means its error is exactly the training signal (which it predicted to be zero in the previous step). That error is then passed through the other training input via another buffer. The signal that comes out of N_1 to N_3 is the error of one of the N_2 predictions delayed by two time steps. In other words, the network has implemented a two step delay line. Taking both N_1 and N_2 into account the system is averaging two predictions and adding a filtered combination of previous errors:

$$\hat{x}_{t+1} = \frac{x'_p + x_p}{2} - 0.3x'_e - (0.3, 0.14, -0.2) \cdot (x_e, x_{e,t-1}, x_{e,t-2}). \qquad (8)$$

The last node in the sequence, N_3, is not fully understood. It receives as input errors of two related but not identical predictions; one is the error of predicting the current target, and the other is the error two steps in the past. So in addition to ESN memory it has strong gradient information about the prediction error. From this, it predicts the error of the current prediction (just like the hand

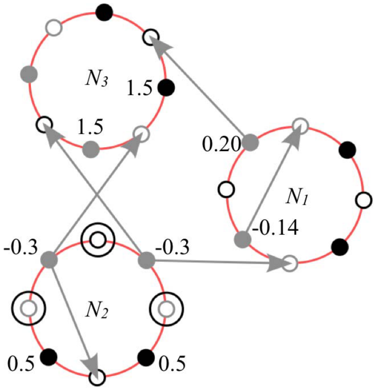


Fig. 5. Simpler network resulting from removing the three least useful links. Combining weights are shown next to each link.

designed cascade error correctors that started this work). The combining weights (listed in Figure 5) are surprising though. Perhaps the magnitude is less important than fact that the weight are equal. The node adds a prediction of error and simultaneously subtracts the error of that prediction from the last time step.

The resulting learning rate and spectral radius parameters also remain to be understood. For the network discussed above the combining filter learning rate was 0.27, while the nodes had the following parameters (note, the increase in learning rate with decreasing radius):

	radius	learning rate
N_1	0.95	0.13
N_2	0.92	0.28
N_3	0.71	0.37

8 Conclusions

The results presented here are a promising step toward understanding evolved generalised composite prediction. Networks which combine all the available internal signals in the same manner to predict a wide range of signals have been demonstrated. Future work needs to address many open questions, including determining how the type sub-nodes influences the CPN, developing analytic techniques, incorporating a FIR filter model of error correction, and understanding the optimal tuning of learning rate and memory for each predictive node. Ultimately, it is hoped that insights gained from evolved CPNs will lead to a better understanding of predictive systems; in the same way, the study of evolved organisms is a guide to the broader field of adaptive systems.

References

1. Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. Evolutionary Intelligence 1, 47–62 (2008)
2. Schrauwen, B., Verstraeten, D., Van Campenhout, J.: An overview of reservoir computing: theory, applications and implementations. In: Proceedings of the 15th European Symposium on Artificial Neural Networks, pp. 471–482 (2007)
3. Webb, R.Y.: Time series prediction with evolved, composite echo state networks. In: Xiaodong (ed.) SEAL 2008. LNCS, vol. 5361, pp. 555–564. Springer, Heidelberg (2008)
4. Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science 304(5667), 78–80 (2004)
5. Webb, Russell: Multi-layer corrective cascade architecture for on-line predictive echo state networks. Applied Artificial Intelligence 22(7), 811–823 (2008)
6. Gao, H., Sollacher, R., Kriegel, H.P.: Spiral recurrent neural network for online learning
7. Fette, G., Eggert, J.: Short term memory and pattern matching with simple echo state networks. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 13–18. Springer, Heidelberg (2005)
8. Cernansky, M., Makula, M.: Feed-forward echo state networks. In: IJCNN 2005. Proceedings. 2005 IEEE International Joint Conference on Neural Networks, vol. 3, pp. 1479–1482 (2005)
9. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science 197(4300), 287–289 (1977)

Special Time Series Prediction: Creep of Concrete

Juan L. Pérez[1], Fernando Martínez-Abella[2], Alba Catoira[1], and Javier Berrocal[1]

[1] RNASA-IMEDIR group, Information and Communication
Technologies Research Centre (CITIC),
University of A Coruña, Campus de Elviña, A Coruña, Spain
[2] Construction group - gCONS, Higher Technical University
College of Civil Engineering,
University of A Coruña, Campus de Elviña, A Coruña, Spain
`{jlperez,fmartinez,acatoira}@udc.es,`
`javierberrocalbenito@gmail.com`

Abstract. This paper presents an algorithm, different from the classical time series, specialised in extracting knowledge from time series. The algorithm, based on Genetic Programming, enables the dynamic introduction of non-terminal operators shaped as mathematical expressions (operator-expression) that works as an unique node for the purpose of genetic operations (crossover and mutation). A new characteristic of this algorithm is the possibility of expansion the individuals, which, besides inducing a better global fitness, enables breaking up the expressions (operator-expression) into basic operators in order to achieve expression recombination. The performance of the implemented algorithm was showed by means of its application to the creep of structural concrete, a specific case of Construction Engineering where a best adjustment to the current regulative codes was subsequently achieved.

Keywords: Data Mining, Evolutionary Computation, Genetic Programming, Civil Engineering.

1 Introduction

The vast amounts of data that currently exist for every specific subject are, in most of cases, quite difficult to be interrelated. The discovery and understanding of such interrelation is usually achieved by knowledge extraction techniques supported by Artificial Intelligence mechanisms [1-6]. These techniques are known as Data Mining[4]; they use several data analysis tools in order to discover data patterns and connections that might be used for performing valid predictions. As in other study areas, the type of problems in Construction Engineering (framed into the Civil Engineering) is to achieve knowledge extraction (by using mathematical models) from the different databases that contain trials related to those physical phenomena affecting the constructive processes. The predictions are achieved in this field by using knowledge inductive techniques on data. The goal achievement might be interestingly supported by Data Mining techniques focused on regression problems. There are currently several approaches that fulfil these tasks as the Artificial Neural Networks (ANN)[1] or other techniques based on Evolutionary Computation (EC) [2]. This paper presents an

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1184–1191, 2009.
© Springer-Verlag Berlin Heidelberg 2009

algorithm, different from the classical time series, specialised in extracting knowledge from time series. This algorithm uses Genetic Programming (GP) [3], a specific Data Mining technique, and it also enables the insertion of operators as mathematical expressions (named operator-expression) previously decided by the expert. An interesting innovation is the possibility of expansion the operator-expression group (mathematical functions previously inserted by experts) for disaggregating the expressions into basic operators; this fact enables the recombination of the expressions by using the basic operations of the evolutionary algorithms (crossover and mutation). The algorithm performance has been showed by its application to the creep of structural concrete, a specific case of Construction Engineering.

2 State of the Art

There exist currently numerous articles related to the application of Data Mining techniques to Civil Engineering aspects [7-10]. The Construction Technology is a study field where theses techniques are especially interesting, as the knowledge extraction techniques can be used for improving, in the broad sense of the term, the constructive process, which extends from structural design to the mixture procedure used for obtaining concrete. In this field there are several studies that attempt the extraction of knowledge from a specific aspect. I-Cheng Yeh [11] uses ANN and GA jointly for obtaining optimum concrete mixtures; the ANN extracts knowledge from the composition of concrete mixtures, in order to achieve the performance model for every mixture element (Cement, Fly Ash, Slag, Water, Super Plasticizer…). The EC techniques were used for searching the optimum mixture of concrete composition, a mixture with the lowest cost and required performance.

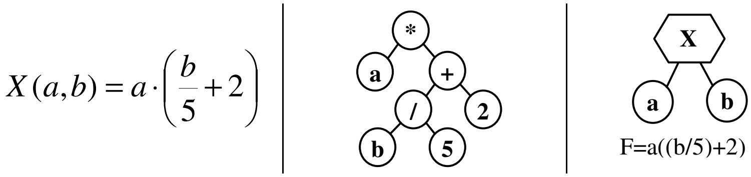
Other example of application of Data Mining techniques in Construction Technology that should be highlighted is the work of Cladera and Marí [12]. They use of AI techniques in the development of new expressions and regulative codes. Cladera and Marí studied, by means of ANNs, the shear strength response of concrete beams without web reinforcement. These studies were based on experimental data gathered by Bentz and Kuchma, who identified 5 variables that had influence on the process and on the beam fracture values. The ANNs were used in this case as a virtual lab where predicting values of not physically performed trials.

3 Development System

There are different Data Mining strategies to solve a specific problem. The main goal of Data Mining is obtaining new knowledge by creating a real world model after data compiled from several heterogeneous sources [4]. The Data Mining techniques can be applied for value prediction and, according the required prediction (commonly classification and regression), specific techniques exist that obtain new knowledge. The present work is focused on data symbolic regression: the prediction of values for a given variable according certain experimentation data. The regression uses the existing values in order to predict new coming ones.

The technical approach of the situation offers several options. The standard statistical techniques, as lineal regression, are useful for simple cases but unfortunately the majority of problems are quite complex and they need other suitable techniques. The ANNs have shown their capability for regression tasks, as it was mentioned in the previous section. The main disadvantage of the ANNs use in Construction is that they work as black boxes: after being well trained they can offer reliable predictions but they cannot explain how they obtain them. This fact, together with the difficult determination of the optimal configuration for the network, is the reason why the experts do not use ANNs more frequently in the Construction field. Other Data Mining technique is the Genetic Programming, included itself within the Evolutionary Computation. Among other qualities, the GP can algebraically connect input data (in Construction they would be those ones coming from the trials performed) with the objective result. In this way it is able to model with mathematical equations the physical-chemical phenomenon deduced from the trial data.

However, in many of the cases studied in Civil Engineering, the GP cannot create a valid predictive model by using the classical operators (addition, subtraction, multiplication, etc.) Besides, as in most of the studied phenomena there are formulations capable of modelling the solution partially, it would be quite interesting to insert in GP formulas that might work as compact operators, performing as an unique node. For example, for problem where it is known - due to experience or to intuition- that the X formula (Figure 1.1) contributes somehow to its solution; in this case, the X non-terminal operator is defined in GP for working as another operator that similarly will be comprised of two variables (a and b) coming from the trials database. At the GP level the X operator would not be modified, as it would not be affected by the crossover and mutation operations. Internally, the X operator works as it is shown in Figure 1.



1)Expression 2)Equivalent tree 3)Notation

Fig. 1. Example of new expression-operator

The developed system is based on the possibility of inserting any operator-shaped mathematical expression to the use of GP techniques. In this way the GP algorithm could use the classical operators together with dynamically inserted expressions (operator-expression set) for creating/mutating new individuals. As a consequence of GP own nature, elements of the operator-expression set will be used provided that the resulting individuals achieve good fitness.

Figure 2 shows an outline of the developed algorithm. The set operation-expression consists of the operators defined by the user.

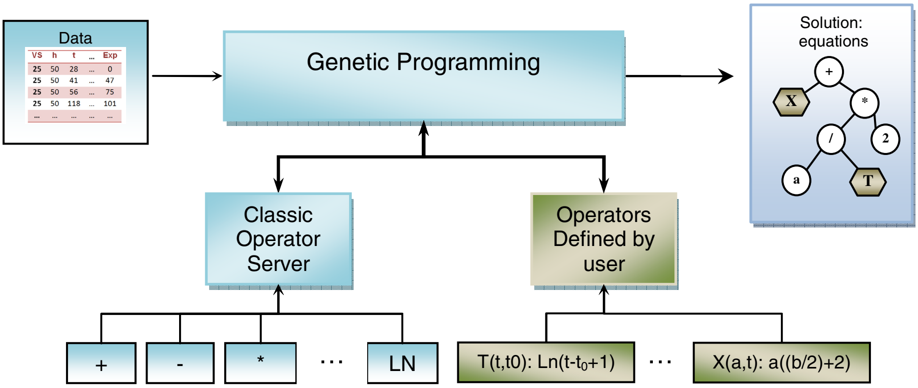


Fig. 2. Diagram of the developed Algorithm

For a given individual, every operator-expression would be considered as a unique node; in this way, the quantity of the respective sub-tree nodes will not account for individual size penalty when calculating the fitness.

Other interesting characteristic is that the set operator-expression of the individuals can be expanded in any generation. Every operator-expression (Figure 1.3) turns into classical operators (addition, subtraction, etc.) (Figure1.b); this fact brings about the expressions to be recombined or modified in future generations.

The addition of operator-shaped expressions results quite suitable for studying data with temporal behaviour, as the expressions are able of roughly modelling such behaviour and, subsequently, the explorations of the results can be directed. Lastly, it should be highlighted that the expansion capability of the expressions qualify them, resulting in an improved fitness.

4 Example

Once the algorithm has been developed, it is interesting its application to a Civil Engineering case, more specifically to the complex modelling problem of the creep phenomenon of structural concrete under constant load and in controlled lab conditions. This phenomenon is observed by studying the deformation evolution experienced by a cylindrical specimen under compression throughout time (Figure 3). In this type of trials the initial elastic deformation (Figure 3, time t_t variable ε_i) is followed by an increasing deformation (creep-induced deformation ε_t) that tends to an asymptotic value.

A database with experimental series of 185 trials was used for training and test periods. The database belongs to the Réunion Internationale des Laboratoires et Experts des Matériaux, systèmes de construction et ouvrages - RILEM (International union of laboratories and experts in construction materials, systems and structures) [13] and contains characteristic data of every trial (non time-dependent data) together with data related to the moment when the measurements were registered. The first data group includes volume/surface rate (V/S), relative humidity(h%), age drying commenced (end of moist curing) (t_c), Age of concrete at loading(t_0), concrete type, cement content (c), water content (w) , aggregate content (a), deformation module in the initial load age and concrete compression strength after 28 days(fcm_{28}). The time-dependent data include the day when the measurement was registered (t) and the deformation produced from the beginning of the trial(J). There are 2323 total readings. Figure 4 shows 4 examples of real data with different age and type of concrete; as it can be noticed, the curves are similar to those of Figure 3.

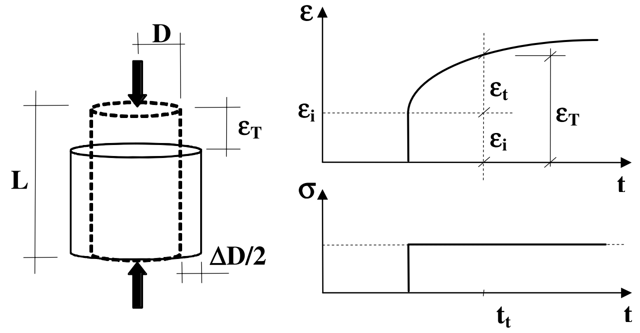


Fig. 3. Creep phenomenon

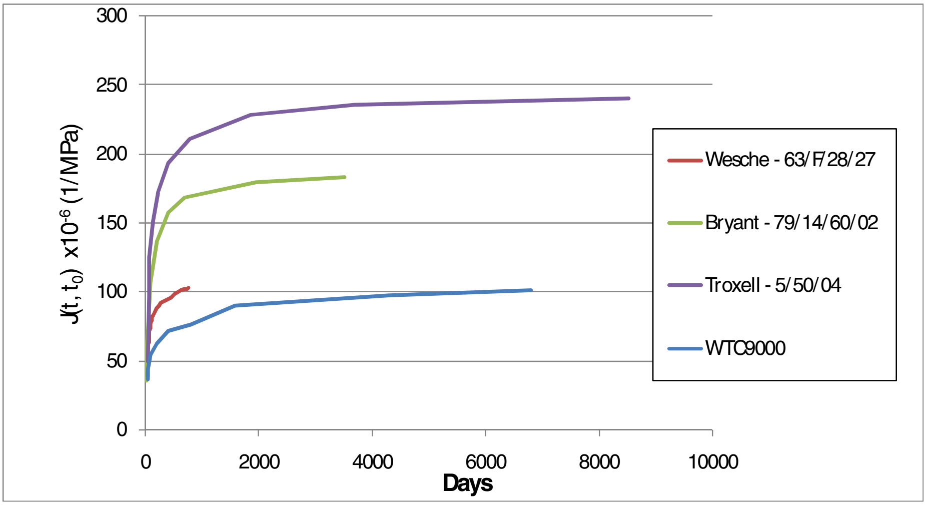


Fig. 4. Example of data

After carrying out several tests, and together with the help of the area experts, it was not only determined a group of variables, but also a set of expressions for studying the concrete creep phenomenon. Table 1 shows the ranges of the variables used: for the operator-expression set it was defined the *TemporalOperator* function as *ln* $(t-t_0+1)$ in the group of tests performed with non-standardised data and the *TOStandardised* as *ln((t*10000)-(to*3600)+1)* for the group of test performed with standardised data

Table 1. Variables used

Variables	Range
V/S (mm)	15.5 – 200
h%	20 – 100
t_c (days)	0.4 – 400
t_0 (days)	0.5 – 3300
t (days)	0.5 - 8525
c Type	1 – 3
c (kg/m^3)	275 - 564
w (kg/m^3)	129 - 251
a (kg/m^3)	1661- 2110
fcm_{28} (MPa)	17.2 - 119
$J(t, t_0)$- $J(i_{nitial})$ x10^{-6} (1/MPa)	0 – 262

The non-terminal nodes used during the tests were: addition, subtraction, product, protected division, power and the previously defined operator-expression group. The variables from the database (Table 1) and the constants included within the -1 and 1 interval were used for the terminal nodes.

Table 2 shows the configuration parameters used during all the executions.

Table 2. Configuration parameters used

Configuration parameters	Value
Selection algorithm	Tournament
Mutation rate	20
Crossover rate	95
Population size	400
Maximum tree height	7
Parsimony	0.1 (Non-standardised data) 0.000001 (Standardised data)

5 Results

The results obtained were classified according the use of standardised or non-standardised data. The second group was intentionally set in order to minimally to change the resulting formulas, which then might be directly applied on data by the Construction experts.

The expansion operation of the operator-expression set was also applied after obtaining the results. Table 3 shows the results of the two best individuals of each type. The errors of the standardised data were made non-standardised for the data to be compared.

Table 3. Configuration parameters used

	Training data without Standardisation process				Training data with Standardisation process				International regulative codes[1]	
	F1		F2		**F3**		F4		CEB[14]	ACI[15]
	Nor.	Exp.	Nor.	Exp.	Nor.	**Exp.**	Nor.	Exp..		
ME	11.402	10.703	12.257	10.492	10.370	**8.467**	10.388	9.508	13.673	14.691
MSE	314.960	263.068	347.2762	239.185	269.526	**187.911**	255.518	209.132	410.461	549.389
R^2	0.737	0.768	0.687	0.763	0.750	**0.798**	0.736	0.784	0.782	0.764

6 Conclusion

An expression named as F3 was obtained for predicting the concrete creep phenomenon. This formula obtains the best result when using the RILEM database. From Civil Engineering viewpoint the best obtained result might be used (according the experts of the area) as concrete creep model for the structural concrete regulations, as it improves the results than the current regulations (CEB and ACI) obtain.

As it was expected, the best fitness were obtained with standardised data; also the good choice of the data set is a key factor for obtaining good results in this type of problems.

The expansion possibility improves the results substantially. This characteristic enables the introduction of a group of hypotheses (more or less valid) that the own nature of GP will refine.

As in other areas, the close collaboration between Computing Engineers and Civil Engineers is a key point for success.

The GP is a technique suitable for the symbolic regression in many engineering fields.

With the system developed for the study of concrete, a wide range of possibilities appear to be open for study due to the existence of several databases.

Acknowledgements. This work was partially supported by the Spanish Ministry of Development (Ministerio de Fomento) (Ref. CLEAM-CENIT, AIE), grants (Ref. 202/PC08/3-03.2) funded by the Spanish Environmental Ministry (Ministerio de Medio Ambiente) and grants from the General Directorate of Research, Development and Innovation (Dirección Xeral de Investigación, Desenvolvemento e Innovación) of the Xunta de Galicia (Ref. 07TMT011CT, Ref. 08TIC014CT and Ref. 08TMT005CT). The work of Juan L. Pérez is supported by an FPI grant (Ref. BES-2006-13535) from the Spanish Ministry of Science and Innovation (Ministerio de Ciencia e Innovación).

References

1. Graupe, D.: Principles of Artificial Neural Networks. World Scientific, Singapore (1997)
2. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Michigan (1975)

---

[1] Mathematical equations in the Civil Engineering regulative codes or rules in creep phenomenon.

3. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
4. Cios, K., Pedrycz, W., Swiniarrski, R., Kurgan, L.: Data Mining: A Knowledge Discovery Approach. Springer, New York (2007)
5. Brown, M., Harris, C.: Neurofuzzy adaptive modelling and control. Prentice-Hall, Hertfordshire, UK (1994)
6. Rivero, D., Rabuñal, J.R., Dorado, J., Pazos, A.: Time Series Forecast with Anticipation Using Genetic Programming. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 968–975. Springer, Heidelberg (2005)
7. Öztaş, A., Pala, M., Özbay, E., Kanca, E., Çağlar, N., Bhatti, M.A.: Predicting the compressive strength and slump of high strength concrete using neural network. Constr. Build. Mater. 20, 769–775 (2006)
8. Morcous, G., Lounis, Z.: Maintenance optimization of infrastructure networks using genetic algorithms. Automat. Constr. 14, 129–142 (2005)
9. Ashour, A.F., Alvarez, L.F., Toropov, V.V.: Empirical modelling of shear strength of RC deep beams by genetic programming. Comput. Struct. 81, 331–338 (2003)
10. Kicinger, R., Arciszewski, T., De Jong, K.A.: Evolutionary computation and structural design: a survey of the state of the art. Comput. Struct. 83, 1943–1978 (2005)
11. Yeh, I.-C.: Computer-aided design for optimum concrete mixtures. Cement. Concrete. Comp. 29, 193–202 (2007)
12. Cladera, A., Marí, A.R.: Shear design procedure for reinforced normal and high-strength concrete beams using artificial neural networks. Part I: beams without stirrups. Eng. Struct. 26, 917–926 (2004)
13. Réunion Internationale des Laboratoires et Experts des Matériaux, systèmes de construction et ouvrages, `http://www.rilem.net`
14. ACI Committee 209: Prediction of Creep, Shrinkage and Temperature Effects in Concrete Structures. ACI 209-82. American Concrete Institute, Detroit (1982)
15. Müller, H.S., Hilsdorf, H.K.: Evaluation of the Time Dependent Behavior of Concrete. CEB Comite Euro-International du Beton. Bulletin d'Inforrnation No 199. France (1990)

Artificial Neural Networks in Urban Runoff Forecast

Mónica Miguélez[1], Jerónimo Puertas[2], and Juan Ramón Rabuñal[1,2]

[1] Department of Information and Communication Technologies, University of Coruña,
Campus de Elviña, A Coruña, Spain
[2] Centre of Technological Innovations in Construction and Civil Engineering (CITEEC),
University of Coruña, Campus de Elviña, A Coruña, Spain
{mmiguelez,jpuertas,juanra}@udc.es

Abstract. One of the applications of Data Mining is the extraction of knowledge from time series [1][2]. The Artificial Neural Networks (ANNs), one of the techniques of Artificial Intelligence (AI), have proved to be suitable in Data Mining for handling this type of series. This paper presents the use of ANNs and Genetic Algorithms (GA) with a time series in the field of Civil Engineering where the predictive structure does not follow the classic paradigms. In this specific case, the AI technique is applied to a phenomenon that models the process where, for a specific area, the fallen rain concentrates and flows on the surface.

Keywords: Artificial Neural Network, Genetic Algorithms, Civil Engineering.

1 Introduction

In many fields of Civil Engineering the most common way of designing models and structures is application of the laws of physics; experimentation, trial resulting data for their study and validation of the results are the following steps. However, the AI advances have had strong influence on different areas of Civil Engineering, as engineers can use these techniques in several ways and in very different problems.

The present work shows the application of ANNs and GA techniques to a specific field of Civil Engineering, the Hydrology, and more specifically to the modelling of water flow, generated after a rain event, in a given basin. The several ANNs architectures used and the results obtained with them are following presented in order to illustrate the process.

2 State of Art

One of the most important processes in Hydrology is the so-named "Rainfall-Runoff transformation process" [3], meaning the process in which the rain fallen over an area concentrates and runoff-flows over the surface. As the importance of this process lies basically in it being the cause of overflows, the fact of having a model for the rainfall-runoff transformation process enables forecasting the increase of water levels due to rain episodes.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1192–1199, 2009.
© Springer-Verlag Berlin Heidelberg 2009

2.1 Current Models for Rainfall-Runoff Transformation Process

In the field Civil Engineering there are currently several methods based on mathematical equations for modelling the rainfall-runoff process; some of them are the Hydraulic Equations and the Unit Hydrographs.

The Hydraulic Equations try to model, by means of physical equations, the hydraulic processes occurred in a given basin that has been figured as a group of "channels" that represent the roofs, the streets and their interconnections, etc.

These models are based on Saint-Vennant equations [4] and they are used quite frequently; however, they are complex to use and also require a wide knowledge about the basin to be modelled.

The Unit Hydrographs are based on a common representation of how rain fall-flow is generated.

They are based on experimental model, being the SCS Unitary Hydrogram the most used one.

2.2 AI Techniques

There is other type of predictive methods, not based on physical equations, than merely try to provide a formulation that might be used as forecasting tool: the ANNs and the GAs.

An ANN is an information processing system based on biological concepts related to human brain and to neuronal biology [5]. An ANN works as a black box, producing outputs according the inputs it receives. An ANN can "learn" how to model a process and it has been proved that its predictive capability might be quite high.

There are algorithms for training classic ANNs [6], however in some cases, or if some of the ANN elements is modified, these algorithms become invalid; in such cases, a generic GA-based training method can be used.

A GA is a search algorithm supported by Darwin's Theory of Evolution (1859) according the Principle of Natural Selection [7][8]. According to this, it has been observed that the sexual reproduction and the Principle of Survival of the Fittest enable the biological species, not only to adapt themselves to their environments, but also to compete for achieving the resources available. With regards to computing, the algorithm works through the evolution of different generations of individuals created from other previous ones.

3 Development

This section describes the specific problem "rainfall-runoff transformation" that is going to be tackled by means of ANNs. Once the problem has been described, the different used ANNs architectures will be showed, together with the different modifications that might be required for them to achieve the best results.

3.1 Problem Description

It is intended to measure and predict the runoff flow produced by the rain in Vitoria basin; a pluviometer will be used for measuring the rainfall of the city and a flow sensor will be also placed in the lowest point of the sewer system. The system will be therefore modelled by using both signals, which will be obtained in 5-minutes intervals.

However, despite having the two mentioned signals, the final objective is to predict the flow produced in Vitoria using only the pluviometry signal.

Fig. 1 shows the pluviometry and flow signals (5426 data points – 19 days) that will be used respectively as inputs and outputs for ANN training process.

Fig. 3 shows the pluviometry and flow signals (250 data points – 20 hours) that will be used for the validation process of the ANN system in order to test that the obtained model works correctly.

Both signals were normalised for applying on them the ANNs.

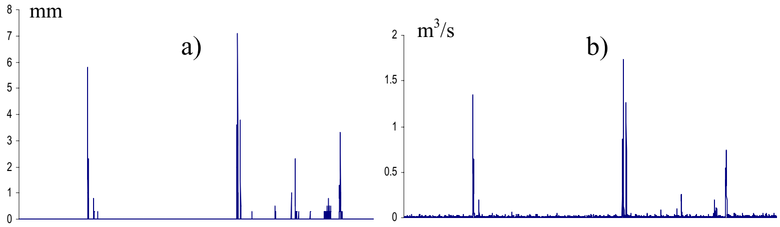


Fig. 1. Pluviometry signal (a) and flow signal (b) used for training the ANN system (5426 points)

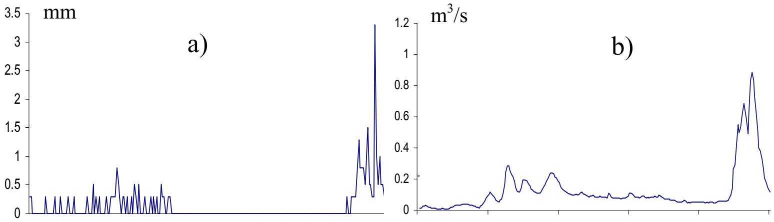


Fig. 2. Pluviometry signal (a) and flow signal (b) used for testing the ANN system (250 points)

3.2 ANNs Application

The first possibility handled when trying to solve this problem by using ANNs is to choose a classic feed-forward ANN with a time window that might consider the flow as a time series. The ANN will use as inputs the flow values at the current and previous moments (as many as the time window indicates) and it will obtain as output the flow value at the following moment.

For instance, when using a feed-forward ANN with a size 3 time window, given the X_1, X_2,…,X_n time series (where the X_{n+1}value should be predicted), the ANN inputs would be the current values of the series together with the three previous ones, whereas the output will be the value at the following moment. Fig. 3 shows a possible architecture of this ANN.

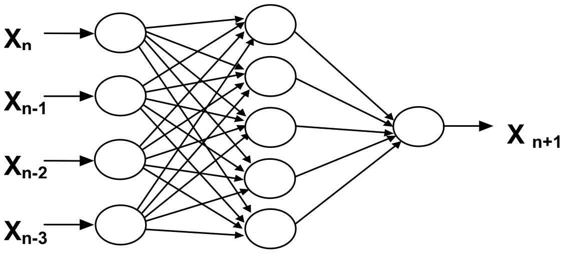


Fig. 3. ANN architecture

Many solutions to prediction problems use feed-forward ANNs with time window [9]; however, for the "Rainfall-Runoff transformation" problem, considering flow as a time series is not completely correct, as it does not present the typical characteristics of a time series, but it is a signal that depends on rain: time after rainfall has occurred and due to several factors (previous rainfalls, floor permeability, etc.) the flow is generated. Besides the fact that using this type of ANNs is not possible to make long-term prediction, it should be also mentioned that obtaining flow data is quite costly, whereas obtaining the pluviometry data is more easy and inexpensive from the viewpoint of Civil Engineers.

Once discarded the previous option, the following one is the application of an ANN with shifting time window using only pluviometry data. It should be highlighted that it is an atypical case, as the pluviometry signal is not a time series. The ANN is provided with the current, and some previous (as the time windows indicates) rainfall values as inputs; the output value will be the current flow value.

Fig. 4 shows an example of the pluviometry signal and of the signal of runoff flow generated by rainfall. Next, it can be observed an ANN with size 3-time window applied on the pluviometry that might predict the generated flow: the ANN inputs are the four last recorded pluviometry values, and the output is the value of the generated flow.

Nevertheless, when applying this type of ANN architecture on the real data of Vitoria basin, the results obtained are not satisfactory, as the ANN needs to know some data of previous flow in order to predict the resulting flow value. There are two options for solving this: on one hand by providing as input not only the pluviometry data required by the time window, but also the value of the previous flow; on the other hand by using an ANN with recurrent architecture (RANN), where the training process implies the identification of the values of previous flows in order to achieve a correct prediction.

In the first option, after several tests the suitable value of the time window was found to be 5 (the lag time of basin is 20-25 minutes, then 5 previous values represent 25 minutes). Different ANN architectures were also tested bearing in mind the configuration of the layers, the neuron number per layer and the activation function. Fig. 5 presents the ANN architecture that achieves the best results. Fig. 6 shows the comparison of the fitness obtained after using the ANN with validation data or with test data. Numerically, the Mean Square Error (MSE) obtained is 0.00013, and the Mean Error (ME) is 0.0101 when using the validation data. However, these good results are due to the use of the value of the previous flow as one of the inputs, which, as it was previously mentioned, is not adequate. Given that the ANN will learn to use the value of the past flow for achieving the value of the current flow, it will find difficult to reach long term predictions, as it will not count with such information; besides, and as it was mentioned, obtaining flow data result quite costly.

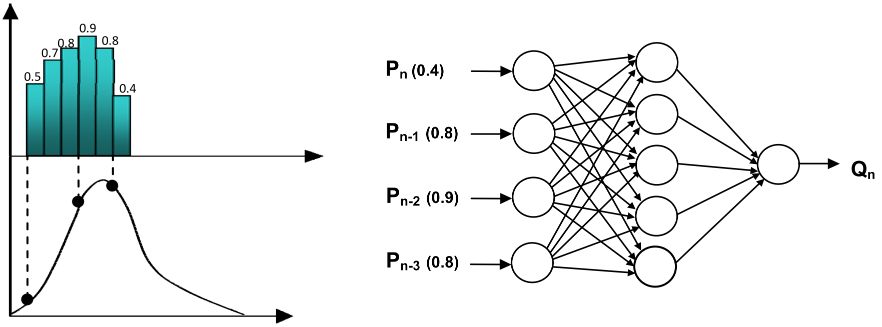


Fig. 4. Pluviometry signal and runoff flow signal – ANN with time window using pluviometry data

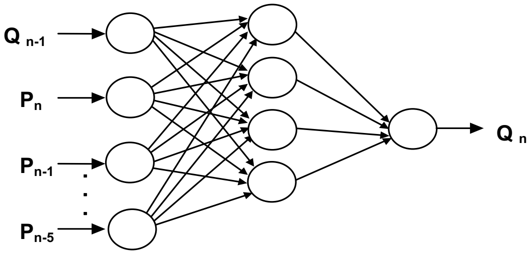


Fig. 5. ANN architecture with time window (size 5)

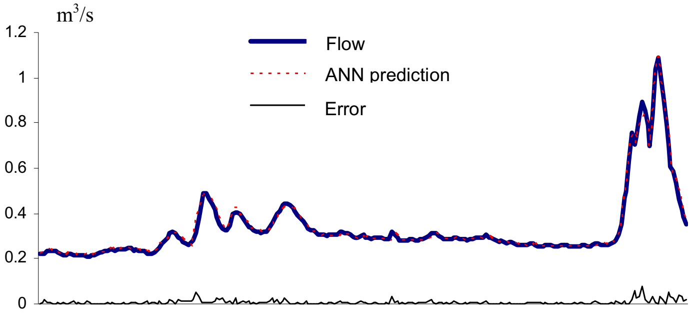
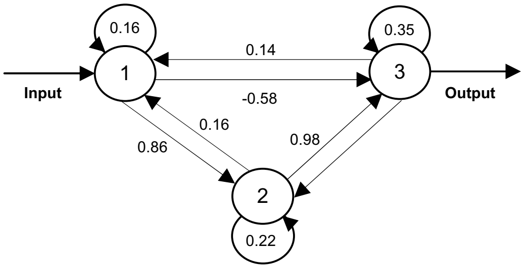


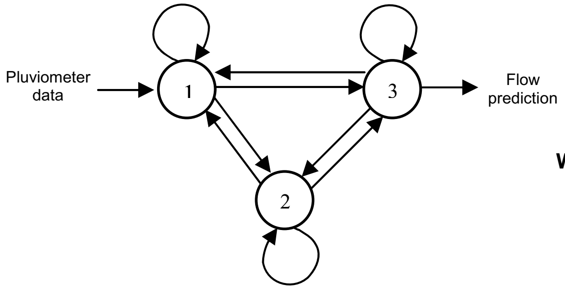
Fig. 6. Flow and prediction of the ANN architecture with temporal window (size 5)

In the second proposed option, the use of RANN, the GA has been used for its training, as they are the unique technique that can be used for training any ANN architecture. In this way, as the ANN is proposed to be encoded as a GA individual, during the GA training it will search for the ideal weights in the connections of the ANN neurons. Fig. 7 shows the matrix notation that has been.



		Input neurons		
		1	*2*	*3*
Output neurons	*1*	0.16	0.86	-0,58
	2	0.16	0.22	0.98
	3	0.14	-0.91	0.35

Fig. 7. RANN and the encoding of its connections



Weight matrix

-0.922	0.760	0.217
-0.927	0.730	0.723
-0.071	0.968	0.267

Fig. 8. Architecture of the RANN 1

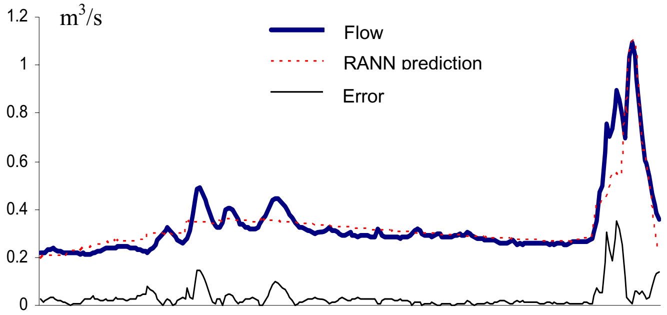


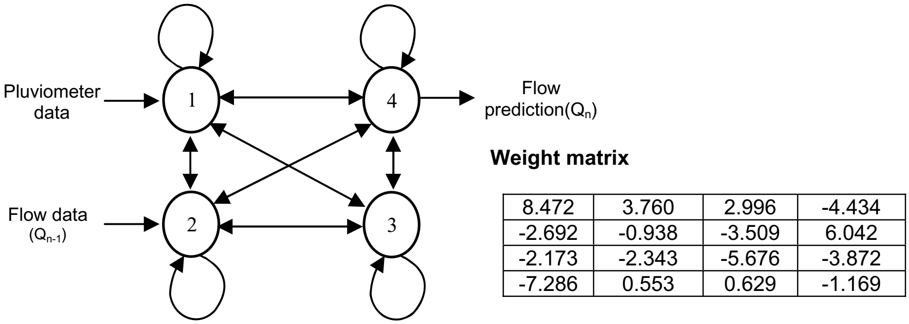
Fig. 9. Flow and RANN 1 prediction with test data

In the first approach carried out in Vitoria basin with GA-trained RANN, the ANN was solely provided with the pluviometry data. After testing several architectures and several activation functions it was stated that the RANN that achieve better results include 3 neurons (1 input, 1 output and 1 hidden) and a lineal function as activation function, as it can be observed in Fig. 8.

This RANN, named as RANN1, produces 0.0037 MSE and 0.0343 ME with validation data. Fig. 9 presents the comparison between the real flow and the value predicted by the RANN with validation data.

If, together with the pluviometry data, the flow data of the previous moment is also provided to the GA-trained RANN, it can be obtained a better fitness value but that flow data is always required. Fig. 10 shows the RANN that better results has achieved in this case: it has 4 neurons (2 input, 1 output and 1 hidden) with hyperbolic tangent activation function.

This RANN (named as RANN 2) produces 0.00038 MSE and 0.0098 ME with validation data. The comparison between the real flow and the value obtained when using validation data can be observed in Fig. 11.



Weight matrix

8.472	3.760	2.996	-4.434
-2.692	-0.938	-3.509	6.042
-2.173	-2.343	-5.676	-3.872
-7.286	0.553	0.629	-1.169

Fig. 10. Architecture of the RANN 2

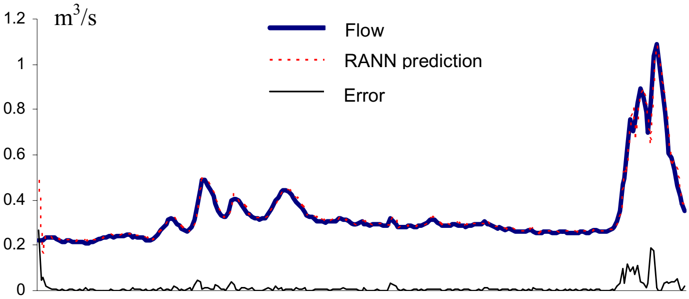


Fig. 11. Flow and RANN 2 prediction with test data

4 Conclusion

The results achieved by ANNs when modelling the runoff flow of a basin by using either feed-forward architecture with time window or recurrent architecture, show that they have a satisfactory performance.

Table 1 shows a comparison of the different ANN architectures used; as it can be noticed, the RANN has good results by merely using the pluviometry data but the feed-forward ANN improves the previous results by adding the real value of previous flow and a time window of pluviometry data. However, as it was previously indicated, the flow data is difficult and expensive to obtain and therefore is not a viable input from the view point of Civil Engineers. In this way, the architecture that provides the optimal solution by merely using pluviometry data is, in this case, the recurrent architecture.

Table 1. Comparison among the different ANN architectures used

	ANN time window	*Recurrent ANN 1*	*Recurrent ANN 2*
Inputs	P_n, P_{n-1}, P_{n-2}, P_{n-3}, P_{n-4}, P_{n-5}, Q_{n-1}	$\mathbf{P_n}$	P_n, Q_{n-1}
ME	0.0101	**0.343**	0.0098
MSE	0.0001	**0.0037**	0.0003

Acknowledgments. This work was partially supported by Ministerio de Ciencia e Innovación (Spanish Ministry of Science and Innovation) (Ref CGL2007-64155) and by several grants from the Dirección Xeral de Investigación, Desenvolvemento e Innovación (General Directorate of Research, Development and Innovation) de la Xunta de Galicia (Galicia regional government) (Ref. 08MDS003CT, Ref. 08TMT005CT and Ref. 08TIC014CT). The work of Mónica Miguélez is supported by a FPI grant (Ref. BES-2008-002519) from the Ministerio de Ciencia e Innovación (Spanish Ministry of Science and Innovation).

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2006)
2. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2006)
3. Viessmann, W., Lewis, G.L., Knapp, J.W.: Introduction to Hydrology. Harper Collins (1989)
4. Huber, W.C., Dickinson, R.E.: Storm Water Management Model, user's manual, version 4. U.S. Envir. Protection Agency, Athens, Ga (1992)
5. Kohonen, T.: Self-Organization and Associative Memory, 2nd edn. Springer, New York (1998)
6. Freeman, J., Skapura, D.: Neural Networks algorithms, applications and programming techniques. Addison-Wesley, Reading (1991)
7. Darwin, C.: On the origin of species by means of natural selection or the preservation of favoured races in the struggle for life, 6th edn. Cambridge University Press, Cambridge (1864) (original published in 1859)
8. Darwin, C., Wallace, A.R.: On the tendency of Species to form varieties: and on the perpetuation of varieties and species by Natural Means of Selection. Journal of the Proceedings of the Linnean Society of London. Zooly 3, 46–50 (1858)
9. Atiya, A., El-Shoura, S., Shaheen, S., El-Sherif, M.: A comparison between neural-network forecasting techniques. Case study: River flow forecasting. IEEE Transactions on Neural Networks 10, 402–409 (1999)

A Secret Sharing Scheme for Digital Images Based on Cellular Automata and Boolean Functions

Ángel Martín del Rey[1] and Gerardo Rodríguez Sánchez[2]

[1] Department of Applied Mathematics, E.P.S. de Ávila, Universidad de Salamanca
C/Hornos Caleros 50, 05003-Ávila, Spain
`delrey@usal.es`
[2] Department of Applied Mathematics, E.P.S. de Zamora, Universidad de Salamanca
Avda. Cardenal Cisneros 34, 49022-Zamora, Spain
`gerardo@usal.es`

Abstract. In this paper an efficient probabilistic secret sharing scheme for black and white digital images is introduced. It is an extension of the algorithm proposed originally in [11]. Specifically, it is a (m, n)-threshold scheme. It is based on the use of a simple matrix boolean function which involves the Hadamard matrix product modulo 2, and the addition modulo 2 and m. Moreover, an algorithm based on cellular automata is used for the enhancement of the recovered image. The algorithm introduced is shown to be secure and its expansion factor is equal to 1.

Keywords: Secret sharing schemes. Boolean functions. Cellular automata. Image enhancement.

1 Introduction

The worldwide distribution of digital information with the advent of computers and Internet is easy and economic. Nevertheless there are several security problems associated with the processing and transmission of digital images over an open network: It is necessary to assure the confidentiality, the integrity and the authenticity of the digital image transmitted. To meet these challenges, a wide variety of cryptographic protocols have been proposed. One of the most important cryptographic schemes for digital images are secret sharing algorithms.

Secret sharing algorithms are multi-party cryptographic protocols originally related to key establishment. More precisely, they allow one to share a secret among a set of participants such that only qualified subsets of these participants can recover the original secret. Moreover, no information about it can be obtained when non-qualified subsets of participants try to recover the secret. The original motivation for secret sharing was to safeguard cryptographic keys from loss. Currently, they have many applications in different areas such as access control, opening a safety deposit box, etc. The basic example of secret sharing algorithms is the (m,n)-threshold scheme, where m and n are integer numbers such that $1 \leq m \leq n$. Secret sharing schemes are methods by which a third trusted party (called the dealer) computes n secret shares from an initial secret S and securely distributes them among the n participants. Only subsets of m or more of these participants who pool their shares may easily recover the original secret, but any group knowing only $m - 1$ or fewer shares are unable to recover the secret.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1200–1207, 2009.
© Springer-Verlag Berlin Heidelberg 2009

A secret sharing scheme is said to be perfect if the shares corresponding to each unauthorized subset of participants provide absolutely no information about the secret *S*. Moreover, when the sizes of the secret and the shares are equal, the secret sharing scheme is called ideal (this property is called "pixel expansion" when *S* is a digital image). Other interesting property when original secret is a digital image is the reconstruction precision ("contrast" or "accuracy").

These schemes were introduced independently by Shamir (see [9]) and Blakley (see [2]) and they are based on the use of Lagrange interpolation polynomial and the intersection of affine hyperplanes, respectively. Since then, several proposals have been appeared in the literature based on different mathematical primitives: matrix theory, prime numbers, etc. These protocols are specially designed for digital data instead of digital images. Due to the main characteristics of digital images (they have a large amount of data and the difference between two neighboring data is very small), it is very difficult to apply directly traditional secret sharing schemes to digital images.

The first proposal to share digital images was due to Naor and Shamir (see [7]) and it was called Visual Cryptography. It is based on visual threshold schemes *k* of *n*, i.e. the secret image is divided in *n* shares such that each of them is photocopied in a transparency and then, the original image is recovered by superimposing any *k* transparencies but no less. In the last years several construction methods based on Visual Cryptography have been proposed (see, for example, [4, 6]). Besides the Visual Cryptography, another protocols to share digital images have been designed based on vector quantization ([3]), Shamir's ideas ([10]), Sharing circle ([8]), cellular automata ([1]), etc. An important type of secret sharing algorithms for images are those based on the probabilistic paradigm. They are called probabilistic visual schemes (see, for example, [5, 11, 12]) and they use simple computations and have no pixel expansion. These methods encode each original pixel into one pixel per share. In the recovered image, not every single pixel will be exactly reconstructed, but "areas" in the recovered image look similar to the original (they suffer from a certain loss of reconstruction accuracy).

In [12] a probabilistic visual (2, *n*)-threshold scheme was proposed for black and white images. The goal of this paper is to extend such scheme to a (*m*, *n*)-threshold scheme such that only subsets of *m* participants can recover the original image. Moreover a method to enhance the recovered image is proposed based in cellular automata paradigm.

The rest of the paper is organized as follows: In Section 2 an overview of the mathematical tools used is introduced; in Section 3 the algorithm and its analysis is detailed with an example; the image enhancement protocol is shown in Section 4, and finally, the conclusions are presented in Section 5.

2 Mathematical Background

2.1 Hadamard Product

Definition. Suppose $A=\left(a_{ij}\right)$ and $B=\left(b_{ij}\right)$ are two $r \times c$ -matrices with entries in a field. Then their *Hadamard product* is the entry-wise product of *A* and *B*, that is, the $r \times c$ -matrix $A \circ B$ whose (i, j)-th entry is $a_{ij} \cdot b_{ij}$.

It is easy to check the following result:

Proposition. Let *A*, *B* and *C* be three matrices. The Hadamard matrix product satisfies the following properties:

1. It is commutative: $A\circ B=B\circ A$.
2. The distributive property of the sum with respect the Hadamard product holds: $A\circ(B+C)=(A\circ B)+(A\circ C)$.

This work deals with binary matrices and the arithmetic is performed in $\mathbb{F}_2$. In this case, the last properties also hold but using the XOR operation (⊕), which is the sum modulo 2 instead of the traditional sum +.

2.2 Bidimensional Cellular Automata

Bidimensional cellular automata (CA for short) are finite state machines formed by a finite number of *r*×*c* identical objects called cells which are arranged uniformly in a two-dimensional cellular space (see, for example, [13]). Each cell is endowed with a state (0 or 1), that changes at every step of time according to a local transition rule. In this sense, the state of a particular cell at time *t* depends on the states of a set of cells, called its neighborhood, at the previous time step *t* – 1. More precisely, a CA is defined by the 4-uplet (*C*, $\mathbb{F}_2$, *V*, *f*), where *C* is the cellular space: $C=\left\{(i,j),1\le i\le r,1\le j\le c\right\}$; $\mathbb{F}_2$ is the finite state set whose elements are the all possible states of the cells; $V=\left\{(\alpha_k,\beta_k),1\le k\le n\right\}\subset\mathbb{Z}\times\mathbb{Z}$, is the finite set of indices defining the neighborhood of each cell, such that the neighborhood of the cell (i,j) is $V_{ij}=\left\{(i+\alpha_1,j+\beta_1),\ldots,(i+\alpha_n,j+\beta_n)\right\}$. Finally, the local transtion function is the boolean function $s_{ij}^t=f\left(s_{i+\alpha_1,j+\beta_1}^{t-1},\ldots,s_{i+\alpha_n,j+\beta_n}^{t-1}\right)\in\mathbb{F}_2$, where $1\le i\le r,1\le j\le c$ and s_{ij}^t stands for the state of the cell (i,j) at time *t*. Let V_{ij}^t be the set given by the states of the neighbor cells of (i,j) at time *t*. The matrix $C^t=\left(s_{ij}^t\right)_{1\le i\le r,1\le j\le c}$ is called the configuration at time *t* of the CA.

The most important type of neighborhood is Moore neighborhood which is given by the cell itself and its eight nearest cells. That is, its set of indices is defined as follows: $V=\{(-1,-1),(-1,0),(-1,1),(0,-1),(0,0),(0,1),(1,-1),(1,0),(1,1)\}$.

As is mentioned above, the CA evolves deterministically in discrete time steps, changing the states of the cells by means of the local transition function *f*. As the cellular space is considered to be finite, boundary conditions must be considered in order to assure a well-defined dynamics of the CA. These boundary conditions depends on the process to be simulated.

Finally, every black and white image can be represented in terms of a binary matrix. Let *I* be a black and white image defined by *r* × *c* pixels, then its associated binary matrix is the $r\times c$-matrix $A=\left(a_{ij}\right)_{1\le i\le r,1\le j\le c}$ where:

$$a_{ij} = \begin{cases} 0, & \text{if the } (i,j)\text{-th pixel is white} \\ 1, & \text{if the } (i,j)\text{-th pixel is black} \end{cases}$$

Note that a black and white image can be considered as a configuration at a particular time step of a CA.

3 The Probabilistic Secret Sharing Algorithm

The probabilistic secret sharing algorithm consists of three phases: the setup phase, the sharing phase and the recovery phase.

3.1 The Setup Phase

In this phase the dealer defines the parameters of the protocol: n+1 random binary matrices and a specific boolean function. Specificially,

- The dealer computes n + 1 random binary matrices of order $r \times c$:
$$M_1 = \left(m_{ij}^1\right), \ldots, M_n = \left(m_{ij}^n\right), Z = \left(z_{ij}\right).$$
- The dealer considers the following boolean matrix function: $f(X,Y) = (X \circ Y) \oplus Z.$

3.2 The Sharing Phase

In this phase the dealer constructs the n shares and they are securely distributed among the participants.

Let $A = \left(a_{ij}\right)$ be the binary $r \times c$ -matrix representing a black and white image I to be shared. Then, the shares are the following: $S_i = f\left(M_i, A\right), 1 \le i \le n.$ Note that each share is a matrix which stands for a binary black and white image.

3.3 The Recovery Phase

As the protocol proposed is a probabilistic secret sharing scheme, we will not recover exactly the same original image but a very similar one, $\tilde{I}$, whose associated binary matrix is $\widetilde{A}$. It is obtained as follows: Let $S_{i_1}, S_{i_2}, \ldots, S_{i_m}$ be the m shares, then we perform the following computation:

$$\begin{aligned} \left(\sum_{k=1}^{m} S_{i_k} \,(\operatorname{mod} m)\right)(\operatorname{mod} 2) &= \left(\sum_{k=1}^{m} f\left(M_{i_k}, A\right)(\operatorname{mod} m)\right)(\operatorname{mod} 2) \\ &= \left(mZ + \sum_{k=1}^{m}\left(M_{i_k} \circ A\right)(\operatorname{mod} m)\right)(\operatorname{mod} 2) = \sum_{k=1}^{m}\left(M_{i_1} \circ A\right)(\operatorname{mod} 2) \\ &= A \circ \sum_{k=1}^{m}\left(M_{i_1}\right)(\operatorname{mod} 2) = A \circ \widetilde{M} = \widetilde{A}. \end{aligned}$$

The matrix $\widetilde{A}$ is not equal to the matrix *A* but they are very similar since $\widetilde{M}$ is a random binary matrix and the Hadamard product is used, that is, the (i,j)-th coefficient of $\widetilde{A}$ is:

$$\widetilde{a}_{ij} = \widetilde{m}_{ij} \cdot a_{ij} = \begin{cases} 0\left(= a_{ij}\right), & \text{if } a_{ij} = 0 \\ 1\left(= a_{ij}\right), & \text{if } a_{ij} = 1 \text{ and } \widetilde{m}_{ij} = 1 \\ 0\left(\neq a_{ij}\right), & \text{if } a_{ij} = 1 \text{ and } \widetilde{m}_{ij} = 0 \end{cases}$$

As a consequence if the (i,j)-th pixel of the original image *I* is a white pixel ($a_{ij} = 0$), then the (i,j)-th pixel of the recovered image $\widetilde{I}$ is a white pixel, whereas if the (i,j)-th pixel of the original image is black ($a_{ij} = 1$), the (i,j)-th pixel of the recovered image can be white or black depending on the value of $\widetilde{m}_{ij}$. As $\widetilde{M}$ is a random binary image, then $\widetilde{m}_{ij} = 1$ in 50% of the cases and $\widetilde{m}_{ij} = 0$ in the other 50%. Consequently, if *A* has the same number of black pixels and white pixels, the 75% of the pixels of *A* and $\widetilde{A}$ are equal. In general, if *w* is the number of white pixels of *A*, the percentage of equal pixels in both images is $\frac{50\left(r \cdot c + w\right)}{r \cdot c}$ %, which is called the estimated value of coincidences.

3.4 An Illustrative Example

Let *I* be a black and white image defined by 83×77 pixels (see Figure 1 – (a)). In this case there are four participants and *m* = 2, and consequently four shares are computed (see Figure 1 –(b)-(e)). If the first two shares, S_1 and S_2, are pooled together, the

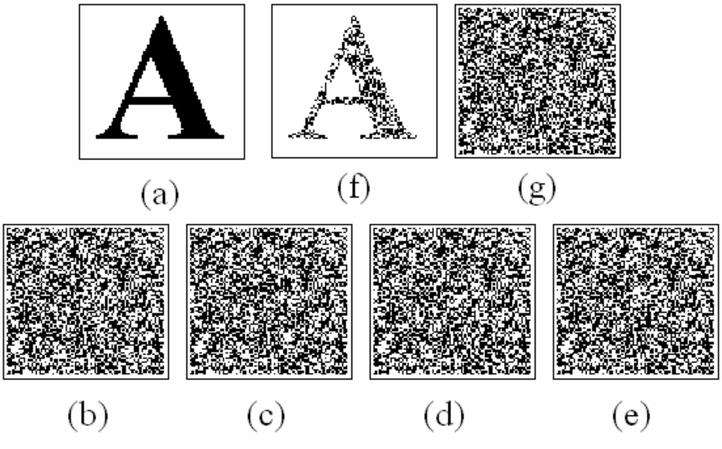


Fig. 1. (a) Original image *I* defined by 83×77 pixels; (b)-(e) The four shares computed: S_1, S_2, S_3 and S_4; (f) Recovered image using the first two shares: S_1 and S_2; (g) The recovered image using three shares: S_1, S_2 and S_3.

obtained original image is $\tilde{I}$ (see Figure 1-(f)). On the other hand, if an odd number of shares are used to try to recover the original image, for example S_1, S_2 and S_3, the result is the hardly noise image given in Figure 1-(g).

The percentage of coincident pixels between $\tilde{I}$ and I is 89,94%. Note that the estimated value is of 90,05%.

3.5 Security Analysis

The proposed scheme is perfect since no information is obtained when an odd number of shares are used to recover the original image (see Figure 1 –(g)).

The following procedure will be carried out to test the correlation between two adjacent pixels in the original and the shared images: First of all randomly select 1.000 pairs of two adjacent pixels from the image, and then, calculate the correlation coefficient of each pair by using the following formula:

$$r_{xy} = \operatorname{cov}(x,y) \big/ \sqrt{D(x)} \cdot \sqrt{D(y)},$$

where x and y are the grey-scale values of the two adjacent pixels in the image and:

$$\operatorname{cov}(x,y) = \sum_{i=1}^{N} \frac{(x_i - E(x)) \cdot (y_i - E(y))}{N}, \quad E(x) = \sum_{i=1}^{N} \frac{x_i}{N}, \quad D(x) = \sum_{i=1}^{N} \frac{(x_i - E(x))^2}{N}.$$

As a consequence, the results obtained are shown in Table 1.

Table 1. Correlation coefficients of Example 1

	I	S_1	S_2	S_3	S_4
Horizontal	0.7597	0.0275	0.0379	0.0415	0.0238
Vertical	0.9325	0.0095	0.0459	0.0214	0.0099
Diagonal	0.7419	0.0194	0.0139	–0.0024	0.0038

The correlations coefficients of the original image and shared images are far apart (note that the correlation coefficients of the original picture are close to 1, whereas the corresponding coefficients of the shared images are very close to 0). Consequently, the secret sharing algorithm satisfies zero co-correlation.

4 Image Enhancement

As the algorithm proposed in this work is probabilistic, the image recovered is not exactly the same than the original one. This recovered image can be enhancement by means of a suitable cellular automata such that its initial configuration is the matrix defined by the B&W image $\tilde{A}$. The neighborhood of the CA used is the Moore neighborhood, the boundary conditions are reflexive boundary conditions, and the local transition function is the following:

$$s_{ij}^t = \begin{cases} 0, & \text{if } O_{ij}^t < u \\ 1, & \text{if } O_{ij}^t \geq u \end{cases}$$

where $O_{ij}^{t} = \{\text{number of } 1's \text{ in } V_{ij}^{t}\}$ and u is a predefined threshold with $0 \le u \le 9$.

In Figure 2 we show the enhancement image obtained from the image in Figure 1-(f) and using the last mentioned CA with $u = 2$.

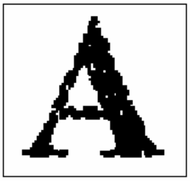


Fig. 2. Image obtained when the CA with $u = 2$ is applied once from $C^0 = \widetilde{A}$. The percentage of coincidences is 96.65%.

5 Conclusions

In this paper a modification of the original work due to Wang et al. (see [11]) is introduced. Specifically, we propose a probabilistic secret (m, n)-threshold scheme in addition with an enhancement final phase based on cellular automata. It is shown that the scheme obtained is perfect and ideal. Moreover, its statistical security is also shown.

Acknowledgements

This work has been supported by Ministerio de Ciencia e Innovación (Spain) under grant MTM2008-02773.

References

1. Álvarez, G., Hernández, L., Martin, A.: A multisecret sharing scheme for color images based on cellular automata. Inform. Sci. 178(22), 4382–4395 (2008)
2. Blakley, G.R.: Safeguarding cryptographic keys. In: AFIPS Conf. Proc., vol. 48, pp. 313–317 (1979)
3. Chang, C., Hwang, R.: Sharing secret images using shadow codebooks. Inform. Sciences 111, 335–345 (1998)
4. Chen, Y., Chan, Y., Huang, C., Tsai, M., Chu, Y.: A multiple-level visual secret-sharing scheme without image size expansion. Inform. Sciences 177, 4696–4710 (2007)
5. Cimato, S., De Prisco, R., De Santis, A.: Probabilistic visual cryptography schemes. Comput. L. 49(1), 97–107 (2006)
6. Cimato, S., De Prisco, R., De Santis, A.: Colored visual cryptography without color darkening. Theor. Comput. Sci. 374, 261–276 (2007)
7. Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
8. De Santis, A., Masucci, B.: New results on non-perfect sharing of multiple secrets. J. Syst. Software 80, 216–223 (2007)

9. Shamir, A.: How to share a secret. Commun. ACM 22, 612–613 (1979)
10. Wang, R., Su, C.: Secret image sharing with smaller shadow images. Pattern Recognition Lett. 27, 551–555 (2006)
11. Wang, D., Zhang, L., Ma, N., Li, X.: Two secret sharing schemes based on Boolean operations. Pattern Recognition 40(10), 2776–2785 (2007)
12. Yang, C.N.: New visual secret sharing scheme using probabilistic method. Pattern Recongnition Lett. 25(4), 481–494 (2004)
13. Wolfram, S.: A new kind of science. Wolfram Media, Champaign, Illinois (2002)

Shapes Description by a Segments-Based Neural Network

J.A. Gómez-Ruiz, J. Muñoz-Perez, and M.A. García-Bernal

Dept. of Languages and Computer Science, University of Málaga,
29071 Málaga, Spain
`{janto,munozp,magb}@lcc.uma.es`

Abstract. Skeletonization is the process of transforming a shape in a digital image, composed of several pixels, to a line-based shape so that the topological properties of the original shape are preserved. The resulting shape is called skeleton. Such skeletons are useful in the recognition of elongated shaped objects, for example, character patterns, chromosome patterns, etc. The skeleton provides an abstraction of geometrical and topological features of the object, so that the skeletonization can be viewed as data compression. In this paper, a model of competitive neural network based in segments is proposed. This model is suitable for obtaining the skeleton of elongated shapes with an unsupervised method.

Keywords: skeletonization, competitive neural networks, unsupervised learning, shapes description.

1 Introduction

Pattern recognition is a discipline that has been in continuous expansion during the last years, especially in image analysis and computer vision. Artificial neural networks are a powerful tool within this area, especially Kohonen's self-organizing feature maps (SOFM) [1] due to their ability to capture the topology and probability distribution of input data. The SOFM have been used by several authors to perform invariant pattern recognition [2-5].

Skeletonization is the process of transforming a shape in a digital image, composed of several pixels, to a line-based shape so that the topological properties of the shape are preserved. The resulting shape composed of lines is called skeleton. The skeleton provides an abstraction of geometrical and topological features of the object.

The skeleton concept was introduced by Blum in 1964 [6] and since then, a great number of techniques for its detection have been developed. These techniques differ from each other in performance and in implementation [7]. In the last decade, with the improving of artificial neural networks, several algorithms for skeleton detection have been proposed [7-10]. The majority of these techniques are based on SOFM with several modifications depending on the kind of the shape being studied.

With the segments-based competitive neural network proposed by the authors in this paper, the skeleton of shapes is detected without any previous classification or distinction and with an unsupervised method.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1208–1215, 2009.
© Springer-Verlag Berlin Heidelberg 2009

The paper is organized as following: in section 2 the segment based competitive neural network is proposed; in section 3 the process of obtaining the skeleton for shapes description is explained and, finally, in section 4 conclusions are presented.

2 The Segments-Based Competitive Neural Network

2.1 Topology

The topology of the network is similar to the topology of the simple competitive neural network. The neural network is composed by *N* input sensors and *K* neurons, where *N* is the dimension of input patterns. All input sensors are connected with all neurons.

The synaptic weight of every neuron in the network is determined by a pair of vectors ($\mathbf{w}_{i1}$, $\mathbf{w}_{i2}$). The notation is the following:

$$\begin{aligned}\mathbf{w}_{i1} &= (w_{i1}^1, w_{i1}^2, w_{i1}^3, \dots, w_{i1}^N)\,,\ i = 1,2,\dots K\\ \mathbf{w}_{i2} &= (w_{i2}^1, w_{i2}^2, w_{i2}^3, \dots, w_{i2}^N)\,,\ i = 1,2,\dots K\end{aligned} \qquad (1)$$

When an input pattern, **x,** is presented to the network, the synaptic potential associated to the *i*-th neuron is

$$h_i = \alpha_i(\mathbf{x})h_{i1} + \overline{\alpha}_i(\mathbf{x})h_{i2} + \frac{1}{2}\alpha_i(\mathbf{x})\overline{\alpha}_i(\mathbf{x})\left\|\mathbf{w}_{i1} - \mathbf{w}_{i2}\right\|^2,\quad i = 1,2,\dots,K. \qquad (2)$$

where h_{i1} and h_{i2} are real numbers determined by the following expressions:

$$h_{i1} = \mathbf{w}_{i1}^T\mathbf{x} - \frac{1}{2}\mathbf{w}_{i1}^T\mathbf{w}_{i1} \text{ and } h_{i2} = \mathbf{w}_{i2}^T\mathbf{x} - \frac{1}{2}\mathbf{w}_{i2}^T\mathbf{w}_{i2},\quad i = 1,2,\dots,K \qquad (3)$$

and

$$\alpha_i(\mathbf{x}) = \frac{(\mathbf{w}_{i1} - \mathbf{w}_{i2})^T(\mathbf{x} - \mathbf{w}_{i2})}{\left\|\mathbf{w}_{i1} - \mathbf{w}_{i2}\right\|^2}\ ,\quad \alpha_i(\mathbf{x}) + \overline{\alpha}_i(\mathbf{x}) = 1 \qquad (4)$$

Now, the segment composed by the pair of reference vectors ($\mathbf{w}_{i1}$,$\mathbf{w}_{i2}$) can be defined as:

$$S_{\mathbf{w}_{i1}\mathbf{w}_{i2}} = \left\{\mathbf{s} = \alpha_i(\mathbf{x})\mathbf{w}_{i1} + (1-\alpha_i(\mathbf{x}))\mathbf{w}_{i2}\ , \alpha_i(\mathbf{x}) \in [0,1]\right\} \qquad (5)$$

This segment is called *structural segment* and can also be denoted by S_i.The distance, *D*, from an input pattern **x** to a structural segment is evaluated by the expression:

$$D(\mathbf{x}, S_i) = \left\|\mathbf{x} - S_{\mathbf{w}_{i1}\mathbf{w}_{i2}}\right\| = \begin{cases}\left\|\mathbf{x} - (\alpha_i(\mathbf{x})\mathbf{w}_{i1} + (1-\alpha_i(\mathbf{x}))\mathbf{w}_{i2})\right\| & \text{if } \alpha_i(\mathbf{x}) \in (0,1)\\ \left\|\mathbf{x} - \mathbf{w}_{i1}\right\| & \text{if } \alpha_i(\mathbf{x}) \geq 1\\ \left\|\mathbf{x} - \mathbf{w}_{i2}\right\| & \text{if } \alpha_i(\mathbf{x}) \leq 0\end{cases} \qquad (6)$$

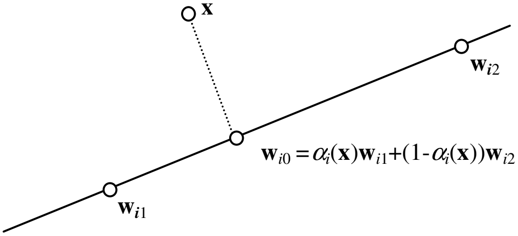


Fig. 1. Distance from an input pattern **x** to the segment $S_{\mathbf{w}_{i1}\mathbf{w}_{i2}}$

If $\alpha_i(\mathbf{x}) \in (0,1)$, then the point $\mathbf{w}_{i0} = \alpha_i(\mathbf{x})\mathbf{w}_{i1}+(1-\alpha_i(\mathbf{x}))\mathbf{w}_{i2}$ is the point of the segment $S_{\mathbf{w}_{i1}\mathbf{w}_{i2}}$ nearest to the input pattern **x** (see figure 1).

2.2 Computation Dynamics

When an input pattern **x** is presented to the network only one neuron is activated, the neuron whose structural segment is nearest to the input pattern **x**. This neuron *r* is called the *winning neuron* and satisfies the following expression:

$$\left\|\mathbf{x}-S_r\right\| = \min_{1\le i\le K}\left\|\mathbf{x}-S_i\right\| \tag{7}$$

In this sense, the neuron activated is the neuron with the highest synaptic potential, as is proved by the following theorem:

Theorem 1

$$h_r > h_i \Leftrightarrow \quad \left\|\mathbf{x}-S_r\right\| < \left\|\mathbf{x}-S_i\right\| \quad ,\forall\, r \ne i,\ i=1,2,...,K \tag{8}$$

Proof: using the structural segment definition we have:

$$\left\|\mathbf{x}-S_i\right\|^2 = \left\|\mathbf{x}-\left(\alpha_i(\mathbf{x})w_{i1}+\overline{\alpha}_i(\mathbf{x})w_{i2}\right)\right\|^2$$

so that:

$$\begin{aligned}
\left\|x-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\right\|^2 &= \left(\mathbf{x}-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\right)^T\left(\mathbf{x}-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\right)\\
&= \mathbf{x}^T\mathbf{x}-\mathbf{x}^T\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)^T-\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)^T\mathbf{x}\\
&\quad +\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)^T\left(\alpha_i(\mathbf{x})\mathbf{w}_{i1}+\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\\
&= \mathbf{x}^T\mathbf{x}-\alpha_i(\mathbf{x})\mathbf{x}^T\mathbf{w}_{i1}-\overline{\alpha}_i(\mathbf{x})\mathbf{x}^T\mathbf{w}_{i2}-\alpha_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{x}-\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}{}^T\mathbf{x}+\alpha_i^2(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}\\
&\quad +2\alpha_i(\mathbf{x})\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2}+\overline{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}\\
&= \mathbf{x}^T\mathbf{x}-2\alpha_i(\mathbf{x})\mathbf{x}^T\mathbf{w}_{i1}-2\overline{\alpha}_i(\mathbf{x})\mathbf{x}^T\mathbf{w}_{i2}+\alpha_i^2(\mathbf{x})\mathbf{w}_{i1}{}^T\mathbf{w}_{i1}+2\alpha_i(\mathbf{x})\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}{}^T\mathbf{w}_{i2}+\overline{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i1}{}^T\mathbf{w}_{i2}
\end{aligned}$$

From expression (3) we obtain that

$$\mathbf{w}_{i1}{}^{T}\mathbf{x} = h_{i1} + \frac{1}{2}\mathbf{w}_{i1}{}^{T}\mathbf{w}_{i1}\ ,\ \mathbf{w}_{i2}{}^{T}\mathbf{x} = h_{i2} + \frac{1}{2}\mathbf{w}_{i2}{}^{T}\mathbf{w}_{i2}$$

and now

$$\begin{aligned}
&\left\|x - \left(\alpha_i(\mathbf{x})\mathbf{w}_{i1} + \overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}\right)\right\|^2 = \\
&= \mathbf{x}^T\mathbf{x} - 2\alpha_i(\mathbf{x})\left(h_{ii} + \frac{1}{2}\mathbf{w}_{i1}^T\mathbf{w}_{i1}\right) - 2\overline{\alpha}_i(\mathbf{x})\left(h_{i2} + \frac{1}{2}\mathbf{w}_{i2}^T\mathbf{w}_{i2}\right) + \alpha_i^2(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1} \\
&\qquad + 2\alpha_i(\mathbf{x})\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2} + \overline{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2} \\
&= \mathbf{x}^T\mathbf{x} - 2\alpha_i(\mathbf{x})h_{i1} - \alpha_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1} - 2\overline{\alpha}_i(\mathbf{x})h_{i2} - \overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2} + \alpha_i^2(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1} \\
&\qquad + 2\alpha_i(\mathbf{x})\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2} + \overline{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2} \\
&= \mathbf{x}^T\mathbf{x} - 2\alpha_i(\mathbf{x})h_{i1} - 2\overline{\alpha}_i(\mathbf{x})h_{i2} - \alpha_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1} + \alpha_i^2(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1} - \overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2} \\
&\qquad + 2\alpha_i(\mathbf{x})\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2} + \overline{\alpha}_i^2(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2} \\
&= \mathbf{x}^T\mathbf{x} - 2\alpha_i(\mathbf{x})h_{i1} - 2\overline{\alpha}_i(\mathbf{x})h_{i2} - \alpha_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i1}\left(1 - \alpha_i(\mathbf{x})\right) - \overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i2}^T\mathbf{w}_{i2}\left(1 - \overline{\alpha}_i(\mathbf{x})\right) \\
&\qquad + 2\alpha_i(\mathbf{x})\overline{\alpha}_i(\mathbf{x})\mathbf{w}_{i1}^T\mathbf{w}_{i2}
\end{aligned}$$

Using the expression in (2) for the synaptic potential of the structural segment, the following expression is obtained:

$$\left\|\mathbf{x} - \left(\alpha_r(\mathbf{x})w_{r1} + \overline{\alpha}_r(\mathbf{x})w_{r2}\right)\right\|^2 = \mathbf{x}^T\mathbf{x} - 2h_i$$

Since in hypothesis it is supposed that $h_r > h_i$, then $\mathbf{x}^T\mathbf{x} - 2h_r < \mathbf{x}^T\mathbf{x} - 2h_i$. That is,

$$\left\|\mathbf{x} - \left(\alpha_r(\mathbf{x})w_{r1} + \overline{\alpha}_r(\mathbf{x})w_{r2}\right)\right\|^2 < \left\|\mathbf{x} - \left(\alpha_i(\mathbf{x})w_{i1} + \overline{\alpha}_i(\mathbf{x})w_{i2}\right)\right\|^2$$

$$\left\|\mathbf{x} - S_r\right\| < \left\|\mathbf{x} - S_i\right\|\ ,\ \forall i \neq r\ ,\ i = 1,2,\ldots,K \qquad \square$$

2.3 The Learning Rule

Once the winning neuron *r* is activated, the synaptic potential for it must change according to a learning rule. The problem is to determine the *K* segments S_1, S_2,…,S_K such that minimize the following representation quadratic error:

$$E = \sum_{\mu=1}^{p}\sum_{i=1}^{K} M_i^{\mu}(\mathbf{x})\left\|\mathbf{x}^{\mu} - S_i\right\|^2 \tag{9}$$

where (M_i^{μ}) is the cluster membership matrix which specifies whether or not an input pattern, $\mathbf{x}^{\mu}$, actives the neuron *i* as winner:

$$M_i^{\mu}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x}^{\mu} \text{ belongs to cluster } i \\ 0 & \text{otherwise} \end{cases}$$

Gradient descent on expression in (9) yields the following learning rule:

$$\begin{array}{l} \mathbf{w}_{r1}(k+1)=\begin{cases} \mathbf{w}_{r1}(k)+\eta(k)\cdot\left[\mathbf{x}(k)-\mathbf{w}_{r0}(k)\right]\cdot\alpha_r(\mathbf{x}(k)) & \text{if} \quad 0<\alpha_r(\mathbf{x}(k))<1 \\ \mathbf{w}_{r1}(k)+\eta(k)\cdot\left[\mathbf{x}(k)-\mathbf{w}_{r1}(k)\right]; & \text{if} \quad \alpha_r(\mathbf{x}(k))\geq 1 \\ \quad \mathbf{w}_{r2}(k+1)=\mathbf{w}_{r2}(k)+\beta(k)(w_{r1}(k+1)-\mathbf{w}_{r2}(k)) & \\ 0 & \text{if} \quad \alpha_r(\mathbf{x}(k))\leq 0 \end{cases} \\ \mathbf{w}_{r2}(k+1)=\begin{cases} \mathbf{w}_{r2}(k)+\eta(k)\cdot\left[\mathbf{x}(k)-\mathbf{w}_{r0}(k)\right]\cdot\overline{\alpha}_r(\mathbf{x}(k)) & \text{if} \quad 0<\alpha_{\mathrm{r}}(\mathbf{x}(k))<1 \\ 0 & \text{if} \quad \alpha_{\mathrm{r}}(\mathbf{x}(k))\geq 1 \\ \mathbf{w}_{r2}(k)+\eta(k)\cdot\left[\mathbf{x}(k)-\mathbf{w}_{r2}(k)\right]; & \text{if} \quad \alpha_{\mathrm{r}}(\mathbf{x}(k))\leq 0 \\ \quad \mathbf{w}_{r1}(k+1)=\mathbf{w}_{r1}(k)+\beta(k)(w_{r2}(k+1)-\mathbf{w}_{r1}(k)) & \end{cases} \end{array} \tag{10}$$

The learning parameters, $\eta(k)$ and $\beta(k)$, satisfy the following expressions:

$$\eta(k)=\eta_0\left(1-\frac{k}{T}\right), \quad \beta(k+1)=\beta(k)\left(1-\frac{k}{T}\right)^u, \text{ with } \beta(0)=\beta_0 \tag{11}$$

where k is the iteration in course, T is the total number of iterations of the learning process and u is a constant real number. The learning parameters take values in the interval [0,1], being η_0 and β_0 the initial values. The parameter $\eta(k)$ controls the moving of the structural segments in the input space whereas the parameter $\beta(k)$ controls the size of these segments (see section 3).

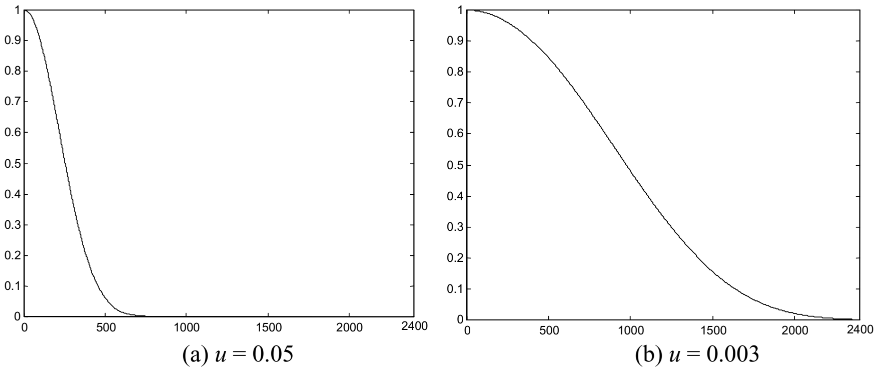
3 Shapes Description by the Segments-Based Neural Network

The constant u of the learning parameter $\beta(k)$ in expression (11) is very important. Depending on its value, the segments obtained by the network after the learning process are more or less elongated. In order to obtain the skeleton of a shape, the segments must be distributed along the entire shape. Visually, this set of segments approximate to the skeleton of the shape, except that they are not joined. In this sense, the extreme of the segments, having the lower euclidean distance between then, are joined by a line, obtaining now a continuous trace that determines the skeleton.

The shapes used for experiments are composed of 600 patterns randomly distributed and, since the input patterns are randomly introduced four times during the learning process, T=2400.

If the learning parameter $\beta(k)$ could reach the value zero long before the total number of iterations T, the segments could expand without any control and they were not distributed along the entire shape. However, if the learning parameter $\beta(k)$ reaches the value zero just in the final iterations, the segments are expanded with control, distributing now along the entire shape.

By way of example, figures 3 and 4 illustrate the result of the neural network with 6 neurons (segments) from the same shape with two different values for the constant u of the learning parameter $\beta(k)$. Figure 2(a), shows the evolution of the learning parameter $\beta(k)$ with u=0.05 and figure 2(b) with u=0.003.



(a) $u = 0.05$ (b) $u = 0.003$

Fig. 2. Evolution of $\beta(k)$ parameter with different values of u

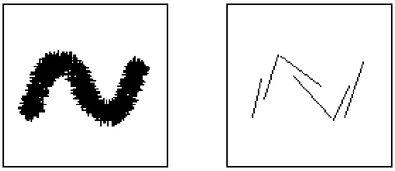


Fig. 3. Result of the neural network with $u = 0.05$

The result of the neural network with u=0.05 is shown in figure 3 whereas the result with u=0.003 is shown in figure 4(b).

In order to obtain the skeleton of the shape, it is only necessary to join the extremes of the segments, obtained with the neural network, having the lower euclidean distance. To avoid undesirable connections, the joins are not performed if the euclidean distance is greater than a unit. The skeleton obtained is shown in figure 4(c).

It's important to emphasize that all shapes presented to the network are always novel shapes, that is, the neural network is not previously trained with any set of shapes. The topology of every shape is obtained during the learning phase, in where segments are distributed along the entire shape.

Since all shapes used in the experiments have the same number of patterns, the function for the learning parameter $\beta(k)$, shown in figure 2(b), is used for all of them. The Shapes used in the experiments are shown in figures 5(a), 6(a), and 7(a). The number of neurons (segments) used for them are 7, 12, and 13 respectively. The final configurations of the neural network for these shapes are shown in figures 5(b), 6(b), and 7(b). The skeletons for these shapes are achieved by joining the extremes of the segments, obtained with the neural network, which have the lower euclidean distance. These skeletons are shown in figures 5(c), 6(c) and 7(c) respectively.

In any experiment, the number of segment needed must be sufficient for representing the geometrical and topological features of the shape.

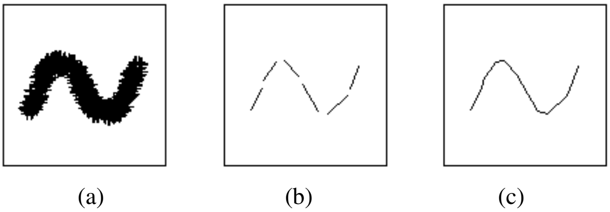

(a) (b) (c)

Fig. 4. Skeleton obtained by the neural network with 6 segments.

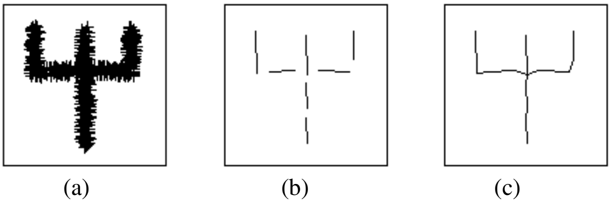

(a) (b) (c)

Fig. 5. Skeleton obtained by the neural network with 7 segments

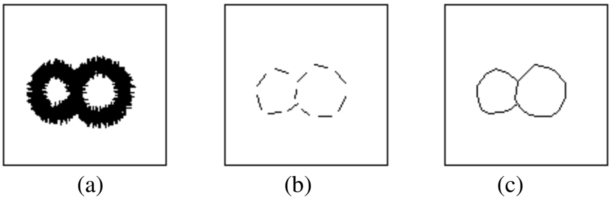

(a) (b) (c)

Fig. 6. Skeleton obtained by the neural network with 12 segments

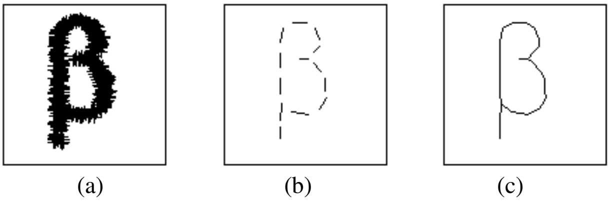

(a) (b) (c)

Fig. 7. Skeleton obtained by the neural network with 13 segments

4 Conclusions

In this paper, an unsupervised competitive neural network based on segments has been proposed. In a simple competitive neural network, the synaptic weight vectors represent the prototypes of the classes in which the input data are organized. In the proposed model, the synaptic weights are composed of two reference vectors, which determine the extremes of a segment. That is, every unit of the network has associated a segment that will evolve in the learning process. Since the prototype of every unit is a segment, instead of a single vector, additional information can be obtained, for example the predominant direction of the classes.

The model has two different learning parameters, $\eta(k)$ and $\beta(k)$. The parameter $\eta(k)$ controls the moving of the structural segments in the input space whereas the parameter $\beta(k)$ controls the size of these segments in the learning process.

The importance of the constant u of the learning parameter $\beta(k)$ has been noted. Depending on its value, the segments obtained by the network after the learning process are more or less elongated. With an appropriate selection for the constant *u,* the segments are expanded with control, and distributed along the entire shape.

After the learning process, the extremes of the segments, having the lower euclidean distance between then, are joined by a line, resulting in a continuous trace that determines the skeleton of the shapes. In order to avoid undesirable connections, the joins are not performed if the euclidean distance is greater than a unit.

We have shown that this model is suitable for obtaining the skeleton of elongated shapes without any previous classification or distinction with an unsupervised method.

Acknowledgements

This work has been partially supported by TIN-7362 project (MEC).

References

1. Kohonen, T.: Self-organizing Maps. Springer, Berlin (1997)
2. Corridoni, J.M., Bimbo, A., Landi, L.: 3D Object classification using multi-object Kohonen networks. Pattern Recognition 29, 919–935 (1996)
3. Lopez-Rubio, E., Muñoz-Perez, J., Gomez-Ruiz, J.A.: Invariant pattern identification by self-organizing networks. Pattern Recognition Letters 22, 983–990 (2001)
4. Subba Reddy, N.V., Nagabhushan, P.: A three-dimensional neural network model for unconstrained handwritten numeral recognition: a new approach. Pattern Recognition 31, 511–516 (1998)
5. Wang, S.S., Lin, W.G.: A new self-organizing neural model for invariant pattern recognition. Pattern Recognition 29, 677–687 (1996)
6. Blum, H.: A transformation for extracting new descriptions of shape. In: IEEE Proceedings of the Symposium on Models for the Speech and Vision Form, Boston, USA, pp. 362–380 (1964)
7. Datta, A., Parui, S.K., Chaudhuri, B.B.: Skeletonization by a topology-adaptive self-organizing neural network. Pattern Recognition 34, 617–629 (2001)
8. Datta, A., Pal, T., Parui, S.K.: A modified self-organizing neural net for shape extraction. Neurocomputing 14, 3–14 (1997)
9. Amin, A., Al-sadoun, H., Fischer, S.: Hand-printed arabic character recognition system using an artificial network. Pattern Recognition 29(4), 663–675 (1996)
10. Cardoner, R., Thomas, F.: Residuals + directional gaps = skeletons. Pattern Recognition Letters 18, 343–353 (1997)

Protecting DCT Templates for a Face Verification System by Means of Pseudo-random Permutations

Marco Grassi[1] and Marcos Faundez-Zanuy[2]

[1] Department of Biomedical, Electronic and Telecommunication Engineering
Università Politecnica delle Marche, Ancona, Italy
`margra75@hotmail.com`
[2] Escola Universitària Politècnica de Mataró (Adscrita a la UPC)
08303 MATARO (BARCELONA), Spain
`faundez@eupmt.es`
`http://www.eupmt.es/veu`

Abstract. Biometric template security and privacy are a great concern of biometric systems, because unlike passwords and tokens, compromised biometric templates cannot be revoked and reissued. In this paper we present a protection scheme for an identity verification system through biometrical face recognition based on a user dependent pseudo-random ordering of the DCT template coefficients and MPL and RBF Neural Networks for classification. In addition to privacy enhancement, because a hacker can hardly match a fake biometric sample without knowing the pseudo-random ordering this scheme, it also increases the biometric recognition performance.

1 Introduction

Biometric template security is an important issue because unlike passwords and tokens, compromised biometric templates cannot be revoked and reissued. Thus, there is a strong interest on the possibility to cancel and replace a given biometric data when compromised. If a biometric is lost once (illegally acquired by a hacker), it is compromised forever, hence this information may need to be protected for a very long time [1].

An ideal biometric template protection scheme should possess the following properties [2]:

1) Diversity: the secure template must not allow cross-matching across databases, thereby ensuring the user's privacy.
2) Revocability: it should be straightforward to revoke a compromised template and reissue a new one based on the same biometric data.
3) Security: it must be computationally hard to obtain the original biometric template from the secure template. This property prevents a hacker from creating a physical spoof of the biometric trait from a stolen template.
4) Performance: the biometric template protection scheme should not degrade the recognition performance (identification or verification rates).

For the next years, we are evolving towards ambient intelligence, pervasive networking or ubiquitous computing, which have special characteristics [1]. In addition

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1216–1223, 2009.

© Springer-Verlag Berlin Heidelberg 2009

there is a gradual erosion of the computational difficulty of the mathematical problems on which cryptology is based, due to developments in computation (progress in electronics and in the future in optical and maybe even quantum computing). This increases the vulnerability of biometric systems [3].

Encryption is not a smooth function [2] and a small difference in the values of the feature sets extracted from the raw biometric data would lead to a very large difference in the resulting encrypted features. While it is possible to decrypt the template and perform matching between the query and decrypted template, such an approach is not secure because it leaves the template exposed during every authentication attempt. Thus, standard encryption techniques are not useful for securing biometric templates. The solutions proposed in the literature can be split into two categories [3]:

1) Feature transformation.
2) Biometric Cryptosystems.

We will describe these categories in the next sections.

1.1 Feature Transformation

A transformation function $Y = f(x)$ is applied to the biometric information and only the transformed template is stored in the database. The parameters of the transformation function are typically derived from a random key or password. The same transformation function is applied to the test signal and the transformed query is directly matched against the transformed template.

Feature transformation can be divided into salting and non-invertible transforms. In salting $Y = f(x)$ is invertible. Thus, if a hacker knows the key and the transformed template, he can recover the original biometric template, and the security is based on the secrecy of the key or password. This is the unique approach that requires a secret information (key). This is not necessary in the other categories. The second group is based on noninvertible transformation systems. They apply a one-way function on the template and it is computationally hard to invert a transformed template even if the key (transform function) is known. The main drawback of this approach is the trade-off between discriminability and noninvertibility of the transformation function. Transformed features belonging to the same user should have high similarity after transformation, while features from different users should be quite different.

While our intuition seems to suggest that it is very easy to design a function that is “easy” to compute but “hard” to invert, so far the best theoretical results can prove that there exists functions that are twice as hard to invert as to compute [4]. It is clear that such functions would be completely useless to practical cryptology [1].

1.2 Biometric Cryptosystems

In this approach some public information about the biometric template is stored. This public information is called helper data. For this reason they are also known as helper data-based methods. While the helper data does not reveal any significant information about the original biometric template, it is needed during the matching to extract a cryptographic key from the input signal. Matching is performed indirectly by verifying the correctness of the extracted key. In order to cope with intra-user variations, error correction coding techniques are usual.

Biometric cryptosystems can also be split into two groups: key binding and key generation systems depending on how the helper data is obtained. When the helper data is obtained by binding a key (that is independent of the biometric features) with the biometric template, it is known as key-binding biometric cryptosystem. In the second case, the helper data is derived only from the biometric template and the cryptographic key is directly generated from the helper data and the query biometric features. This second approach is known as key generation biometric cryptosystem [5]. Authentication in a key-binding biometric cryptosystem is similar except that the helper data is a function of both the template and the key *K*.

Hybrid techniques combining the previous four approaches are also possible.

2 Pseudo-random Permutations

In this paper, in order to secure the templates, we use a different random permutation of template coefficients for each person. Thus, our proposal corresponds to salting feature transformation described in section 1.1. Figure 4 shows the diagram for the proposed approach. The template coefficients are equal to the DCT components of the two dimensional transform of the face image [6]. The permutation order is different for each person (although more than one permutation per person is possible) and it is given by a Key, which must be kept secret. The advantages of this strategy are the following:

- There is an increase on privacy, because it is impossible (computationally very hard) to obtain the face image without knowing the permutation order. Figure 5 shows the number of different permutations that can be done for a given feature vector length. If a feature vector template has *N* coefficients, the number of permutations of these coefficients is equal to *N*!
- There is an improvement on the recognition rates because an impostor does not know the correct permutation order. This is similar to the privacy achieved by CDMA (Code Division Multiple Access) used in some mobile telephone standards in order to secure the communications. If you do not know the correct order (provided by the pseudo-random frequency hopper order) you cannot decode the message. Obviously if the impostor knows the permutation order then he can sort his/her feature vector and then the protection is equal to the biometric system without permutation. Anyway, it is as difficult to get the Key as in other security systems (VISA number, password, PIN, etc.).
- In contrast to some encryption systems, where the template must be decrypted before comparison, in our approach there is no need to re-order the coefficients. They can be directly compared.

Although this is a very simple approach, it has a main advantage: experimental results are always better or equal than the baseline biometric system without encryption. This is not the case with a large majority of existing systems in the literature, where intra-user variability is hard to manage and provides an increase in False Rejection Rates.

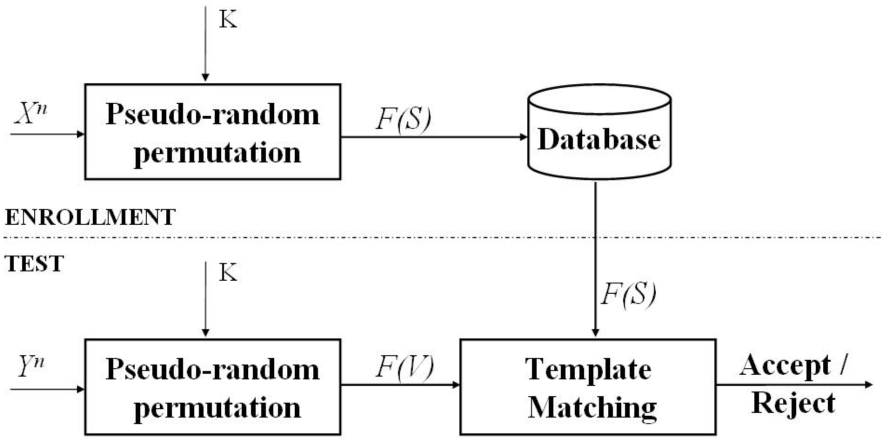


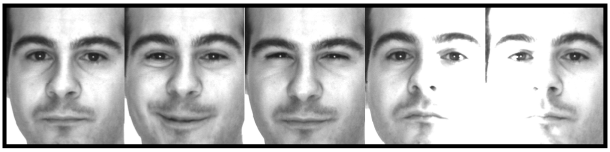
Fig. 4: Scheme for proposed system. In the enrollment phase, the biometric template X^n coefficients are permuted according to the secret key *K*. This permuted template is stored in the database. In the test phase a noisy version of the biometric template Y^n is acquired, and the user provides his/her Key. The permuted template $F(V)$ is matched with the template of the claimed identity. If $distance(F(V),F(S)) < threshold$, the user is accepted. Otherwise he/she is rejected.

3 Experimental Results for Face Recognition System

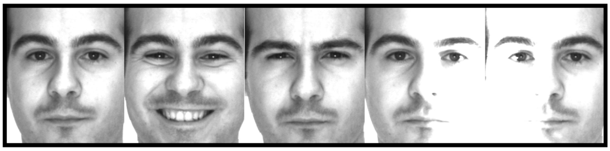
The proposed protection schema has been applied to an identity verification system based on biometric face recognition [7]. In the system, the DCT has been used for features extraction and neural networks (both MLP and RFB) have been used for classification [8].

The AR database [9], used for the experiments, is a publicly available database (http://cobweb.ecn.purdue.edu/RVL/ARdatabase/ARdatabase.html) of 126 individuals, with 26 images of each, taken in two different sessions at a time distance of two weeks, varying the lighting and the facial expression. We have used 10 of the 26 images (5 images as enrollment templates and 5 different ones for testing the system), excluding the ones overexposed and the ones in which the face was partially occluded by sunglasses or scarves, of 117 of the 126 individuals, those of the rest being either not complete or not available. All the images have been cropped and normalized to 64x74 grey ones.

We have used as verification error the minimum Detection Cost Function, to deal with a trade-off between two different kinds of possible errors in verification: missed detection (those situations where a user is incorrectly rejected) and false alarms (those situations where an impostor is accepted), which has usually to be established by adjusting a decision threshold. The DCF is defined by [10]:



1st images set of AR database



2st images set of AR database

Fig. 2. AR database samples of one person

$$DCF = \left(C_{Miss} \times P_{Miss} \times P_{T\arg et}\right) + \left(C_{FalseAlarm} \times P_{FalseAlarm} \times \left(1 - P_{T\arg et}\right)\right) \tag{1}$$

where C_{Miss} and $C_{FalseAlarm}$ represent respectively the cost of a missed detection and of a false alarm, P_{Miss} and $P_{FalseAlarm}$ represent respectively the Miss and the False Alarm probability and $P_{T\arg et}$ represents the a priori probability of the Target to be known. An error type weighting of 1:1 and an equal probability that the target be known or unknown ($C_{Miss} = C_{FalseAlarm} = 1$; $P_{T\arg et} = 0.5$) have been chosen.

The DCT (Discrete Cosine Transform) has been used for characteristics extraction, we obtain one model vector from each training image. A vector dimension of N'xN'=10x10=100 coefficients has been chosen in order to grant at the same time a fast computation (the number of coefficients corresponds to the number of the input neurons for the neural networks used as classifiers), good performances in recognition and in security through an elevate number of possible permutations (100! means more than 10^{157} permutation). Experiments have been done to compare the performances with or without a pseudo-random permutation of the DCT's coefficients, using MLP and RBF neural networks, as in the following.

3.1 Single Multi Layer Perceptron (MLP)

A three layer perceptron with a variable number of hidden neurons has been used in the simulations, with a scaled conjugate gradient back-propagation algorithm, that is based on conjugated directions, setting up the following parameters:

- Number of epochs: 15000
- Input neurons: 100
- Hidden layer neurons: 10 : 10 : 150
- Output layer neurons: 117 neurons (one for each person)
- Performance function: regularized mean square error (MSEREG) [11]

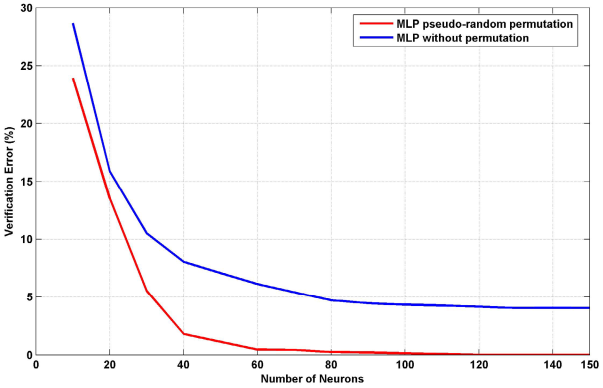


Fig. 3. Verification rates using a MLP-NN, as function of the number of neurons, with 100 DCT coefficients, with and without pseudo-random permutation

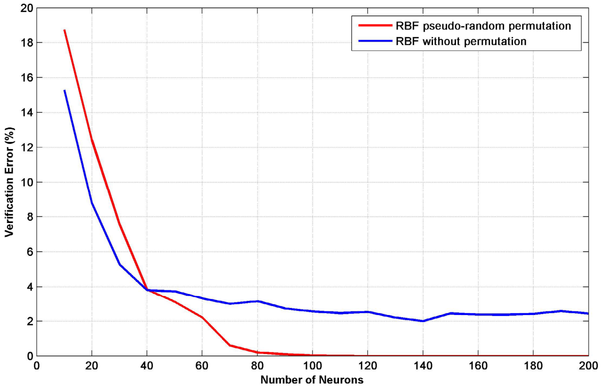


Fig. 4. Verification rates using a RBF-NN, as function of the number of neurons, with 100 DCT coefficients, with and without pseudo-random permutation

The results, in Figure 3, show how the application of the pseudo-random permutation leads to a significant improvement of system performances, with verification error that goes to 0 using more than 100 neurons.

3.2 Radial Basis Function Neural Network

Radial Basis Function Neural Network can require more neurons than a standard feed-forward back-propagation networks, but they can be trained in a fraction of time needed by standard feed-forward networks.

In the simulations a RBF-NN has been used, with Gaussian function, applying the same training methodology of the MLP neural network, setting up the following parameters:

- RBF neurons: 10 : 10 : 200
- Output layer neurons: 117
- Spread: 1.5 : 0.25 : 4

Also in the case of a RBF-NN, Figure 4, the application of permutated template lead to a significant performance improvement, using a number of neurons greater than 40, completely avoiding verification errors for more than 110 neurons.

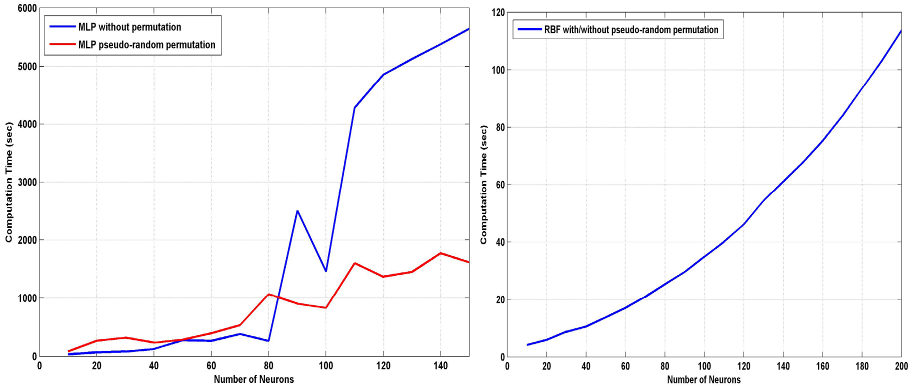


Fig. 5. Training times for the MLP and the RBF (medium time for each spread), as function of the number of neurons, with 100 DCT coefficients, with and without pseudo-random permutation

Interesting results appear also from a training time comparison, with and without the application of the pseudo-random permutation of the DCT coefficients [Figure 5]. For the RBF-NN the permutation of the coefficients doesn't lead time training differences. In the case of the MPL-NN, while for a low number of neurons the use of the coefficients without permutation requires a slightly minor training time, for a number of neurons superior than 80 the permutation of the coefficients leads to a great improvement in terms of computational time.

4 Conclusions

In this paper we have presented an authentication system through face identity verification that uses a simple approach for biometric template protection that belongs to the salting feature transformation. The unique drawback of this approach is that it requires a secret key. If it is compromised, the biometric template can be exactly recovered. Nevertheless, the chance to crack the secret key is at least as difficult as to obtain the VISA number, PIN, or password of a classical security system, taking in account that if the key is kept secret the number of different permutations equals *100*!, which is a very large value for typical face template sizes.

In addition to security enhancement, there is always an improvement on the biometric verification errors which are dramatically reduced, both for an MPL and RBF neural network, and on the training time for the MPL neural networks. This is not the case of a large majority of the proposed methods in the literature, where privacy implies a degradation of biometric (False Rejection Rate is increased) and computational performances.

References

1. Preneel, B.: ECRYPT: The Cryptographic Research Challenges for the Next Decade. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 1–15. Springer, Heidelberg (2005)
2. Jain, A.K., Nandakumar, K., Nagar, A.: Biometric template security. Eurasip journal on Advances in Signal Processing. Special issue on Biometrics, 1–20 (January 2008)
3. Faundez Zanuy, M.: Biometric security technology. IEEE Aerospace and Electronic Systems Magazine 21(6), 15–26 (2006)
4. Hiltgen, A.P.L.: Constructions of feebly-one-way families of permutations. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, pp. 422–434. Springer, Heidelberg (1993)
5. Tuyls, P., Goseling, J.: Capacity and examples of template-protecting biometric authentication systems. In: Maltoni, D., Jain, A.K. (eds.) BioAW 2004. LNCS, vol. 3087, pp. 158–170. Springer, Heidelberg (2004)
6. Ratha, N.K., Chikkerur, S., Connell, J.H., Bolle, R.M.: Generating cancellable fingerprint templates. IEEE Trans. On Pattern Analysis and Machine Intelligence 29(4), 561–572 (2007)
7. Abate, A.F., Nappi, M., Riccio, D., Sabatino, G.: 2D and 3D Face Recognition: A Survey. Pattern Recognition Letters (December 4, 2006)
8. Grassi, M., Faundez-Zanuy, M.: Face Recognition with Facial Mask Application and Neural Networks. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 709–716. Springer, Heidelberg (2007)
9. Martinez, A.M.: Recognizing Imprecisely Localized, Partially Occluded, and Expression Variant Faces from a Single Sample per Class. IEEE Transaction On Pattern Analysis and Machine Intelligence 24(6), 748–763 (2002)
10. Martin, A.: The DET curve in assessment of detection task performance. In: European Speech Processing Conference Eurospeech 1997, vol. 4, pp. 1895–1898 (1997)
11. Zanuy, M.F.: Face Recognition in a Transformed Domain. In: Proceedings 37th Annual International Carnahan Conference On Security Tecnology, pp. 290–297 (2003)

Efficient Parallel Feature Selection for Steganography Problems

Alberto Guillén[1], Antti Sorjamaa[3], Yoan Miche[3],
Amaury Lendasse[3], and Ignacio Rojas[2]

[1] Department of Informatics - University of Jaen, Spain
[2] Department of Computer Architecture and Technology
University of Granada, Spain
[3] Department of Information and Computer Science
Helsinki University of Technology, Finland

Abstract. The steganography problem consists of the identification of images hiding a secret message, which cannot be seen by visual inspection. This problem is nowadays becoming more and more important since the World Wide Web contains a large amount of images, which may be carrying a secret message. Therefore, the task is to design a classifier, which is able to separate the genuine images from the non-genuine ones. However, the main obstacle is that there is a large number of variables extracted from each image and the high dimensionality makes the feature selection mandatory in order to design an accurate classifier. This paper presents a new efficient parallel feature selection algorithm based on the Forward-Backward Selection algorithm. The results will show how the parallel implementation allows to obtain better subsets of features that allow the classifiers to be more accurate.

1 Introduction

Steganography has been used and known for a very long time and it aims at hiding some content (usually called *message*) into an apparently innocuous document – mostly digital files nowadays.

Image steganography is currently one of the most investigated field of steganography, since there are many images available online and hence, potentially embedding a secret message.

Steganalysis is the "opposite" process: the main goal is to detect, with the highest possible accuracy, the presence of a secretly embedded content in another document. This can be seen as a typical classification problem, since an optimal separation between images embedding a content (stego images) and genuine ones (cover images) is to be searched.

There exists different ways to perform steganalysis, but the most classical one remains to extract features from each considered image (considered suspicious). Fridrich *et al.* feature set for image steganalysis is widely used [1] since it enables achieving a high detection rate of stego images. Even though Fridrich's set of features is very large (274 features), there are other steganalysis feature sets with even greater number of features.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1224–1231, 2009.
© Springer-Verlag Berlin Heidelberg 2009

This paper presents an application of a new parallel feature selection scheme for the steganalysis problem. The methodology is a parallelized version of the Forward-Backward Selection method, which has been recently successfully applied in Time Series prediction problems [2].

The rest of the paper is organized as follows: Section 2 presents the original Forward-Backward algorithm. Then, Section 3 introduces the new improvements incorporated to the algorithm. Afterwards, Section 4 describes the dataset used and the experiments performed. Finally, in Section 5, conclusions are drawn.

2 Forward-Backward Selection

Forward-Backward (FB) Selection is an algorithm that results from the joining of two methodologies: Forward and Backward selections [2]. Both the Forward Selection and the Backward Elimination (or Pruning) methods suffer from an incomplete search. FB offers the flexibility to reconsider input variables previously discarded and *vice versa*, to discard input variables previously selected. It can start from any initial input set, including empty, full or randomly initialized input set.

Let us suppose a set of inputs $\mathbf{X}^i$, $i = 1, 2, \cdots, d$ and an output $\mathbf{Y}$, the procedure of the Forward-Backward Selection is summarized in Fig. 1. In the procedure and in this paper, the k-Nearest Neighbors (kNN) criteria is used as an example criteria for evaluating the input set, but the criteria can be almost any criteria or a suitable approximator or classification algorithm.

1. (Initialization)
   Let S be the selected input set, which can contain any input variables, and F be the unselected input set containing the inputs, which are not in set S. Compute k-NN(S, Y).
2. (Forward-backward selection)
   Find:

$$\mathbf{X}^s = \arg\max_{\mathbf{X}^i, \mathbf{X}^j} \{k\text{-NN}(\{S, \mathbf{X}^j\}, \mathbf{Y})\} \cup \{k\text{-NN}(S \backslash \mathbf{X}^i, \mathbf{Y})\}, \mathbf{X}^i \in S, \mathbf{X}^j \in F.$$

   If the previously computed k-NN is larger than the new k-NN, stop; otherwise, update set S and save the new k-NN, repeat step 2 until S is equal to any former selected S.
3. (Result)
   The selection result is in set S.

Fig. 1. Forward-Backward Selection Strategy

It is noted that the selection result depends on the initialization of the input set. In this paper, several options are considered and the options are discussed more deeply in Section 3.3.

The number of input sets to be evaluated varies and is dependent on the initialization of the input set, the stopping criteria and the nature of the problem. Still, it is not guaranteed that in all cases this selection method finds the global optimal input set.

2.1 *k*-Nearest Neighbors

The k-Nearest Neighbors (k-NN) approximation method is a very simple but powerful method. It has been used in many different applications, particularly for classification tasks [3]. The key idea behind the k-NN is that samples with similar inputs have similar output values. Nearest neighbors are selected, according to Euclidean distance, and their corresponding output values are used to obtain the approximation of the desired output.

In this paper, since the problem is a binary classification task, the estimated output value of a sample is the class of the majority of the nearest neighbors k. The k has to be determined beforehand.

3 Parallel Forward-Backward Selection

The following sections describe the modifications that have been done to the original Forward-Backward in order to improve the performance in terms of computational cost and classification accuracy. The parallel implementations were done in MATLAB using the MPI standard and the interface used in [4].

3.1 Improvement 1: Efficient Computation of the Distance Matrix

As described in the previous Section, the k-NN has to be computed each time a subset of variables has to be evaluated. This requires the computation of the distances between an input vector and the others to determine the k closest ones. This results in a heavy computational load and in order to save some time, a distance matrix will be computed in advance. These calculations become crucial when dealing with the feature selection using scaling, that is, weighting the importance of a variable instead of merely selecting it. The distance matrix will have an element for each couple of values, storing the distances separately between each input and each sample.

Nonetheless, because the computation of this matrix is quite time consuming, an another improvement has been done: parallel calculation of the distance matrix. The matrix is divided between a set of processes and each one of them will perform locally a part of the calculations. Once they are done, they will communicate with the other processes in a collective way so that all processes have the complete matrix. Each process computes its starting and ending columns as $(rank) * ceil(d/size_w) + 1$ and $(rank + 1) * ceil(d/size_w)$, except for the last process that ends in d, where d is the number of all possible input variables, $size_w$ is the number of all processes and $rank$ is the process identifier from 0 to $size_w - 1$.

3.2 Improvement 2: Parallel FB

The search for the best solution is also distributed to several computers so that more solutions can be evaluated in less time. The parallel implementation is quite straightforward and consists of the division of the newly generated subset of variables in the first iteration of the FB. This division is shown in Fig. 2. Once each process has a part of the subset, they proceed as the original FB. The algorithm stops when all the processes have converged to a solution. Then, the best solution is found among the final solutions of the individual processes.

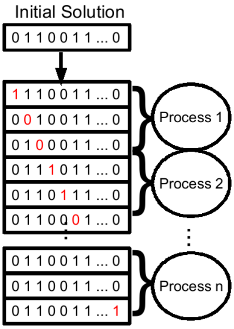


Fig. 2. Parallel scheme for the Forward-Backward Selection

The advantage of this approach is that each process has a different starting solution. Thus, it is possible to find different local minima and have a more thorough exploration of the solution space than in the original FB.

3.3 Improvement 3: FB Initialization

One of the main issues, when using the FB, is how to choose the initial solution from where the search will begin. Due to the locality of the FB, the risk of falling into a local minimum is quite high. Therefore, the starting point should be close enough to a good local minimum in order to get proper results. In this paper, two approaches are used: Mutual Information and Slice Division. A comparison of their performances will be shown in the Experiments section.

Mutual Information Based Initialization. The Mutual Information (MI) is used as a heuristic to find an adequate starting point for the FB selection. Let $\mathbf{X}^l = \{\boldsymbol{x}_m^l\}$ with $l \in 1, ..., d$ (i.e. $\mathbf{X}^l$ is the l-th input variable) and $\mathbf{Y} = \{y_m\}$ with $\{m = 1...M\}$. The Mutual Information between $\mathbf{X}^l$ and $\mathbf{Y}$ can be defined as the amount of information that $\mathbf{X}^l$ provides about $\mathbf{Y}$, and can be expressed as:

$$I(\mathbf{X}^l, \mathbf{Y}) = \sum_{y \in \mathbf{Y}} \sum_{x \in \mathbf{X}} \mu_{\mathbf{X}^l, \mathbf{Y}}(x, y) \log \frac{\mu_{\mathbf{X}^l, \mathbf{Y}}(x, y)}{\mu_{\mathbf{X}^l}(x) \mu_{\mathbf{Y}}(y)}. \quad (1)$$

$\mu_{\mathbf{X}^l,\mathbf{Y}}$ is the joint probability distribution function of $\mathbf{X}$ and $\mathbf{Y}$, and $\mu^l_{\mathbf{X}}(x)$ and $\mu_{\mathbf{Y}}(y)$ are the marginal probability distribution functions of $\mathbf{X}$ and $\mathbf{Y}$ respectively.

Therefore, in order to obtain the MI between $\mathbf{X}^l$ and $\mathbf{Y}$, only the estimate of the joint probability density function is needed. This value can be computed using several techniques based on histograms, kernels or the k-NN [5].

For each input variable, the MI between that variable and the output is computed and, once finished, it is possible to rank all the input variables according the values of MI. Then, the initial solution for the FB is defined as a number of *first* variables in the ranking. The problem now is to determine the actual number of variables, since the value obtained by the MI is not enough to perform this selection. Unfortunately, the only chance is to set this value manually, as it will be shown in the experiments section.

Slice Division. Being aware of the two significant drawbacks of the used MI heuristic (the determination of the k and the final number of variables to be chosen), another heuristic is considered, which requires the definition of only one parameter. This value can again be set manually as in the MI, or as a function of the available resources making the heuristic more flexible when executed in different computer architectures or systems.

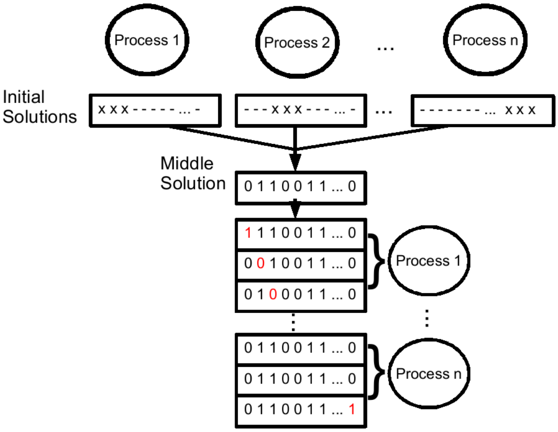


Fig. 3. Scheme of the algorithm using the Slice Division heuristic to initialize the starting point for the FB

The heuristic works as follows: it performs a preinitialization by dividing the original input vector into subvectors of a smaller dimension. Then, the local search is applied to each subvector although, the evaluation of a subvector implies the evaluation of the complete set of variables. Therefore, four starting alternatives arise: 1) all zeros, 2) subvectors ones, the rest zeros, 3) all ones, 4) subvectors zeros, the rest ones. Although the first one seems the most reasonable, since it is easier to find a local minimum with a reduced set of variables, the different alternatives were studied in the experiments. The issue remains, how large the subvectors should be? This value can be set manually or as a function of the number of available processors.

In the scheme shown in Fig. 3, the initial solutions are divided into slices of size 3 (depicted as x) and the remaining values are not changed (depicted as -) during the first FB. This first FB is done sequentially and separately in each process. Once all the processes have converged to a local optima, the processes perform a collective communication and share the results found by the other processes. Then, the initial starting point for the parallel FB (named *middlesolution* in Fig. 3) is computed by concatenating all local solutions.

4 Experiments

Two experiments are shown in order to illustrate the benefits of the newly added features: analysis of the computational cost and number of evaluations, and the accuracy of the classifications.

The dataset has been taken from a real-life steganalysis problem, where a set of features are extracted from an image. The dataset contains 10000 images from the BOWS2 challenge [6] database, hosted by Andreas Westfeld and it is publicly available in [7]. All images are 512 × 512 greyscale (color versions are also available). The steganographic algorithms have been used only on these specific dimensions of the images from the database, but any size would work just as well.

For half of the whole base of images, chosen randomly, steganography is applied. For this application, one well-known steganographic algorithm, called OutGuess [8], has been used. The OutGuess algorithm is known to be rather weak in steganalysis. An embedding rate of 15% of the maximum capacity of the OutGuess has been used.

Once half of the images are stego, the feature extraction is performed [1], leading to 274 features for each image. This feature extraction process is also carried out on the non-stego images. The final dataset is obtained by joining the two halves, giving a total of 10000 samples in 274-dimensional space.

The dataset is further divided into training and test sets, 1000 samples and roughly 9000 samples, respectively. Both sets are normalized using the mean and standard deviation of the training set.

4.1 Experiment 1: Parallel Performance Evaluation

The original FB algorithm and the improved versions were executed in identical machines in order to compare the computational times. The results are shown in Table 1.

In Table 1, the Variable Selection time measured depends greatly on when the search converges to a local optimum. Regarding the time consumed for the calculation of the distances, the proposed parallel implementation turns out to be efficient and scalable. The scalability remains good also in terms of the number of solutions evaluated, which is growing linearly with the number of processes.

Table 1. Times (in minutes) spent on the distance matrix calculation and in the variable selection and the number of solutions of the sequential and the parallel implementations using the MI and the FBS methodologies

	Distance	Variable Selection	# Evaluated Solutions
Sequential	15	238	1650
Parallel 2p	11	297	3848
Parallel 4p	5	284	6596
Parallel 6p	4	273	10169
Parallel 8p	3	348	13742

4.2 Experiment 2: Classification Accuracy

The second experiment concentrates to the results obtained by the classifier after the feature selection. Table 2 shows the errors obtained for both training and test subsets for the algorithm using different heuristics to set the initial starting point.

Table 2. Results of the variable selection. For each variable selection scheme the following results are shown: a number of initial and finally selected variables, respective k-NN classification error and training and test errors of the OP-KNN classification methodology.

	Variables		k-NN	OP-KNN	
	Initial	Selected	Train	Train	Test
All Variables	274	274	0.190	0.172	0.210
FBS	30	33	0.118	0.120	0.146
	50	52	0.122	0.110	0.141
	70	73	0.131	0.134	0.145
	100	102	0.134	0.127	0.155
pFBS	30	35	0.113	0.115	0.141
	50	52	0.115	0.125	0.135
	70	73	0.126	0.130	0.145
	100	103	0.118	0.120	0.141
pFBSv2	All Ones	245	0.157	0.142	0.192
	All Zeros	56	0.114	0.110	0.138

The classification errors using the k-NN algorithm and the OP-KNN [9] show, how important it is to perform feature selection beforehand: all the models that were trained after the dimensionality reduction outperform the ones trained with all the variables. The results also show, how the two initialization heuristics have a good behavior, although it is remarkable that the last one, based on a local slice optimization, has the best performance. Furthermore, this initialization, starting from the empty subset of variables, was able to select 50 variables, avoiding many local minima.

5 Conclusions

The problem of identifying if an image has a hidden message is really interesting, because of its possible applications, specially in the World Wide Web. The work presented in this paper proposes a methodology for the dimensionality reduction in order to make the data samples easier to learn by the artificial models.

Regarding the feature selection problem, the Forward-Backward algorithm was used as a first choice, but it provided poor results requiring a significant computational cost. Therefore, three major improvements were presented in this paper: parallel implementation of the computation of the distance matrix, parallelization of the search methodology and two new heuristics to determine the starting point of the search. This last element is crucial due to the large amount of local minima in the solution space.

The results have shown how the new elements allow the algorithm to perform an adequate feature selection, allowing the models to provide accurate classifications.

Acknowledgement. This work has been partially supported by the projects TIN2007-60587, P07-TIC-02768 and P07-TIC-02906, TIC-3928. Part of the work of Antti Sorjamaa has been supported by a grant from Nokia Foundation, Finland.

References

1. Pevny, T., Fridrich, J.: Merging markov and dct features for multi-class jpeg steganalysis. In: IS&T/SPIE 19th Annual Symposium Electronic Imaging Science and Technology, January 29-February 1, 2007. LNCS, vol. 6505 (2007)
2. Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., Lendasse, A.: Methodology for long-term prediction of time series. Neurocomputing 70(16-18), 2861–2869 (2007)
3. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
4. Guillén, A., Pomares, H., González, J., Rojas, I., Herrera, L.J., Prieto, A.: Parallel multi-objective memetic rBFNNs design and feature selection for function approximation problems. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 341–350. Springer, Heidelberg (2007)
5. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Phys. Rev. 69, 066138 (2004)
6. The 2nd bows contest (break our watermarking system), watermarking virtual laboratory (wavila) of the european network of excellence ecrypt (2007)
7. Westfeld, A.: Reproducible signal processing (bows2 challenge image database)
8. Provos, N.: Defending against statistical steganalysis. In: 10th USENIX Security Symposium, April 13-17, 2001, pp. 323–335 (2001)
9. Miche, Y., Lendasse, A.: A faster model selection criterion for op-elm and op-knn: Hannan-quinn criterion. In: ESANN 2009: European Symposium on Artificial Neural Networks, Bruges, Belgium (April 2009)

Mobile Applications: MovilPIU and Mobiblio

Roberto Berjón Gallinas, M. Encarnación Beato Gutiérrez, Montserrat Mateos Sánchez,
Miguel Ángel Sánchez Vidales, and Ana Fermoso García

Escuela Universitaria de Informática, Universidad Pontificia de Salamanca,
C/ Compañía 5, 37002 Salamanca, España
`{rberjonga,ebeatogu,mmateossa,masanchezvi,afermosoga}@upsa.es`

Abstract. Mobile devices are a new platform with a lot of possibilities to realize studies and to implement projects. It is a new applications platform very interesting to make innovative projects.

In this article we present two real projects of mobile applications that use some of the most popular mobile services. First, MovilPIU allows to access to academic services using a mobile phone, a PDA or a computer. The second project, Mobiblio, applies mobile technology to libraries world.

1 Introduction

Mobile devices (mobile phones, PDAs, ...) have dramatically evolved from those first devices that were used only for voice communication. The first generation of mobile devices was born in 1979. Those terminals were based on analog technology and using radio waves to transmit voice without any encryption. Currently, these devices are missing.

The fundamental gap in mobile devices was in 2001. In this year appeared the first mobile with a color LCD screen. This feature significantly increased the chances of these devices. In the same year in Japan, it is launched the third generation of mobile phones based on UMTS. The most significant characteristic was the addition of a second camera for video calls. These third-generation devices are being currently used.

The current mobile devices are very powerful. They are able of transmitting, receiving and storing information, connecting to networks and running applications. Therefore, they are a very interesting platform for new research projects.

These devices incorporate obvious possibilities. Besides its market penetration has been spectacular. For example, in Spain in 2007 was exceeded the 100 lines per 100 inhabitants for the first time [4]. Specifically Spain in 2007 reached 109 lines per 100 inhabitants (see Figure 1). A few years ago it was unthinkable to reach the barrier of 100 lines. In the figure 2 can be seen a Pyramid Research survey in 2005 [10]. This study shows the dramatic growth of mobile lines in Spain. This growth has not happened so fast ever with other technologies, and it notes that the market is at a very high saturation levels.

The rest of the paper is structured as follows. Section two presents the actually mainly mobile services. Section three focuses on practical cases, showing MovilPIU and Mobiblio[1] projects and, finally, the conclusions are presented along with possible future work.

---

[1] These projects are included in *Club de Innovación Universitaria* project supported by *Fundación Caja Duero*, *Caja Duero* and *Universidad Pontificia de Salamanca [13].*

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1232–1239, 2009.
© Springer-Verlag Berlin Heidelberg 2009

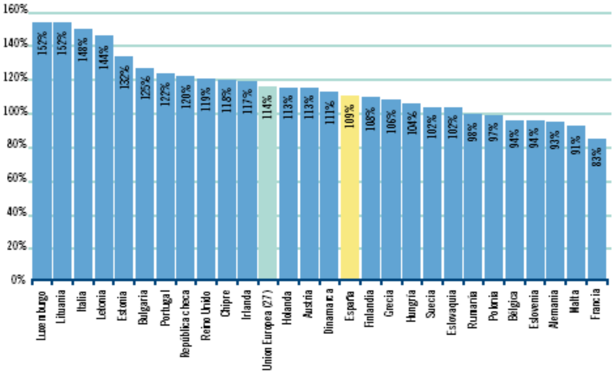


Fig. 1. Mobile phone penetration in the European Union. October 2007 (lines per 100 habitants).

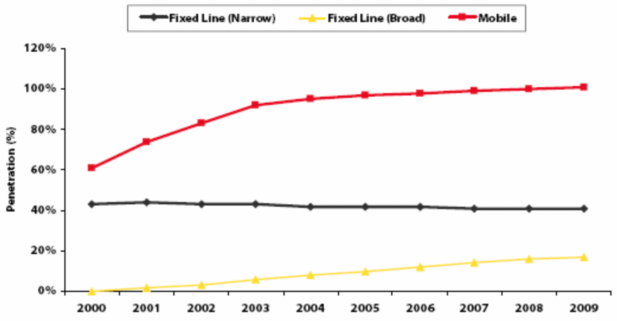


Fig. 2. Mobile phone penetration in Spain

2 Mobile Services

The most important mobile services are:

- **Voice service.** It was the first service in mobile terminals. It is mainly used for communication between people and it is the main service. This service justifies the possession of a terminal. Nowadays, it represents 80% of operators business.
- **Short Message Service.** Popularly it is known as SMS. This service allows sending text between mobile terminals with a maximum of 160 characters. Despite being short text messages, the service is the most popular in history of mobile telephony. Gartner [6] says that in 2006 936,000 million SMS were sent in the world and it is expected to reach 2.3 trillion messages in 2010.
- **Multimedia services.** Much of current mobile phones can store and send ringtones, logos, photographs, graphics or music. These have become small multimedia consoles. Multimedia message reception is not as popular as sending SMS. On the one hand because not all terminals have this service and on the other hand because of the price. In 2006 there were sent 8300 million multimedia messages (less than 1% of total) and in 2010 they will reach 47,400 (less than 5%) [6].
- **Location service.** This service is based on the nature of GSM network. The coverage of a network operator is established on the basis of cells, the size of the cells is dependent on the density of antennas. In urban areas where the number

of antennas is large, the size of the cells is smaller; in this case the location is established with an accuracy of about 100 to 500 meters. Obviously this is less accurate when we are on roads or populations with a lower density of antennas.

Location service is available on all phones, but its precision is not very good and therefore the uses of this service are very limited. However, at present time there are mobile devices already incorporating GPS (Global Positioning System). GPS is a satellite navigation system that can determine the position of an object with accuracy up to centimeters, making it much more interesting.

- **Video services.** 3G/HSDPA terminals allow video calls and audio in real time. In addition, this service lets you send, receive and play video. Using this service, for instance, you can access to video-monitoring circuit through the mobile device or watch television. In any case, video service is not very popular, but in future, experts are betting on it.
- **Mobile software applications.** Mobile terminals have become small computers. They have a powerful processor, an operating system and storage capacity that make possible to run mobile applications on it. To develop this software we can use languages like J2ME (Java Micro Edition) [5, 15], C++ or C#. These features expand significantly the number and type of projects can be done on this new platform.
- **Data service.** The last generation of mobiles has the possibility of Internet connection. So, this connection can be the base for other services. For example, e-mail is now one of the main applications based on data services and it is expected to grow in future.

These are services that can be found mainly in mobile devices today. Some of them will be used in case studies detailed in the following section.

3 Cases Study

Following there are two projects in which we have studied and used some of the mobile services exposed in the previous section.

3.1 MovilPIU

This first project is an academic application. Currently, students can access a range of services offered by the University (student record requesting, obtaining certificates, purchase of dining tickets,...) using their smart card in the UIP (University Information Points). Our goal is to facilitate to students to access to these services. MovilPIU allows access to these services from a computer, a mobile phone or a PDA. For the project will be necessary to achieve:

- **Identify the student:** Currently, the University community has a user account to gain access to computers. Through this account the user is identified and authenticated. The same account will be used to identify users in MovilPIU using Active Directory.
- **Study and implementation of services:** MovilPIU must be able to present information to the user according to the device used to having access to it. We will use the following formats: WML, XHTML and HTML [8].

  Moreover, we need to work with the same information (databases) that we are now used by the UIP, so that the UIP will continue to function without making any changes.

- **Payment:** Some services in the UIP need a pre-payment. For example, if they wish to apply for a mark certificate, students will have to pay the amount of the fee. This payment will continue to be. We use the rules defined by Spanish banks for this payment and we use a Web application to manage this function.

3.1.1 Architecture

The architecture used in MovilPIU is a three-tier design: Model, View and Controller (MVC). Communication between components, controller and model is through SOAP messages. Really the model is a Web service (see figure 3).

Response to the user request is obtained by transforming the XML information retrieved from the model using XSLT stylesheet. Different CSS are applied depending on user-agent. The supported formats are HTML, XHTML and WML.

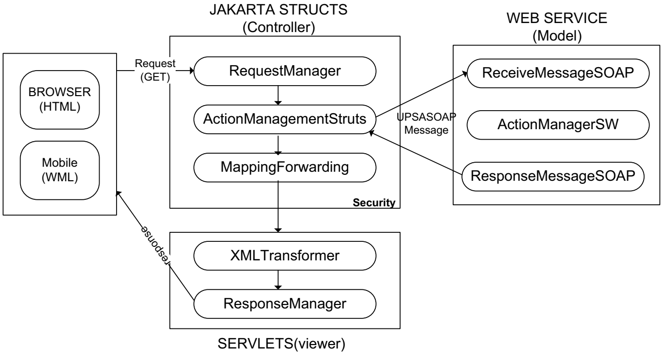


Fig. 3. MovilPIU Architecture

3.1.2 Results

As we can see in figure 4, the user must access via mobile or Web. After that, depending on the user profile, it will be possible to execute different academic functions (figure 5).

MovilPIU is divided into three parts: "Academic Information" (figure 6), "Certificates Request" (figure 7) and "Dining tickets Request" (figure 8). The application works by using the navigation, via mobile and via web, through links.

Parts in which the application is divided are described below.

Academic Information

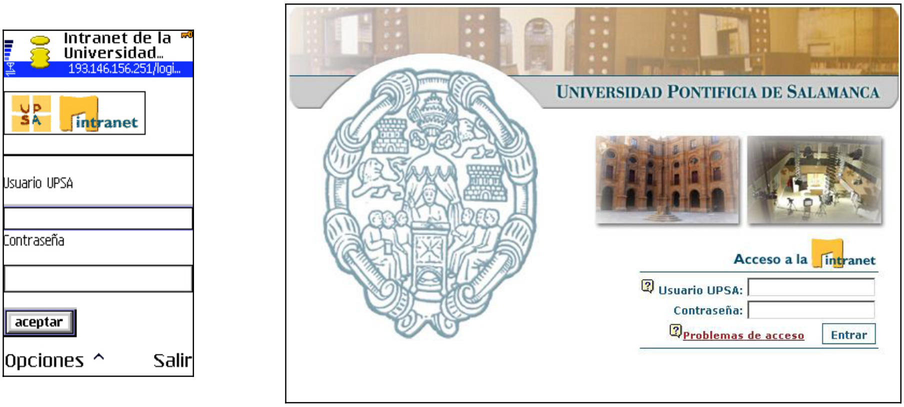
The application allows students to request their academic record. The student can choose between the different graduates that he has studied in the University.

Certificates Request

In this option MovilPIU allows students to request certificates for their qualifications in the University.

Dining tickets Request

Dining tickets are bought in this option. Not only students but also teachers and staff can access to this option and to buy dining tickets. The purchase is recorded and is sent to the bank at the end of the month. The price is showed near to the kind of ticket.



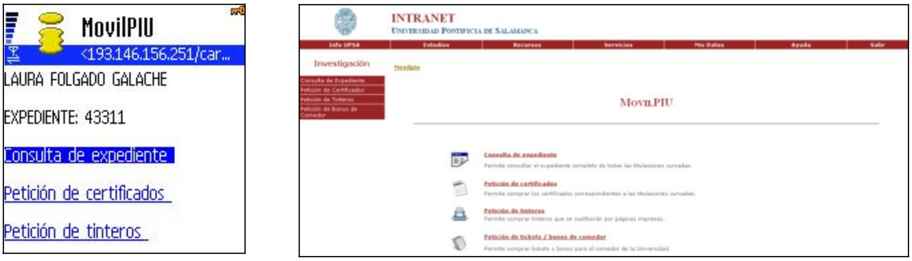
Usuario UPSA

Contraseña

Usuario UPSA:

Contraseña:

Fig. 4. Mobile and Web user identification



LAURA FOLGADO GALACHE

EXPEDIENTE: 43311

Consulta de expediente

Petición de certificados

Petición de tinteros

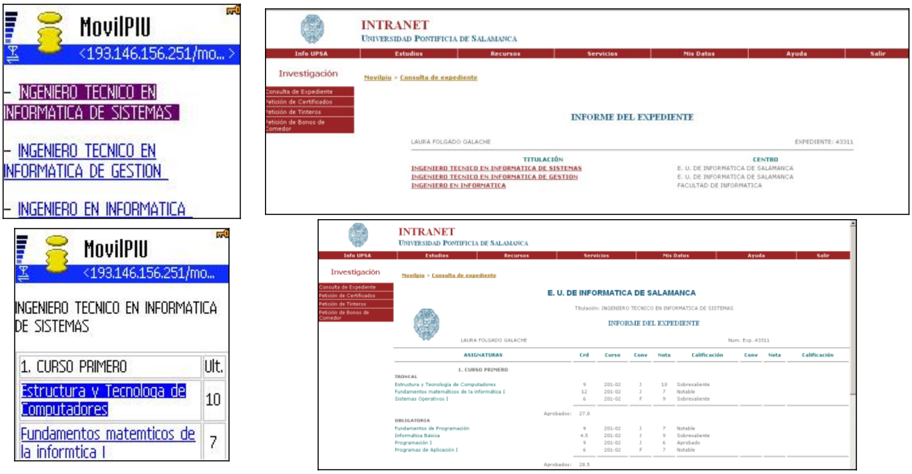
Consulta de expediente
Permite consultar el expediente completo de todas las titulaciones cursadas.

Petición de certificados
Permite comprar los certificados correspondientes a las titulaciones cursadas.

Petición de tinteros
Permite comprar tinteros que se sustituirán por páginas impresas.

Petición de tickets / bonos de comedor
Permite comprar tickets y bonos para el comedor de la Universidad.

Fig. 5. MovilPIU application



- INGENIERO TECNICO EN INFORMATICA DE SISTEMAS
- INGENIERO TECNICO EN INFORMATICA DE GESTION
- INGENIERO EN INFORMATICA

INGENIERO TECNICO EN INFORMATICA DE SISTEMAS

1. CURSO PRIMERO	Ult.
Estructura y Tecnologa de Computadores	10
Fundamentos matemticos de la informtica I	7

INFORME DEL EXPEDIENTE

LAURA FOLGADO GALACHE EXPEDIENTE: 43311

TITULACIÓN	CENTRO
INGENIERO TECNICO EN INFORMATICA DE SISTEMAS	E. U. DE INFORMATICA DE SALAMANCA
INGENIERO TECNICO EN INFORMATICA DE GESTION	E. U. DE INFORMATICA DE SALAMANCA
INGENIERO EN INFORMATICA	FACULTAD DE INFORMATICA

E. U. DE INFORMATICA DE SALAMANCA

Titulación: INGENIERO TECNICO EN INFORMATICA DE SISTEMAS

INFORME DEL EXPEDIENTE

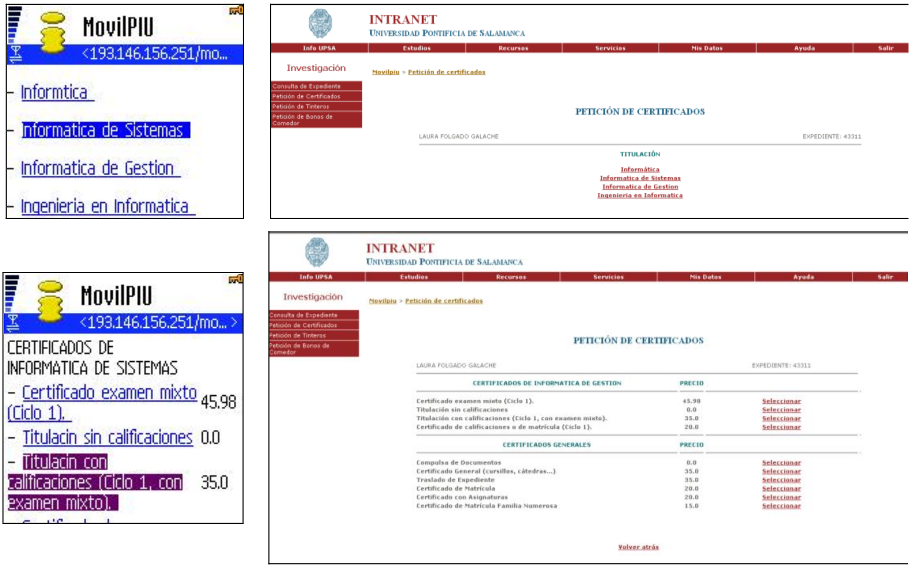
LAURA FOLGADO GALACHE Num. Exp. 43311

ASIGNATURAS	Crd	Curso	Conv	Nota	Calificación	Conv	Nota	Calificación
1. CURSO PRIMERO								
TRONCAL								
Estructura y Tecnología de Computadores	9	201-02	J	10	Sobresaliente			
Fundamentos matemáticos de la informática I	12	201-02	J	7	Notable			
Sistemas Operativos I	6	201-02	F	9	Sobresaliente			
Aprobados:	27.0							
OBLIGATORIA								
Fundamentos de Programación	9	201-02	J	7	Notable			
Informática Básica	4.5	201-02	J	9	Sobresaliente			
Programación I	9	201-02	J	6	Aprobado			
Programas de Aplicación I	6	201-02	F	7	Notable			
Aprobados:	28.5							

Fig. 6. Academic Information

3.2 Mobiblio

This project is to bring mobile technology to libraries, applying it to improve the quality of some basic services offered by them. Specifically we use a messaging system to provide loan services, completed reservations or information available on our loans.



- Informtica
- Informatica de Sistemas
- Informatica de Gestion
- Ingenieria en Informatica

PETICIÓN DE CERTIFICADOS

LAURA FOLGADO GALACHE EXPEDIENTE: 43311

TITULACIÓN

Informática
Informatica de Sistemas
Informatica de Gestion
Ingenieria en Informatica

CERTIFICADOS DE INFORMATICA DE SISTEMAS

- Certificado examen mixto (Ciclo 1).	45.98
- Titulacin sin calificaciones	0.0
- Titulacin con calificaciones (Ciclo 1, con examen mixto).	35.0

PETICIÓN DE CERTIFICADOS

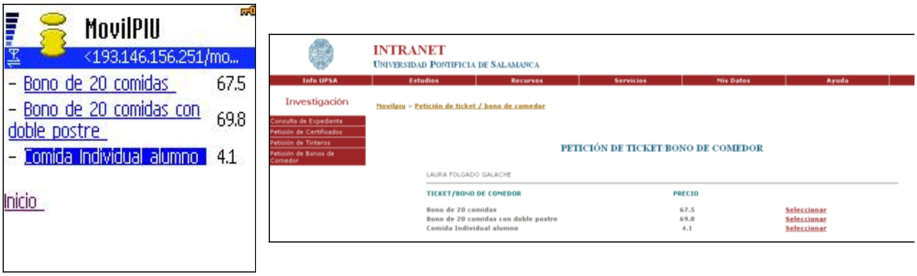
LAURA FOLGADO GALACHE EXPEDIENTE: 43311

CERTIFICADOS DE INFORMATICA DE GESTION	PRECIO	
Certificado examen mixto (Ciclo 1).	45.98	Seleccionar
Titulación sin calificaciones	0.0	Seleccionar
Titulación con calificaciones (Ciclo 1, con examen mixto).	35.0	Seleccionar
Certificado de calificaciones o de matrícula (Ciclo 1).	20.0	Seleccionar

CERTIFICADOS GENERALES	PRECIO	
Compulsa de Documentos	0.0	Seleccionar
Certificado General (cursillos, cátedras...)	35.0	Seleccionar
Traslado de Expediente	35.0	Seleccionar
Certificado de Matrícula	20.0	Seleccionar
Certificado con Asignaturas	20.0	Seleccionar
Certificado de Matrícula Familia Numerosa	15.0	Seleccionar

Volver atrás

Fig. 7. Certificates request



- Bono de 20 comidas	67.5
- Bono de 20 comidas con doble postre	69.8
- Comida Individual alumno	4.1

PETICIÓN DE TICKET/BONO DE COMEDOR

LAURA FOLGADO GALACHE

TICKET/BONO DE COMEDOR	PRECIO	
Bono de 20 comidas	67.5	Seleccionar
Bono de 20 comidas con doble postre	69.8	Seleccionar
Comida Individual alumno	4.1	Seleccionar

Fig. 8. Dining tickets request

In addition, a user can access to the library catalog by mobile phone, request and modify your personal information or renew a book. This is done through a WAP portal that provides access to navigation services [9,14].

In the current situation in the library (see figure 9), users without Internet access should be physically moved to the library to use many of the services (access catalog, users information, alerts, ...). In other cases, the libraries use the mail. The mail has some problems such as slow, problems with changes of address ... So when the user receives the notice may have spent several days. When the user has an Internet connection, these services are faster, more comfortable and the user can access them from anywhere with an Internet connection. The problem is that the user needs an Internet connection but it is not always available.

3.2.1 Project Development

With this project the user can access fast to the library services and from anywhere. This is possible because we use mobile phone as access method.

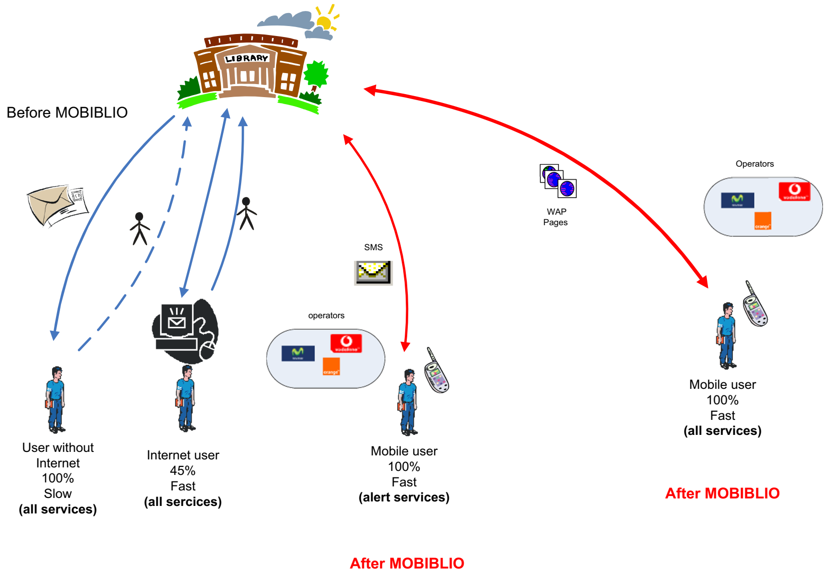

LIBRARY
Before MOBIBLIO
User without Internet 100% Slow (all services)
Internet user 45% Fast (all sercices)
SMS
operators
Mobile user 100% Fast (alert services)
WAP Pages
Operators
Mobile user 100% Fast (all services)
After MOBIBLIO
After MOBIBLIO

Fig. 9. Current and future situation in the library

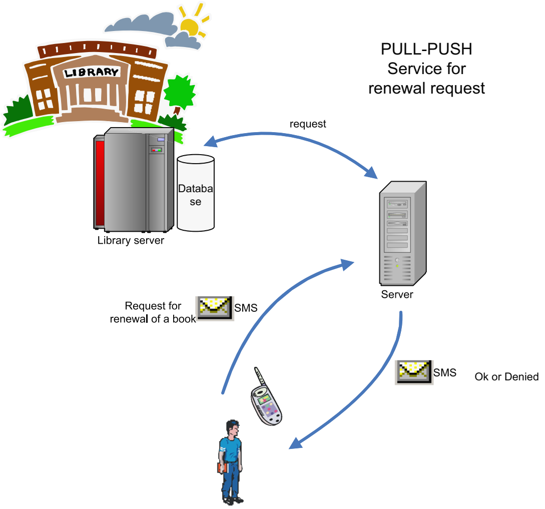
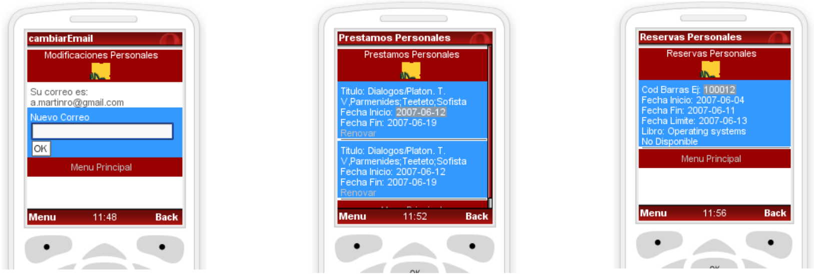

LIBRARY
PULL-PUSH Service for renewal request
request
Databa se
Library server
Server
Request for renewal of a book
SMS
SMS
Ok or Denied

Fig. 10. User record querying and modification

This project has also used the design pattern Model View Controller. This pattern isolates each component, so we can migrate the application to another environment with only changing the data access.

Furthermore, we use WAP technology for communication to the server, using Web browser. The Java programming language to implement the application, JSP (Java Servlet Pages) to implement pages, these pages are displayed on the phone, and JDBC to connect and communicate with the application database.

We work with push and pull-push services. Push is used for announcements and pull-push for the renewal of books (see Figure 10). Project results can be seen in the figure 11.



Su correo es:
a.martinro@gmail.com

Titulo: Dialogos/Platon. T. V,Parmenides;Teeteto;Sofista
Fecha Inicio: 2007-06-12
Fecha Fin: 2007-06-19

Titulo: Dialogos/Platon. T. V,Parmenides;Teeteto;Sofista
Fecha Inicio: 2007-06-12
Fecha Fin: 2007-06-19

Cod Barras Ej: 100012
Fecha Inicio: 2007-06-04
Fecha Fin: 2007-06-11
Fecha Limite: 2007-06-13
Libro: Operating systems
No Disponible

querying and modification.

Fig. 11. User record querying and modification

4 Conclusions and Future Lines

In this report two mobile applications have been showed. These applications use some of the most popular mobile services.

Recently, a new mobile system is born, Android [1,2]. This new system, led by Google, it is a software stack for mobile devices including, among other things, a new java virtual machine optimized for such devices. Developers create their applications using Java programing language. Because it is an open and free platform, it has very good expectations in terms of market penetration, already today, it is being considered as a reference when developing new applications.

References

1. Android (2008), `http://code.google.com/android`
2. Android Market (2008), `http://www.android.com/market`
3. Beato, M.E., Sánchez, M.A., Mateos, M., Fermoso, A., Berjón, R.: Mobile devices: Practical cases on this new applications platform. In: IWPAAMS, pp. 231–240 (2007)
4. Comisión del Mercado de las Telecomunicaciones. Informe anual (2007), `http://www.cmt.es/es/publicaciones/anexos/informe_anual_CMT_2007_web.pdf`
5. De Jode, M.: Programming Java 2 Micro Edition on Symbian OS. John Wiley & Sons, Ltd., Chichester (2004)
6. Garnet (2007), `http://www.garnet.com`
7. Geary, D.M.: Core JSTL. Mastering the SP Standard Tag Library. Prentice-Hall, Englewood Cliffs (2003)
8. Lee, W.M., Foo, S.M., Watson, K., Wugofski, T.: Beginning WAP, WML & WMLScript. Wrox Press, Ltd. (2000)
9. Open Mobile Alliance, `http://www,openmobilealliance.org (2008)`
10. Pyramid research (2007), `http://www.pyramidresearch.com`
11. Telefónica: MovilForum. Web para desarrolladores (2007), `http://www.movilforum.com`
12. Tempos21: Innovación en aplicaciones móviles (2004), `http://www.tempos21.es`
13. Universidad Pontifícia de Salamanca (2005). Proyectos Fundación Caja Duero. Proyectos Movente, `http://www.upsa.es/movente`
14. WAP 2.0 Technical White Paper (2002), `http://www.wapforum.org/what/WAPWhite_Paper1.pdf`
15. Yuan, M.J.: Enterprise J2ME. Developing mobile java applications. Prentice Hall PTR, NJ (2004)

A Case Study of a Pull WAP Location-Based Service Incorporating Maps Services

Montserrat Mateos Sanchez, Roberto Berjon Gallinas,
Miguel Angel Sanchez Vidales, Encarnacion Beato Gutierrez,
and Ana Fermoso Garcia

Universidad Pontificia de Salamanca, C/ Compaia 5, 37002 Salamanca - Spain
{mmateossa,masanchezvi,rberjonga,ebeatogu,afermosoga}@upsa.es

Abstract. The progress in Internet, in network technologies and the rapidly growing of the number of the mobile devices have now resulted that both companies and individuals demanding information and services over mobiles environments, moreover they hope that these services will be provided fitted to their positioning or location context. Services and applications with these requirements could be develop with Location-Based Service (LBS).

In this paper an example of a LBS application that uses the GoogleMaps services is presented. This service or application is placed in a university environment. Both the architecture proposed and the technologies used to develop the application are explained.

Keywords: location based-services, mobile devices, maps, googlemaps, portal wap.

1 Introduction

The mobile technology has suffered a large growing in recent years. This growth has caused that users require new available services and applications over this recent platforms [1][13], moreover they hope that these services could be offered in a fit way to their location context. A LBS [14] uses the actual position of the terminal to provide the service. With LBS the communication and interaction is possible in two ways; the user provides to the services provider her actual context, that is, she provides her position in an automatized way and her preferences and necessities; then the services provider provides to the user the information requested fitted to her necessities and to her actual positioning.

Some possible applications of the LBS [2] are: emergency services to locate to a user in situations that she can't give her position, tourists services to indicate to the user the nearby interesting places or tracking services to use it in packet transport, etc.

The firsts LBS were message-based services, sms (short message services) and mms (multimedia message services)[3]; these services were very limited because of technological restrictions. The terminals that supported these services were simple, with poor user interfaces that didn't offer many possibilities for presentation of information. Nowadays the LBS are developed to be supported by

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1240–1247, 2009.
© Springer-Verlag Berlin Heidelberg 2009

powerful terminals and others technologies [9] such as WAP browsing[17][12], j2me[7] or i-mode are used.

Despite the fact that the mobiles devices and terminals have advanced considerably, nowadays they have some restrictions yet; we have to consider these restrictions when we are going to develop LBS. The mobile devices (pda, phones,...) have usually low-power computing, small memory size, small screens with low resolutions, limited capacities of input, run-time depending of the battery, low and variable bandwith, among others.

The LBS have an add-value when these interact with maps services to provide geographic information; by example, to provide guidance information between the location of mobile device and others geographic points, or to provide services that are about specific distance walking from the position of the terminal.

In this paper we present a case study of LBS developed for a university environment. The application is to help find of accommodations for university students. In the application is included a maps service too. This service is used to place both the user (mobile device) and the searched accommodation. It also provides over the map guidance to the user from her current position to the position of accommodations.

In the rest of the paper we review the different aspects related with LBS: concept, components and associated technologies. Then, in section 3, a case study is presented; we explain the functionality of the application, the architecture and technologies used to implement it. Finally, we expose the main conclusions and future possibilities.

2 Location-Based Services

There are several definitions of the concept *location-based services*; for some authors [11], a LBS is a service of information that is available through mobile devices and it uses the positioning of this to provide the service. With LBS it is possible to develop services and applications available through the mobile devices and these applications could give answers to questions such as *where am i?*, *what are there nearby?* and *how can i go?*

The LBS emerge [15] from 3 technologies: mobile technology, Internet and geographic information systems with spatial databases.

Considering who is the entity that begins the service, the LBS could be classified into two groups: *pull-services* and *push-services*. With *pull-services*, the user carries out the request for the service. With *push-services* the information or service is not directly requested by the user.

2.1 Components of a LBS

All location-based service has four components [14]:

- *Mobile device.* Devices from which the user requests the services and where the results are returned.

- *Mobile communications network.* Mobile phone network by means which the users send from the mobile device both the user data and the service request. Also through it, the answer to information requested by the user is received.
- *Positioning component.* With this component is possible determining the device/user position. The position could be obtained in several ways.
- *Service-Data provider.* This component is the responsible of processing the request service and it also provides the different services, such as, calculating the position, finding a route, finding specific information of interest nearby for a user, etc. The provider may content the requested information by the user, but this information usually will be requested by the data provider to anothers contents providers (yellow pages, directories, etc).

2.2 Methods of Positioning

There are different methods or technologies to compute the positioning of a user/mobile device; we should choose one technology or other depending on the requirements of the application to develop.

- *Network-based positioning.* With this technology is used the mobile communications network. The positioning is carried out by the base stations; the devices send signals to the base station and this is detected by them. The positioning is possible because the coordinates of base stations are known and it it possible to measure (or at last to approximate) the distance between the terminal and the base station. To compute the distance, three or more received signals should be measured. There are different methods to measure the positioning: AOD, TOA, E-OTD. With this technology the accuracy of positioning is variable (between 50 mts and kms); In urban zones the accuracy is higher, but in rural zones it could be lower.
- *Terminal-based positioning.* Also it is named positioning by satellite. This technology uses an infrastructure of satellites of earth orbit and a receiver terminal. The terminals to compute the position are based on the received information through the signals of radio of 3 or more satellites. This technology has a higher accuracy (between 5 mts and 40 mts) than those based on network mobile positioning, but its use is restricted to outdoor. It is independent of mobile phone network. GPS (Global Positioning System) is the most extended but there are others such as Glonass or Galileo.
- *Local positioning.* This is a positioning method oriented to restricted areas and it is based on transmission of signals to short distances. It is basically used for location-based services in indoors such as shop-centers, museums, super-stores, etc. Some of these methods are based on WLAN, Bluetooth or RFID [8].

2.3 Technologies of Development

In order to develop and to exploit LBS is possible to use several widespread technologies such as WAP (Wireless Application Protocol) and J2ME (Java 2 Micro Edition).

WAP [17][12] is an open international standard for applications that uses wireless communications. It is the specification of a set of protocols of communication to normalize the way in that the wireless devices could access and interact to both services and information in a simple and quick way. WAP is oriented to mobile devices with limited screens, smaller keyboards and low bandwidths. This protocol allows to operate to applications and to services over a large number of networks (CDMA, WCDMA, CDPD, UMTS, GSM). It is basically oriented to presentations of contents in the terminal. The version 2.0 presents important improvements with regard to version 1.0. One of these improvements is better designs of the portal contents (it is possible use XHTML and WCSS). WAP is supported by the most of mobile devices that have a microbrowser.

Another possibility is to develop the services using J2ME [7][10]. J2ME is a "mini" version of Java. It is a Java platform specially oriented to devices with more limited capacities than those of a personal computer. It is oriented to the development of applications that run on mobile devices, although it is possible to develop client/server applications which interact by means of network services.

In order to develop location-based applications/services the most suitable is to use WAP [9].

3 Case Study

3.1 Introduction

In this section we present the results of the development of a project that consists on applying the mobile technology to the search of accommodation for university students. Some universities have available services of help for finding accommodation, but the most of them require that users should have to shift to the department of the university that offers this service, or if the service is available through the Web, then the users have to have available a personal computer with Internet connection. This way of searching accommodations could means an additional problem to users because of it is very likely that the users could unknown the city, and therefore, it could be difficult for them both to place the position of an accommodation in a map and to calculate the approximate distance and route between points in a map.

The developed application could have two types of users; On one hand the users that offer their accommodations; these users could introduce their accommodations by means of a form available through the mobile devices. And on other hand, the university students that requires an accommodation; the students could carry out the searches using several criterions: number of rooms, shared, with heating, centrally situated, zone, etc.; the application also places the accommodation on a map, calculates both the distance and route from actual position of student (mobile device) to the accommodation position.

The application is clearly a pull location-based services that incorporates maps service. Because of the requirements of application we decided to develop it using WAP technology; We also decided to use the mobile communication network for

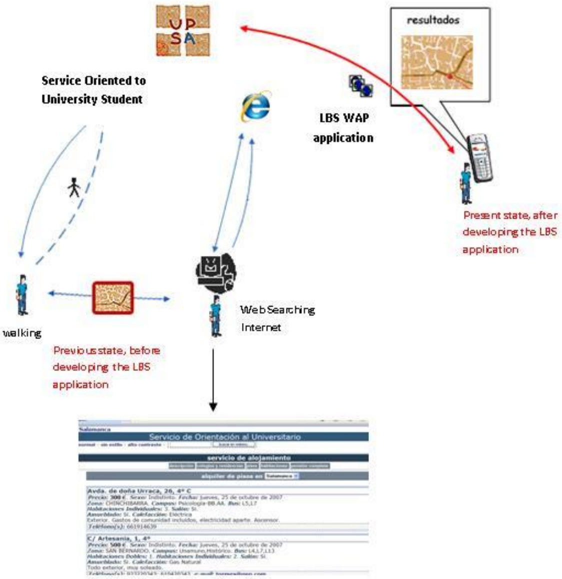


Fig. 1. Before situation and actual situation with the development of the new service

calculating the positioning, because nowadays, there are few phones with GPS, and because, the application will be used in urban zones where the accuracy is high.

3.2 Technologies

The Java language has been used to implement the application. Exactly we have used Java Servlet apis, as well as, JSP (Java Server Pages) to implement the web pages that will be displayed on phone. JDBC has been used for the connection and communication of the application with the Oracle database, where it is stored all managed information of accommodations. According to [5] J2EE is a technology that satisfies the requirements of a Wap-based LBS. The J2EE solutions reduce the cost and the complexity of a distributed development of several levels, which could be more portable and quickly developed and deployed.

By other hand, the application provides both the positioning of accommodations on a map and the route from the positioning of the user towards the accommodation. To implement this functionality we have used the location service that mobile phones network provides, and we have used the apis developed in java that some telecommunications companies have available[1] for this purpose. For implementing the services of maps and guided to the user we have used the apis of maps that Google [4] has available. Exactly we have used the *GoogleStaticMaps* services and *Driving Directions* services.

---

[1] For this application we have used L-Broker api [16] of Tempos-21 company.

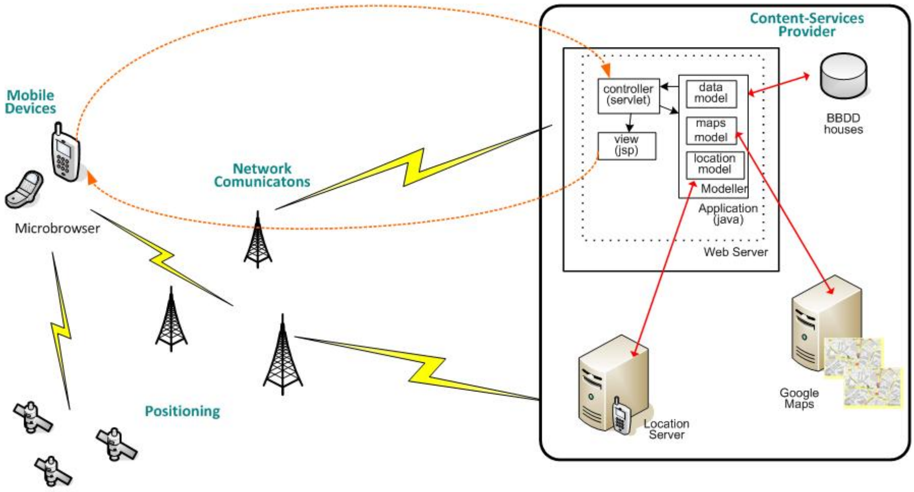


Fig. 2. Architecture used for developing and implementation of the LBS

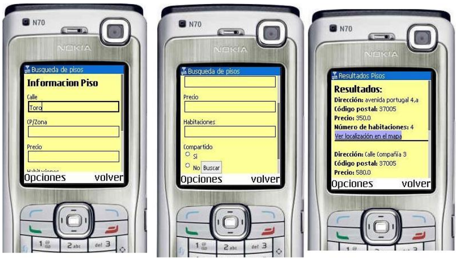
3.3 Architecture

The application developed (figure 2) has the four components of an LBS: mobile device, mobile communications network, positioning component and services-data provider.

The mobile device should connect to services/data provider through the mobile network. This device will be a thin-client that will carry out the request of service to a Web server by means of a micro-browser.

The services/data provider will be an application that runs on a Web server. This application has been developed using a MVC (Model-View-Controller) pattern to separate the part that implements the business logic from which generates the presentation of contents and from the component that takes on all the client requests. In this way, if some component or part should be changed, it will be possible to carry out the change without modify the rest of components.

The controller component will be the component that receives the requests of clients; it has been developed using Servlets. When the controller receives a request, it will pass it to the model and when the model has the answer it will pass it to the view component. The view component will generate the presentation of contents using JSP. As we can see in figure 2, the model component has three independent components: data, maps and positioning. Depending on the request of user, this will be passed to a model or other. If the request is for finding an accommodation with a specific characteristics, then the request will be passed to the data model. If the request if for positioning in a map of one accommodation, then it will be transmitted to maps model; and if it is for guiding of a user from her position towards one accommodation, then the request will be transmitted to both maps and location model.



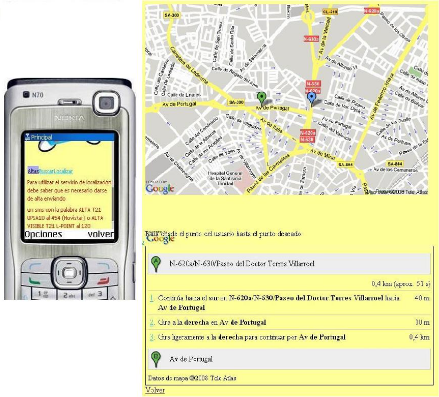
Busqueda de pisos
Informacion Piso
Calle: Toro
CP/Zona:
Precio:

Busqueda de pisos
Precio:
Habitaciones:
Compartido
○ Si
○ No

Resultados Pisos
Resultados:
Dirección: avenida portugal 4,a
Código postal: 37005
Precio: 350.0
Número de habitaciones: 4
Ver localización en el mapa

Dirección: Calle Compañía 3
Código postal: 37005
Precio: 580.0

Fig. 3. Search of an accommodation



Principal
AltasBuscarLocalizar
Para utilizar el servicio de localización debe saber que es necesario darse de alta enviando
un sms con la palabra ALTA T21 UPSA10 al 454 (Movistar) o ALTA VISIBLE T21 L-POINT al 120

Ruta desde el punto del usuario hasta el punto deseado

N-620a/N-630/Paseo del Doctor Torres Villarroel

0,4 km (aprox. 51 s)

1. Continúa hacia el sur en **N-620a/N-630/Paseo del Doctor Torres Villarroel** hacia **Av de Portugal** 40 m
2. Gira a la **derecha** en **Av de Portugal** 10 m
3. Gira ligeramente a la **derecha** para continuar por **Av de Portugal** 0,4 km

Av de Portugal

Datos de mapa ©2008 Tele Atlas

Fig. 4. Positioning in a map of the accommodation and guided to the user

We can see that our system is a LBS with an multilevel architecture which favors the reutilization. It would be valid for any other business, only it would be necessary to change the logic of data access, the rest of elements (maps, positioning,..) could be reutilized.

In figures 3 and 4 we can see some parts (web pages) of application in the mobile device; we can see the type of map and the type of guided provided to the user too.

4 Conclusions

In this paper has been exposed an example of pull Wap LBS that incorporates a maps service and a service of guided of routes to the user. At present, this types of services are more and more required by users because of they would like have available information/services in any moment, in any location and that these services/information are fitted to her positioning.

The service that we have exposed is applied to the search of accommodation for university students but it could be extrapolated to any other business; we would only have to change the logic of business; this is possible thanks to the architecture that we have used to develop the project.

At present new platforms of development as Android are emerging. Also new phones that will work with this platforms are coming out. With the appearance of these new technologies the possibilities of research in this area are increasing significantly.

References

1. Beato, M.E., Sánchez, M.A., Mateos-Sanchez, M., Fermoso, A., Berjón, R.: Los dispositivos móviles: casos prácticos sobre esta nueva plataforma para aplicaciones. In: 6th IWPAAMS. Universidad de Salamanca, pp. 231–240 (2007)
2. Dru, M.A., Saada, S.: Location-Based mobile services: The essentials. Alcatel Telecomunications Review, pp. 71–76 (2001)
3. Fangxiong, W., Zhiyong, J.: Research on a distributed architecture of mobile GIS based on Wap. Wuhan University (2004)
4. GoogleMaps, `http://code.google.com/intl/es/apis/maps/documentation/`
5. Haifeng, M.: Distributed GIS for Agriculture based on J2EE. Geomatics and Information Science of Wuhan University 29(2), 142–143 (2004)
6. `http://java.sun.com/j2ee/overview.html`
7. `http://java.sun.com/javame/index.jsp`
8. Krishnamurthy, P., Pahlavan, K.: Wireless Communications. Telegeoinformatics, pp. 111–142. CRC Press, Boca Raton (2004)
9. Lei, Y., Hui, L.: Which one should be chosen for the Mobile Geographic Information Service Now, Wap vs. i-mode vs. J2ME? Mobile Networks and Applications 11, 901–915 (2006)
10. Lei, W., Yin, S.: J2ME technology on mobile information device. Modern computer (1), 17–20 (2004)
11. Open Geospatial Consortium. Open Location Services 1.1 (2005)
12. Open Mobile Alliance, `http://www.openmobilealliance.org`
13. Sanchez, M.A., Beato, E.: Proyectos de Innovación Tecnológica Móvil y Chip. Publicaciones Universidad Pontificia de Salamanca. Salamanca, pp. 43–48 (2008)
14. Schiller, J.H., Voisard, A.: Location-based services. Morgan Kaufmann Publishers, San Francisco (2004)
15. Shiode, N., Li, C., Batty, M., Longley, P., Maguire, D.: The impact and penetration of Location Based Services. Telegeoinformatics: Location-Based Computing and Services. CRC Press, Boca Raton (2004)
16. Lbroker, `http://www.tempos21.com/web/files/productos/Lbroker.pdf`
17. WAP 2.0 Technical White Paper (2002), `http://www.wapforum.org/what/WAPWhitePaper1.pdf`

A Mobile Tourist Decision Support System for Small Footprint Devices

Wouter Souffiau[1,2,*], Joris Maervoet[1], Pieter Vansteenwegen[2],
Greet Vanden Berghe[1], and Dirk Van Oudheusden[2]

[1] KaHo St.-Lieven, Information Technology,
Gebr. Desmetstraat 1, 9000 Ghent, Belgium
{wouter.souffriau,joris.maervoet,greet.vandenberghe}@kahosl.be
[2] Katholieke Universiteit Leuven, Centre for Industrial Management,
Celestijnenlaan 300A, 3001 Leuven (Heverlee), Belgium
{pieter.vansteenwegen,dirk.vanoudheusden}@cib.kuleuven.be

Abstract. This paper presents a mobile tourist decision support system that suggests personal trips, tailored to the user's interests and context. The system enables planning a customised trip that maximises the interest of the tourist, while taking the opening hours of the points of interest (POI) and the available time into account. The planning problem is modelled as an orienteering problem with time windows, which is a hard combinatorial optimisation problem. It is solved by an iterated local search metaheuristic procedure, resulting in a personal trip. This procedure is implemented and tested on a mobile phone. Despite the limited computational resources of a small footprint device, the system succesfully solves instances up to 50 POIs in an acceptable execution time. Not more than 1% of the solution quality turned out to be sacrificed in order to keep the worst–case execution time under 5 seconds.

Keywords: Tourism, Orienteering Problem with Time Windows.

1 Introduction

If a tourist wants to visit every tourist attraction during a citytrip to e.g. Paris, Barcelona or London, a considerable amount of time is required. However, most tourists are restricted in available time and budget and thus have to make a selection of the most interesting places to visit. Therefore, a tourist gathers information from different sources about the different POIs, makes a selection of POIs to visit and plans a route between these, taking into account the available time and the opening hours of the different POIs. This is a rather complex and time consuming process.

The personalised electronic tourist guide is an excellent aid for tourists who want real support for this kind of problem [1,2]. These devices address the so–called Tourist Trip Design Problem [3] by suggesting, at short notice, a (near) optimal selection of POIs and a route connecting them, taking into account the weather, opening hours, crowded places and personal preferences. Kramer

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1248–1255, 2009.
© Springer-Verlag Berlin Heidelberg 2009

et al. [4] have analysed the diversity of gathered tourist interest profiles and conclude that they are surprisingly diverse. This conclusion supports the idea of creating personalised tours instead of proposing generic visitor tours.

The Dynamic Tour Guide [5] was the first to calculate personal tourist trips on–the–fly. In order to construct a tour, the system uses a branch–and–bound algorithm to connect so–called Tour Building Blocks, while maximising Interest Matching Points that reflect the user's personal interest. Souffriau et al. [6] use a metaheuristic approach, namely Guided Local Search, to solve the simplest form of the Tourist Trip Design Problem, in which a plan has to be devised only taking the maximum allowed distance into account. They clearly outperform the Dynamic Tour Guide system for problem instances with over 50 POIs. This paper extends that approach by taking opening hours of POIs into account. Moreover, the feasibility of the planning algorithm on a mobile device, with limited computational resources, is examined.

The remaining of the paper is structured as follows. Section 2 describes the automated tourist decision support system. Section 3 presents the computational results of the proposed system on a small footprint mobile device. Finally, Sect. 4 concludes the paper and points out directions for further research.

2 Automated Tourist Decision Support System

The automated tourist decision support system suggests personal trips by first capturing the tourist's operational constraints and interests through a small questionnaire. These interests are used to predict a personal interest score for each POI the tourist is able to visit within his constraints. The coordinates and opening hours of the POIs, the personal interest scores, the tourist's location and available time give rise to a specific instance of a "tourist trip design problem". Next, a heuristic algorithm solves this instance, resulting in a personal trip, tailored to the user's interests, current location, destination, available time and taking into account opening hours. Section 2.1 describes how the problem instance is generated, Section 2.2 presents the algorithm to solve the instance.

2.1 Tourist Trip Design Problem Instance Generation

The decision support system contains a spatially enabled database with all the POIs of the region. Every POI is described by its GPS coordinates, opening and closing hours and an average visiting duration. A POI belongs to exactly one type: abbeys, beguinages, castles, churches, cemeteries, musea, nature, statues or "various buildings". Moreover, each POI is manually classified using the following set of categories: archaeology, architecture, classical art, "local markets and streets", modern art, nature, religious art and science. A system administrator determines the degree to membership of each category: not, a bit, mostly or absolutely of interest.

When a tourist requests a personal trip proposal, this database is queried for relevant POIs. First, a selection of reachable POIs is retrieved. Therefore, the

Table 1. Personal interest score calculation example

		POI value	Tourist interest	Subtotal	Total
Type score	Interest in abbey		2		6
Category score	archaeology	3	2	**6**	
	Architecture	2	1	**2**	
	Classical art	0	2	0	
	Local markets and streets	0	0	0	
	Modern art	0	2	0	
	Nature	1	1	1	
	Religious art	3	3	**12**	
	Science	0	1	0	20
Total score					26

total set of POIs is filtered based on the user's current location (obtained by the GPS signal), his destination, the start and end time of each route. Only the set of POIs that can be visited within the user's operational constraints, are retrieved.

Next, the prediction of the personal interest in each POI is computed, resulting in a score for each possible visit. This interest score is the sum of two components: the type score and the category score. The tourist states his interests in the different POI types and categories: not, a bit, mostly or absolutely interested, quantified to the values 0, 1, 2 and 3. The *type score* of a POI is equal to three times the interest value of the tourist in the particular type. If the tourist is not interested in the type, the total score will be set to 0, and the POI will be removed from the set. A type score equal to 9 indicates that the tourist must see the POI. Therefore, a bonus of 3 is added. In order to calculate the *category score* of a POI, the degree of membership to each interest category is also quantified by 0, 1, 2 or 3. The POI values are multiplied by the user's interest for each corresponding category. Similarly, if this product is 9, an additional bonus of 3 is awarded. The *category score* for a POI is the sum of the three highest scoring categories, with a maximum of 36. The two scores are added and form the *personal interest score* of the tourist in the POI. For the sake of clarity, Table 1 presents an example of the interest score calculation of a tourist in a POI.

At this moment, a set of locations with a score, a visiting duration, opening and closing time is available. Combined with the current location and the destination of the tourist, the start and end time, this set forms an instance of the tourist trip design problem.

This interest capturing and modelling system is implemented as part of CityTripPlanner[1], an online platform that allows to plan tourist trips in the historic city of Leuven, Belgium. The site is operational since October 2008 and numerous test users recognise that the POI scores match their personal interest.

This web application allows to plan multiple day visits in advance, which requires substantial computational resources. When actually executing the trip,

---

[1] www.citytripplanner.com

unexpected events can make the planning infeasible. A mobile device can overcome this problem by dynamically altering the plan. The mobile device only has to take the current day into account, and can discard POIs that are planned to be visited on other days.

2.2 Selection and Routing Algorithm

The Orienteering Problem (OP) [7] serves as a basic model for the Tourist Trip Design Problem. In the OP a set of n POIs i is given, each with a score S_i that reflects the personal interest. The current location and the destination are given. The time t_{ij} needed to travel from POI i to j is known for all location pairs. A given T_{max} limits the time to visit POIs. The goal of the OP is to determine a single route, limited to T_{max}, that maximises the total collected interest score. Each POI can be visited at most once. The OP with Time Windows (OPTW) is an OP with opening hours for each POI, an opening time and a closing time in between which a visit has to take place.

The OPTW is a hard combinatorial optimisation problem. It has to be solved on–the–fly in order to present a personal trip to the tourist dynamically. Calculating the optimal solution using traditional exact methods from operations research literature would require vast amounts of computational resources and time, unavailable in this application scenario. Therefore, the authors opt to use a metaheuristic method for solving the OPTW in order to obtain near optimal solutions in an acceptable amount of time. Vansteenwegen et al. [8] developed an efficient Iterated Local Search (ILS) metaheuristic procedure to tackle the Team OPTW, which is an extension of the OPTW, that attempts to maximise the score of multiple tours, each limited to T_{max}. The metaheuristic starts from a set of empty tours, which only contain the start and end, and then alternates between insertion and shaking procedures for a number of iterations. The best solution found is updated at each iteration, and returned at the end. This paper discusses the application of the metaheuristic procedure to the single tour version of the problem.

The insertion procedure iteratively adds POIs to the current tour, to maximise the collected interest score. Adding a POI to a tour increases the tour's total score and results in a decrease of the available time. For each POI that can be inserted, its cheapest insertion position in the tour is determined, namely, the position with the smallest required insertion time. The POI with the highest ratio square of POI score over insertion time will be selected for insertion. When adding a POI to the route, the arrival time of all POIs after the insertion position will increase. In order for the route to remain feasible, the end of each visit is not allowed to exceed the closing time of the POI. Religiously checking all other visits for feasibility each time a POI is considered for insertion in the route, is computationally unaffordable. This is avoided by recording *MaxShift*, which is defined as the maximum time a visit can be delayed without other visits becoming infeasible. POIs are added in best improving fashion, until the available time in the tour disallows the addition of more POIs, thus reaching a local optimum. At this point, the best solution found so far, is updated.

In order to escape from this local optimum, the shaking procedure intelligently removes visits from the current solution. The behaviour of the shaking procedure is regulated by two parameters. *NumberToRemove* indicates how many consecutive visits should be removed. *StartPosition* indicates the place in the route to start removing POIs. If during the removal the end point is reached, the removal continues after the start point. After the removal, all subsequent visits are shifted towards the beginning of the tour, to avoid unnecessary waiting. Visits before the removed visits require only MaxShift updating.

Algorithm 1 presents the ILS pseudo code. All the parameters of Shake are initialised to one. The heuristic loops until no improvements for the incumbent best solution are identified during *MaxIter* iterations, which is a predefined parameter. First, the Insert procedure is applied until a local optimum is reached. If this solution is better than the incumbent, the solution is recorded and *NumberToRemove* is reset to one. Next, Shake is applied. After each shake step, *StartPosition* is increased by the value *NumberToRemove*, that is also increased by one. If *StartPosition* is greater than the size of the route, this size is subtracted. If *NumberToRemove* reaches $\frac{n}{3}$, it is reset to one.

```
while NumberOfIterationsNoImprovement < MaxIter do
    Insertion;
    if Solution > BestFound then
        BestFound=Solution;
        NumberToRemove=1;
        NumberOfTimesNoImprovement=0;
    else
        NumberOfTimesNoImprovement++;
    Shake(NumberToRemove, StartPosition);
    StartPosition= StartPosition + NumberToRemove;
    NumberToRemove++;
    if StartPosition > size of the route then
        StartPosition=StartPosition - size of the route;
    if NumberToRemove = $\frac{n}{3}$ then
        NumberToRemove=1;
Return BestFound;
```

Algorithm 1: Iterated Local Search for the OPTW

By using the shake parameters as described above, other visits are removed during every iteration during the entire procedure, every visit is probably removed at least once. This proves an excellent technique to efficiently explore the solution space and correct unfortunate decisions taken earlier by the insertion procedure. The algorithm is described and evaluated in detail in [8].

3 Computational Results

The computational tests of Vansteenwegen et al. [8] are executed in a high performing desktop environment, an Intel Core 2 with 2.5 GHz processor and 3.45

Table 2. Solution quality obtained on a mobile device compared to original setting

n	$MaxIter$	#Points Selected	Score Gap
10	10	5.23	0.00%
10	20	5.23	0.00%
10	50	5.23	0.00%
10	100	5.23	0.00%
20	10	6.05	0.36%
20	20	6.05	0.00%
20	50	6.05	0.00%
20	100	6.05	0.00%
30	10	6.51	0.67%
30	20	6.54	0.24%
30	50	6.54	0.05%
30	100	6.51	0.00%
40	10	6.38	3.43%
40	20	6.64	0.79%
40	50	6.85	0.00%
40	100	6.85	0.00%
50	10	6.85	1.85%
50	20	6.95	0.89%
50	50	7.05	0.09%
50	100	7.00	0.00%

Table 3. Minimal, average and maximal execution times on the mobile device compared to the desktop setting

n	$MaxIter$	Min Time (ms)		Avg Time (ms)		Max Time (ms)	
		Desktop	Nokia N85	Desktop	Nokia N85	Desktop	Nokia N85
10	10	0	21	0	51	2	213
10	20	1	37	1	65	2	95
10	50	2	85	2	145	4	184
10	100	3	164	4	277	7	344
20	10	1	39	1	118	4	315
20	20	1	75	3	197	8	557
20	50	3	168	6	404	12	865
20	100	6	320	12	748	23	1 405
30	10	1	81	2	213	6	548
30	20	2	147	4	403	12	1 521
30	50	6	354	11	812	31	2 368
30	100	11	657	22	1 477	51	3 773
40	10	2	115	3	320	14	1 229
40	20	3	218	7	575	19	1 920
40	50	7	449	18	1 347	65	3 892
40	100	14	875	31	2 306	84	6 580
50	10	2	116	5	374	20	1 178
50	20	4	261	9	691	31	2 492
50	50	10	583	22	1 810	62	9 008
50	100	19	1 096	43	3 114	158	12 054

GB Ram. This paper assesses the feasibility of the algorithm on a small footprint mobile device. We implemented the algorithm in the Java Micro Edition environment[2] and ran a series of benchmark tests on a Nokia N85 mobile phone, a device with a CPU clock rate of 369 MHz. The benchmarks are based on the 68 test instances of Righini and Salani [9] for the OPTW. The size of these test instances is altered by varying the number of POIs n. Only the start location and the first n POIs are retained in each problem, the rest is discarded. Experiments are performed for n equal to 10, 20, 30, 40 and 50. Not all problem instances of Righini and Salani [9] are used in this experiment. The instances with 50 POIs are discarded, as they are already reduced versions of the larger instances, resulting in 39 instances in total.

The experiments of Vansteenwegen et al. [8] are performed with parameter setting $MaxIter$ equal to 150. In that setting, the results of the ILS metaheuristic deviate from the optimal solution by only 1.8% using 1 second of computation time on average. On a mobile device, computational resources are very limited. Moreover, the tourist application needs almost real–time response as the algorithm is executed each time the current plan becomes infeasible due to unexpected events. In order to decrease the execution time, the value of $MaxIter$ is reduced. Experiments have been performed with $MaxIter$ equal to 10, 20, 50 and 100. Table 2 presents for each combination of n and $MaxIter$ the average number of POIs selected and the score gap between the original setting of $MaxIter$ equal to 150 and the reduced setting. The score gap is calculated by

$$\frac{Score_{MaxIter=150} - Score_{MaxIter=ReducedSetting}}{Score_{MaxIter=150}} \tag{1}$$

For problem instances with $n = 10$, $MaxIter$ can be set to 10 without loss of solution quality. For $n = 20$, $MaxIter$ is set to 20. For larger instances, a $MaxIter$ setting of 50 seems appropriate, depending on the execution time. Table 3 presents the minimum, average and maximum execution times for the different experimental settings. For the sake of comparison, the experiment has also been performed in a desktop setting, using a HP laptop with a 1.83 GHz Genuine Intel processor equipped with 1 GB of memory.

On average, the maximum calculation time is 80 times higher for the mobile device compared to the desktop setting. The maximum calculation time can be seen as a worst–case scenario for the dynamic mobile tourist application. If almost real–time is defined as a maximum calculation time of 5 seconds, $MaxIter$ should be reduced to 20 for instances of 50 POIs. This reduction results in a quality loss of only 0.89%.

4 Conclusions

This paper presents a tourist decision support system for mobile devices that enables planning a customised personal trip, maximising the interest of the tourist, while keeping the opening hours of the POIs and the available time of the tourist

---

[2] http://java.sun.com/javame

into account. The planning problem is modelled as an orienteering problem with time windows and is solved by a metaheuristic procedure, which is implemented and tested on a small footprint mobile device. Notwithstanding the limited computational resources, the system successfully solves instances up to 50 POIs in an acceptable execution time. For the larger instances, not more than 1% of the solution quality is sacrificed in order to keep the worst–case execution time under 5 seconds. This small quality loss is insignificant for the application, considering the inherent shortcomings of quantifying a tourist's personal interest in a location.

Future work includes planning using public transportation between POI visits, with the mobile device, and a field test to evaluate the system.

Acknowledgements

Pieter Vansteenwegen is a post doctoral research fellow of the Fonds Wetenschappelijk Onderzoek - Vlaanderen (FWO).

References

1. Malaka, R., Zipf, A.: Deep Map - challenging it research in the framework of a tourist information system. In: ENTER, Barcelona, Spain (2000)
2. Cheverst, K., Davies, N., Mitchell, K.: The role of adaptive hypermedia in a context-aware tourist guide. Communications of the ACM: Special Issue on Adaptive Web-Based Systems and Adaptive Hypermedia 45, 47–51 (2002)
3. Vansteenwegen, P., Van Oudheusden, D.: The Mobile Tourist Guide: an OR Opportunity. OR Insight 20(3), 21–27 (2007)
4. Kramer, R., Modsching, M., ten Hagen, K.: A city guide agent creating and adapting individual sightseeing tours based on field trial results. International Journal of Computational Intelligence Research 2(2), 191–206 (2006)
5. ten Hagen, K., Kramer, R., Hermkes, M., Schumann, B., Mueller, P.: Semantic matching and heuristic search for a dynamic tour guide. In: Information and Communication Technologies in Tourism, Springer, Heidelberg (2005)
6. Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., Van Oudheusden, D.: A personalised tourist trip design algorithm for mobile tourist guides. Applied Artificial Intelligence 22(10), 964–985 (2008)
7. Tsiligirides, T.: Heuristic methods applied to orienteering. J. Oper. Res. Soc. 35(9), 797–809 (1984)
8. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: Iterated local search for the team orienteering problem with time windows. Computers & O.R. (accepted for publication, 2009)
9. Righini, G., Salani, M.: Dynamic programming for the orienteering problem with time windows. Technical Report 91, Dipartimento di Tecnologie dell Informazione, Universita degli Studi Milano, Crema, Italy (2006)

Stereo-MAS: Multi-Agent System for Image Stereo Processing

Sara Rodríguez[1], Juan F. De Paz[1], Javier Bajo[2], Dante I. Tapia[1], and Belén Pérez[1]

[1] University of Salamanca
[2] Pontifical University of Salamanca
`{srg,fcofds,jbajope,dante,lancho}@usal.es`

Abstract. This article presents a distributed agent-based architecture that can process the visual information obtained by stereoscopic cameras. The system is embedded within a global project whose objective is to develop an intelligent environment for location and identification within dependent environments that merge with other types of technologies. Vision algorithms are very costly and take a lot of time to respond, which is highly inconvenient if we consider that many applications can require action to be taken in real time. An agent architecture can automate the process of analyzing images obtained by cameras, and optimize the procedure.

Keywords: Stereoscopy, stereo cameras, artificial vision, MAS, agents, correspondence analysis, dependent environments.

1 Introduction

One of the greatest challenges for Europe and the scientific community is to find more effective means of providing care for the growing number of people that make up the disabled and elderly sector. The importance of developing new and more cost effective methods for administering medical care and assistance to this sector of the population is underscored when we consider the current tendencies. Multi-agent systems (MAS) and intelligent device based architectures have been examined recently as potential medical care supervisory systems [1][7][6][3] for elderly and dependent persons, given that they could provide continual support in the daily lives of these individuals.

The study of artificial vision, specifically stereoscopic vision, has been the object of considerable attention within the scientific community over the last few years. Image processing applications are varied and include aspects such as remote measurements, biomedical images analysis, character recognition, virtual reality applications, and enhanced reality in collaborative systems, among others.

The main topic of our research is part of a larger, global project whose objective is to develop a system for the care and supervision of patients in dependent environments, providing an environment capable of automatically carrying out location, identification and patient monitoring tasks. Such an environment would also allow medical personnel to supervise patients and simulate situations remotely via a virtual environment. In order to reach this objective, artificial intelligence techniques, intelligent agents and wireless technologies are used.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1256–1263, 2009.
© Springer-Verlag Berlin Heidelberg 2009

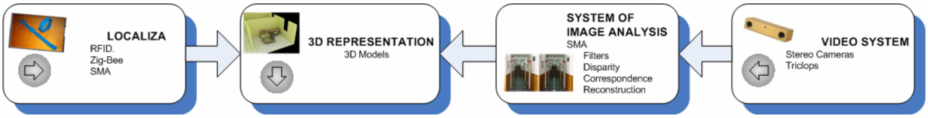


Fig. 1. System for the care and supervision of patients in dependent environments

Within the larger scope of the project, the majority of the current attention is being focused on video surveillance, analysis and 3D representation modules. Our research specifically focuses on the development of an agent-based distributed architecture that allows the visual information obtained by stereoscopic cameras to be processed.

The description of the overall project, along with a summary of the current state of the art is presented in sections 1 and 2 of this article. Section 3 focuses on the techniques used for analyzing the images. Once all the tools have been demonstrated, sections 4 and 5 will present the system proposal itself, and the results and conclusions that have been obtained.

2 State of the Art

This section will focus on stereoscopy and multi-agent systems (MAS), the primary mechanisms used in our system. Traditionally, the use of stereoscopy as a technique for reconstructing images has dealt with two problems. Using a two-dimensional pair of images with spatial coordinates (u,v), the left image (L) and right image (R), *the correspondence problem* attempts to find which two pixels $m_L(u_L,v_L)$ from the left image and $m_R(u_R,v_R)$ from the right image correspond to the same pixel M in three-dimensional space (X,Y,Z). Once these pixels have been found, *the reconstruction problem* attempts to find the coordinates for pixel M[10].

Addressing the correspondence problem is undoubtedly the most difficult task. Since there are generally several possibilities for choosing the element in image R that corresponds to the element in image L, the stereo correspondence problem is said to be ambiguous. It is necessary to determine which characteristics can be applied in order to reduce the ambiguity as much as possible.

The ultimate goal of *reconstruction* is to find the coordinates for pixel M (x,y,z) based on the coordinates from the projections for the same point over the images (u_L,v_L) and (u_R,v_R) [10].

The use of deliberative BDI (*Belief, Desire, Intention*) agents [2][4][8][17] is essential in the development of the platform we are proposing. Apparently, the human visual system deals with a high level of specialization when it comes to classifying and processing the visual information that it receives, such as reconstructing an image by texture, shadow, depth, etc. Computationally, it is difficult to compete with such specialization and separate from an image only the relevant information for any particular purpose. In response to this problem, we propose implementing an algorithm over a distributed agent-based architecture that will allow visual information contained in an image to be processed in real time. Because the system is capable of generating knowledge and experience, the effort involved in programming multiple tasks will also be reduced since it would only be necessary to specify overall objectives, allowing the agents to cooperate and achieve the stated objectives [14].

3 Image Analysis: Phases and Techniques

Techniques for three-dimensional image reconstruction can be classified as both passive and active[9]. Each type presents characteristics that are specific to the process of visual interpretation, but all of them present what could be considered to be common phases in our analysis. The idea is that after processing an image, we can define the resulting depiction as an iconic representation of the visible world. This "iconic model" does not need to represent all of the details from the image, only those that are necessary for developing the task we are interested in. The sequence of actions to take in order to complete the image analysis has been divided into the following modules (included within the *image analysis system* that is illustrated in figure 1):

The *data entry module* that captures the images. Define the number of cameras that will be used, their placement, etc. In this case, the camera that was chosen for this case is the Point Grey [15] Bumblebee2, model BB2-COL-ICX424. The *filtering model* reduces noise, improves contrast, sharpens edges or corrects blurriness. Some of these actions can be carried out at the hardware level, which is to say with the features included with the camera [15]. The *processing module* can be considered the heart of the system since it is where the algorithms are applied to analyze disparities and the correspondence of the stereoscopic pairs, and where the distance measurements for the camera are obtained. The measurements will prove useful in the next phase for reconstructing the image. For this phase, the *position recognition* and *3D representation modules* will model the image with the data that is received. We will focus on the *processing module*.

According to Marr and Poggio[11], there are three steps involved in image reconstruction. They are (i) select a specific pixel from the object in one of the images (preprocessing); (ii) find the same pixel in the corresponding image (correspondence analysis); (iii) measure the relative difference between the two pixels (disparity analysis and obtaining distance).

The aim of *preprocessing* is to identify the representative characteristics of each image [13] that will be used to complete the analysis. A *characteristic* is a relevant piece of information for completing the computational task. It can be an edge, corner or blob. Many vision algorithms apply characteristics detector as the first step, and as a result, there is a large number of detectors that have already been developed[5][19][12]. With artificial vision, edge detection is the most commonly used and reliable techique. The Canny algorithm[5] is considered one of the best methods for edge detection.

Regarding the problem for *obtaining correspondence* described in section 2, there are several strategies that can be classified in different ways[9].

Area based techniques consider the captured images to be a transferred two-dimensional signal. For each one of the pixels in the image, they try to make a transfer, minimizing certain criteria (correlation). One of the most simple techniques is the Sum of Absolute Differences (SAD), since it operates exclusively with whole numbers. Given a pixel with coordinates (x, y) in the left image, a correlation index C(x, y, s) is calculated for each displacement s for the correlation window in the right image. To calculate the correlation index,

$C(x,y,s)=\sum_{u=-w,v=-w}^{u=w,v=w}|\,I_l(x+u,y+v)-I_r(x+u+s,y+v))\,|$ where 2w + 1 is the size of the window centered on the pixel located at position (x, y) and I_l, I_r are the gray values for the pixels in the left and right images respectively. The disparity $d_l(x, y)$ between the left and right image pixels is defined as displacement s which minimizes the correlation index: $d_l(x,y) = arg\ min_s\ C(x,y,s)$. The bookstore that was used in the project (Triclops SDK) establishes a correspondence between the images using this technique.

The *techniques based on features* obtain high quality primitives (edge, segments, curves, regions, etc.) that store a set of properties that remain unchanged with the projection. Among the different methods based on characteristics, there is the Marr-Poggio computational theory[11], the Pollard-Mayhew-Frisby computational theory[16], or the techniques based on border segments. The PMF theory assumes two fundamental restrictions: the features contained in line "n" from the left image should likewise appear (allowing for certain disparity) in line "n" from the right image, thus the correspondence process would be carried out only between those features that are located on the same line in both images. The second restriction is given by the gradient disparity (GD) concept.

Rubio de Lemus [18] applies a comparison and evaluation methodology of search algorithms for correspondences in stereoscopy, showing their most relevant aspects.

The *disparity calculation* allows us to obtain the depth for each of the pixels on the image, obtaining one single image as the disparity map. Given that there is a direct correlation between the depth of the objects in an image and the disparity with a stereo pair, we can use the information from the disparity map as relative values for the depth of the objects.

4 Stereo-MAS

The process of stereoscopy vision is implemented over a distributed agent-based architecture, which allows it to run tasks in parallel using each service as an independent processing unit. It should be noted that because of the high computational costs, not all

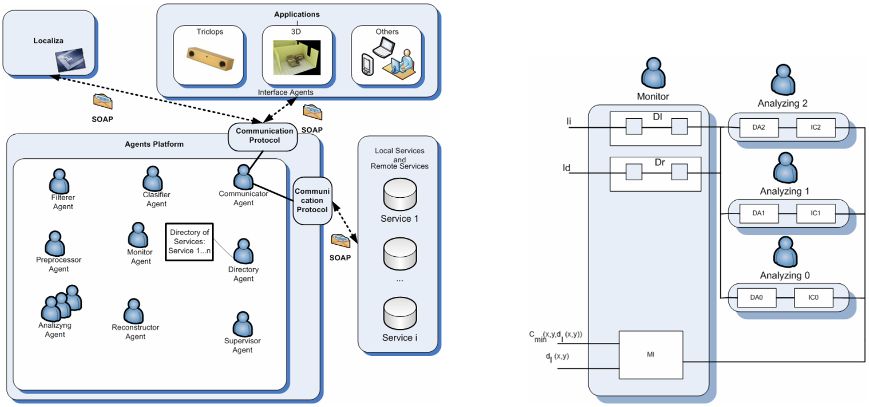


Fig. 2. (a) Diagram of the architecture. (b) Structure of the SAD algorithm

of the processes initially carried by the system are implemented to operate in parallel; only the correspondence algorithm. The system is comprised of a set of agents with defined roles that share information and services. The analysis of images supposes a complex process where each agent executes its task with the information available at each moment. Figure 2a illustrates the proposed multi-agent architecture.

The applications consist of each of the programs can be used for accessing the system functionalities. The applications are dynamic and adaptable to the context, and react differently to particular situations (for example, if it finds itself in one particular room or another, or if the user has some type of disability) and the type of service requested. The services represent the bulk of the system functionalities at the information processing, submission and retrieval levels. The system should count on a flexible directory of local and remote services that can be modified, added or eliminated dynamically and on demand.

The MAS function supports the coordination of agents that cooperate and share information that is necessary during the process of analysis. By breaking down the process of analysis described in the previous sections, the tasks generated are distributed among each of the system roles. The following roles were obtained: *Classifier*: in charge of receiving images, verifying the format, and changing it when necessary. *Filterer*: in charge of the basic filtering tasks. Coordinates filtering during the first phases either from the applications embedded in the cameras, or from the actual system of analysis. *Preprocessor*: in charge of extracting characteristics. It can use the extraction service deemed most convenient (Canny, etc.). *Monitor*: is in charge of controlling processes, assigns tasks to the other agents. Divides the images in blocks and assigns analysis tasks to the analyzing agents. The analyzing agents return the results and the *Monitor* sends the compressed information to the Reconstructor agent that is in charge of representing it three-dimensionally. *Interface*: this type of agent is designed to be embedded in user applications for direct communication with the platform agents. *Analyzing*: carries out the analysis of all correspondence. *Reconstructor*: gathers the information that has been analyzed and sends to the applications for 3D representation. *Communicator*: is responsible for the communications between applications and platform, and services and platform. It manages the requests for entry from the applications in order to be processed by the services. *Supervisor*: analyzes the structure and syntax of all of the messages entering and exiting the system, and supervises the correct functioning of the other system agents (periodically verify the state of all the agents registered in the architecture by sending ping messages). *Directory:* keep a list of all active services architecture.

This case has worked on a prototype, implementing an analytical component by following two of the techniques studied for the analysis of correspondence: the PMF algorithm, based on characteristics, and the pairing algorithm SAD, based on areas. In both cases we optimizes the use of the algorithms by taking advantage of the benefits that they provide to the platform agents, distributing tasks and parallelization, thus reducing the overall processing time.

PMF: The basis of the *PMF parallelization of the correspondence algorithm* was the restriction of the “correspondence between lines”. Based on this premise, the correspondence algorithm was executed for independent processing units, using only the two blocks of data contained in the same “line number” for each image as a parameter. After the images were reduced to their basic features, they were divided into 20 blocks of 24 lines each (as the images used were 640*480). The length of each block is based on the size of the rows in the image to be processes. It is important to mention that we forced the formation of the blocks so that the 24 lines that mage up each block would not be listed sequentially in the image. That is, they were selected so that there are 20 lines of separation between them. An example of the contents of one block would be: A={0,20,40,60,80,100,120,140,160,180,200,220,240,260,280,300, 320,340,360,380,400,420,440,460} and continuing likewise for the remaining blocks. This convention was adopted with the intention of uniforming the workload distribution among the agents, since there are usually certain areas in the image with a greater concentration of information.

Once the blocks have been formed, the Monitor agent sends them for parallel processing in independent agents (Analyzing). When the last agent finishes its process, the information is compressed by the Monitor in order to create a 3D reconstruction.

SAD: The stereo calculation is made with the Triclops library[20], which defaults to a pairing algorithm based on the Sum of Absolute Differences (SAD). As it is the method used in the libraries provided by Point Grey [15], we also implemented it, along with a proposal for optimizing the algorithm via the parallelization of the tasks by the algorithm.

The algorithm used should be broken down into a series of sub-functions, which perform sequentially in the dataflow. These sub-functions are assigned to the Analyzing agents by the Monitor agent who is in charge of identifying the sub-functions that must be carried out in the correspondence analysis, and assigning the tasks to each of the Analyzing agents. The structure of the algorithm is shown in figure 2b. In order to have access to the neighboring pixels in the correlation window, once they have been scanned, the FIFO (First In, First Out) system is used in the Monitor agent for the left and right images (Il, Ir), represented as Dl and Dr. The absolute difference between the pixels in the right and left images is calculated in the modules DAl, where l = 1, 2, … dmax, and dmax is the maximum displacement in the correlation window. The correlation index is calculated in modules ICi. The calculation of the absolute difference and that of the correlation index are performed in parallel by the Analyzing agents. The disparity dl(x, y) is determined by comparing all of the correlation indexes that correspond to the dmax displacements. The MI module determines the minimum correlation index and provides the disparity dl(x, y) as well as the minimum correlation index Cmin(x, y, dl(x, y)), which can be used to calculate a confidence index for the pairing.

5 Results and Conclusions

The proposed agent-based architecture allows us to automate our analysis and optimize its performance. The use of agents provides a great deal of flexibility since they can help move the code to places where immediate action is required and answers can be obtained in execution time, autonomously and without interrupting services. Additionally, we reduce the need to program multiple tasks since it is only necessary to specify the overall objectives for the agents to cooperate and work together to achieve the stated objectives. The stereoscopic vision algorithms are implements in the architecture, allowing tasks to be carried out in parallel, using each service as an independent processing unit.

Initially, the algorithm we chose to optimize was the correspondence algorithm, since computationally it is the most expensive. We defined two proposals for the parallelization of correspondence techniques: the PMF algorithm, based on characteristics, and the SAD algorithm, based on areas. The "Analyzing" agent prototype was implemented because of its ability to carry out this type of analysis.

The first 5 images in figure 3 show the processing of a sample PPM (*Portable Pixmap Format*) image composed of three stereo images.

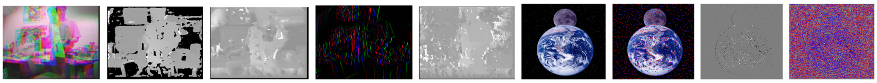


Fig. 3. Processed image results. From left to right: original image; disparity map using edge detection; disparity map obtained without using edge detection; use of the Sobel method on the first sample image; depth map obtained using Sobel (default mechanisms used in Triclops); original sample stereo image right; extraction of characteristics by number of lines n on the original image; edge detection and extraction of characteristics by number of lines m>n.

We can use the information extracted from the disparity map as values relative to the depth of the objects. The darkest values represent the farthest objects on the image, while the lightest values represent the closest objects. Preliminary results showed that the edge detection is one step that, while it may not be necessary, is very useful since it helps us with environments in which lighting conditions change notably. For this reason we decided to expand the prototype by adding more functionalities. In this case, we developed the Sobel [19] and Canny [5] edge detection algorithms. We can see how the disparity map reflects the depth of objects much better than the original image, even though it uses a gray-scale.

The next step was to develop the following correspondence analysis technique based on PMF characteristics. The last 4 images to the right in figure 4 illustrate the sequence of the results obtained.

In order to carry out an objective comparison, the algorithms were applied without enhancements. In order to perfect the results, we developed filters and masks that can adapt to the characteristics of the image (for example, converting JPG stereo images to PPM, and converting to color to gray scale), although the results are not presented in this paper. Additionally, we proposed an improvement in both algorithms using an agent architecture for their parallelization.

References

1. Angulo, C., Tellez, R.: Distributed Intelligence for smart home appliances. Red Española de Minería de Datos (2004)
2. Bajo, J., Corchado, J.M., de Paz, Y., de Paz, J.M., Rodríguez, S., Martín, Q., Abraham, A.: SHOMAS: Intelligent Guidance and Suggestions in Shopping Centres. Applied Soft Computing 9(2), 851–862 (2009)

3. Bajo, J., de Paz, J.F., de Paz, Y., Corchado, J.M.: Integrating Case-based Planning and RPTW Neural Networks to Construct an Intelligent Environment for Health Care. Expert Systems with Applications, part 2 36(3), 5844–5858 (2009)
4. Brenner, W., Wittig, H., Zarnekow, R.: Intelligent software agents: Foundations and applications. Springer, Secaucus, NJ, USA (1998)
5. Canny, J.: A computational approach to edge detection. IEEE Trans Pattern Analysis and Machine Intelligence 8(6), 679–698 (1986)
6. Corchado, J.M., Bajo, J., Abraham, A.: GERAmI: Improving the delivery of health care. IEEE Intelligent Systems 23(2), 19–25 (2008)
7. Corchado, J.M., Bajo, J., De Paz, Y., Tapia, D.I.: Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care. Decision Support Systems. Elsevier Science, Amsterdam (2006)
8. Corchado, J.M., Glez-Bedia, M., de Paz, Y., Bajo, J., de Paz, J.F.: Replanning mechanism for deliberative agents in dynamic changing environments. Computational Intelligence 24(2), 77–107 (2008)
9. Dhond, U.R., Aggarwal, J.K.: Structure From Stereo - A Review. IEEE Trans. on Systems, Man. and Cybernetics 19(6) (November/December 1989)
10. López-Valles, et al.: Revista Iberoamericana de Inteligencia Artificial 9(27), 35–62 (2005), ISSN: 1137-3601
11. Marr, D., Poggio, T.: A computational theory of human stereo vision. Proc. R. Soc. Lond., 301–328 (1979)
12. Marr, D., Hildreth, E.C.: Theory of edge detection. Proc. Roy. Soc. London B-207, 187–217 (1980)
13. Pearson, Don: Image Processing. McGrawHill, Great Britain (1991)
14. Pecora, F., Cesta, A.: Dcop for smart homes: A case study. Computational Intelligence 23(4), 395–419 (2007)
15. Point Grey Research Inc. (2008), `http://www.ptgrey.com/`
16. Pollard, S.B., Mayhew, J.E.W., Frisby, J.P.: A stereo correspondence algorithm using a disparity gradient constraint. Perception 14, 445–470 (1985)
17. Rao, A.S., Georgeff, M.P.: BDI agents: from theory to practice. In: Proceedings of ICMAS 1995, San Francisco, CA, USA (1995)
18. Rubio De Lemus, P.: Aplicacion De Una Metodologia De Evaluacion De Sistemas De Emparejamiento En Vision Tridimensional. Psicothema ISSN edición en papel: 0214-9915 5(1), 135–159 (1993)
19. Sobel, I., Feldman, G.: A 3x3 Isotropic Gradient Operator for Image Processing. In: Presentado en la conferencia Stanford Artificial Project (1968)
20. Triclops StereoVision System Manual Version 3.1. User guide and command reference. Point Grey Research Inc. (2003)

Participatory EHPR: A Watermarking Solution

David Lowe and B.R. Matam

Aston University, Aston Triangle, Birmingham B4 7ET, UK
{d.lowe,matambr}@aston.ac.uk

Abstract. The Electronic Patient Health Record (EPHR), as the basis for future ehealthcare systems, suffers from public suspicion and doubt regarding its security and exploitation.

In this paper a mechanism permitting patient empowerment to increase public acceptance of the EPHR is introduced, which also has the benefits of making its use more secure. We argue that the use of steganography (information hiding) to embed distributed personal patient details within the patient's medical data increases the personalised and participatory aspects. It also empowers the patient to have more control over the use of their own personal data. We discuss a hierarchical watermarking scheme for the embedding of multiple watermarks to provide different levels of security under patient control.

1 Patient Empowerment and the EPHR

This paper considers a technical solution to problem of patient empowerment over their own electronic patient health record, thus allowing for personalisation and participatory aspects automatically.

In February 2007 the *ARTICLE 29 Data Protection Working Party* of the European Union, published a 'Working Document on the processing of personal data relating to health in electronic patient health records (EPHR) [1]. The aim was to provide much needed guidance on the interpretation of the applicable data protection legal framework for EPHR systems, and was in support of EU Directive `95/46/EC` on the protection of individuals with regard to the processing of personal data and on the free movement of such data [2]. Essentially, this Working Document [1] has now created a new baseline for all EPHR systems. However, this document has not been without controversy. In response to this important working document, the Patient and Citizens Task Force of EHTEL (European Health Telematics Association), produced a response [3] which criticised its lack of appropriate recognition of patient rights versus legal positioning. Amongst their recommendations are: (emphasis added):

- The patient has the right to expect that his or her **privacy is respected and protected** and that the EHR is handled with due regard to professional duties of confidentiality
- **The patient should be considered to be the master of the Electronic Health Record**

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1264–1271, 2009.
© Springer-Verlag Berlin Heidelberg 2009

- **Technical innovations should be explored to handle specific situations where the patient may wish to have information withheld from certain organisations or individuals.**
- Managers and administrative staff should only have access to personal medical information on an exception basis and only after they have signed an appropriate confidentiality agreement.
- **The provision of a log-book showing who has accessed the EHR and what actions they have taken is considered to be an essential safeguard.**
- Every eHealth system should have an associated security policy that clearly describes how the confidentiality, integrity and availability of records will be preserved. This should be freely available for patients to access. System administrators should not have the right to access personal medical data.

These concerns imply a change in emphasis towards a more participatory role of the patient and empowerment of the individual over usage rights of their medical information.

Participatory healthcare systems can only succeed when the patient is empowered with control of his personalised EPHR. Confidence in the creation and utilisation of the EPHR therefore is central for a healthcare system. Current security protocols do not address these societal concerns, and the lack of data standards prevents global solutions.

2 A Steganographic Proposal

In this paper we further investigate a more technical solution based on the information-hiding science of steganography [4]. The patient's digital medical records (such as image scans or time series EEG traces) are regarded as 'covertext' data which we denote as $\mathbf{c}$ in which secret 'messages' denoted by $\mathbf{m}$ can be invisibly embedded [5]. These messages could be personal information that the patient wants to remain private, or a special access code known only to the patient, needed to unlock the private information. We can now hide multiple watermarks of different characteristics into a single covertext. So there could be several messages of differing degrees of sensitivity depending on who needs to see them. Controlling this access by digital means would allow the patient to authorise who has access to the personal information in the EPHR, whilst allowing the raw biomedical data still to be accessed by clinician or researcher. This leads to participation from patients in their active personal healthcare management.

In the biomedical domain a private watermark has to be robust to attacks, must be imperceptible (not cause visible distortion to the cover) and must also be capable of carrying sufficient information (rate of information). These three properties of a watermark form a trade-off against each other and therefore need to be calculated based on the application.

Watermark data inserted into low frequencies is more robust to distortions such as filtering, lossy compression and hence can be be used for watermarks requiring a high security. Watermark data inserted into middle and high frequencies is typically less robust to low-pass filtering, lossy compression and hence useful to embed watermarks used for tamper detection. Since the advantages and disadvantages of low and middle-to-high frequency watermarks are complementary, embedding multiple watermarks (namely, one in lower frequencies and the other in higher frequencies) would result in a hierarchical multiple watermarking scheme providing multiple applications [6].

In this paper we take a biomedical time series as an example covertext $\mathbf{c}$ and show how multiple hidden messages, or watermarks $\mathbf{m}_i$ can be embedded and recovered simultaneously. We compare two candidate approaches to this problem: The discrete wavelet transform approach (DWT), an example of an orthogonal transform of the covertext, and the independent component analysis approach (ICA), a non-orthogonal transform of teh biomedical covertext providing statistically independent sources in which the messages can be hidden.

3 Categorisation of Multiple Watermarks

We categorise the following 4 levels of hierarchy useful in the EPHR context:

- **Tamper Detection**, $\mathbf{m}_1$**:** A random binary string embedded randomly across the entire length of c to provide tamper detection. This has to be known by the decoder at the receiving end.
- **Diagnostic Information**, $\mathbf{m}_2$**:** Textual details including specification of medical tests, diagnosis results, doctors' notes. $\mathbf{m}_2$ may be used at the decoder for further processing of the medical data.
- **Data Provider Authentication**, $\mathbf{m}_3$**:** Clinician's name or identification to be used as record of source identification.
- **Patient Privacy**, $\mathbf{m}_4$**:** Patient's personal details embedded to prevent identification of the ownership of a medical record by unauthorised users of the patient data.

$\mathbf{m}_4$ representing personal details requires a high level of robustness to remain hidden/unaltered from malicious attacks, while $\mathbf{m}_1$ is used to estimate the level of attack, and hence needs to be fragile and be destroyed if the cover work $\mathbf{c}$ undergoes any transformation. Hence the watermark characteristics list a hierarchy from fragile to robust watermarking.

4 Data and Experiment

We validate the above ideas by the results of an information hiding experiment on low dimensional time series biomedical data.

The following procedure has been tested in the experiments:

1 An EEG signal of 100s, $\mathbf{c}$ is transformed using either DWT or ICA, $\mathcal{T}$.
2 Messages 1 to 4 are embedded in the selected transform domain components of $\mathbf{c}$, $\mathcal{C}$.

3 The signal with the embedded data $\tilde{\mathbf{c}}$ is reconstructed by applying the inverse of the applied transform, $\mathcal{T}^{-1}$ to the watermarked $\mathcal{C}$, $\tilde{\mathcal{C}}$.
4 As a compression attack, $\tilde{\mathbf{c}}$ is modelled as an autoregressive process of order six. The compression method is based on the AR process presented in [9].
5 The attacked watermarked EEG signal, $\hat{\mathbf{c}}$ is reconstructed at the receiver using the prediction coefficients polynomial A, the N initial coefficients of $\tilde{\mathbf{c}}$ and the error coefficients e_p.
6 Recovery of the embedded message for both the DWT and ICA methods is attempted. The bit error rate *dist* is calculated for each of the embedded messages to find the corresponding bit error rate.
7 Steps 2 to 6 are repeated a hundred times and the mean bit error rate is calculated for each message.

5 Deriving Multiple Channels to Embed Multiple Watermarks

In order to embed the four watermarks of differing security requirements, four signals with matching characteristics are derived from the one-dimensional EEG as follows.

5.1 DWT

The wavelet transform of a one-dimensional signal generates two wavelets d_a representing the scale coefficients of $\mathbf{c}$ and d_b the translation coefficients of $\mathbf{c}$. The two wavelets are derived using a suitable mother wavelet ψ. ψ here denotes the Haar wavelet. $\mathcal{C}_{a_{i+1}}$ and $\mathcal{C}_{b_{i+1}}$ are obtained by applying the DWT transform to $\mathcal{C}_{a_i}$. $\mathcal{C}_a$ represent the slow moving components of the signal (the low frequency content) while $\mathcal{C}_b$ represent the high frequency components. $\mathcal{C}_{b_i}$ for each increasing level of decomposition represent signals of decreasing frequency.

5.2 ICA

The one-dimensional EEG signal $\mathbf{c}$ in the ICA based watermarking method is transformed into an embedding data matrix $\mathbf{X}$ using delay embedding [10].

Delay embedding of $\mathbf{c}$ with delay of one sample between two successive delay vectors, $\mathbf{d}_i$ and $\mathbf{d}_{i+1}$, and an embedding window of size p is used: $\mathbf{d}_i = \mathbf{c}[i, \ldots, p + i - 1]$ where $i = 2, \ldots, N_{ov}$, and $\mathbf{X} = [\mathbf{d}_1; \mathbf{d}_2; \ldots; \mathbf{d}_{N_{ov}}]'$. The input to the ICA is $\mathbf{X}_{p \times N_{ov}}$ where $p = 83$ was chosen.

$$[\mathbf{S}, \mathbf{W}] \overset{ICA}{\longleftarrow} \mathbf{X}.$$

The rows of $\mathbf{S}$ represent the estimated independent sources and rows of $\mathbf{W}$ represent the independent components. Four sources $\mathbf{s}_{m_{\mathsf{i}}}$, $\mathbf{s}_{m_{\mathsf{ii}}}$, $\mathbf{s}_{m_{\mathsf{iii}}}$ and $\mathbf{s}_{m_{\mathsf{iv}}}$ are chosen from the l sources based on the spectrum of each source. A low frequency spectrum source contains a high information content of $\mathbf{c}$. The source with a high information content is more robust to compression compared to the source with a low information content. $\mathbf{s}_{m_1}$, $\mathbf{s}_{m_2}$, $\mathbf{s}_{m_3}$ and $\mathbf{s}_{m_4}$ are ordered in terms of increasing information content represented as $\mathbf{s}_{m_i}$ where i=[1, to 4]. For each watermark $\mathbf{m}_i$, $\mathbf{K}_i$ is the set of samples representing one of the four selected sources. $\mathbf{k}_i \in \mathbf{K}_i$ is randomly selected to embed the watermarks. The p sources were clustered into four different groups based on distances between the frequency spectra to distinguish the effect of embedding information in a particular source on the watermarked EEG.

The sources in each cluster have different spectra, hence the effect of a compression attack on a source in one cluster is different from the effect of compression on a source in a different cluster. A representative of each cluster was used to embed each of the four watermarks.

6 Watermark Generation, Embedding, Transmission and Decoding

6.1 Watermark Generation

Four different watermarks of varying levels of robustness requirement were generated as in [7]. The characteristics of each watermark in our case is different as we believe that the personal details of the patient require the highest security. Any attempt by an intruder with the best possible resources and knowledge of the watermarking technique must not result in the personal details being recovered correctly. The four watermarks are ordered in increasing order of privacy and security requirement with $\mathbf{m}_4$ requiring the highest security.

6.2 Watermark Embedding

Embedding of the watermarks is achieved based on the standard method known as quantisation index modulation [8]. The number of quantisation levels were chosen to depict the robustness of the DWT/ ICA method under a compression attack. We denote the watermark embedding of a message $\mathbf{m}_i$ in a source $\mathbf{s}_{m_i}$ using the samples $\mathbf{k}_i$ to give a modified, watermarked source $\tilde{\mathbf{s}}_{m_i}$ as

$$\mathcal{F}(\mathbf{m}_i, \mathbf{s}_{m_i}, \mathbf{k}_i) \rightarrow \tilde{\mathbf{s}}_{m_i}.$$

DWT. In the DWT based watermarking method $\mathcal{C}_{b_4}$ is more suited to carry the message requiring the highest level of privacy and protection $\mathbf{m}_4$.

$\mathbf{m}_i$ is embedded in the translation coefficients of $\mathcal{C}_{b_i}$, where i=[1 to 4]. $\mathbf{k}_i \in \mathbf{K}_i$ are chosen randomly to embed $\mathbf{m}_i$. The watermarked wavelet coefficients $\tilde{\mathcal{C}_{b_i}}$ are obtained. Applying the inverse of the DWT decomposition to the scale and translation coefficients, $\tilde{\mathbf{c}}$ is obtained.

ICA. The source containing signals of low frequency, was used to embed $\mathbf{m}_4$ which requires the highest robustness to attack. The source containing higher frequency distribution was used to embed the fragile $\mathbf{m}_1$.

The watermarked estimated sources matrix $\tilde{\mathbf{S}}$ contains the four watermarked sources $\tilde{\mathbf{s}}_{m_i}$ and the $l-4$ unwatermarked sources. Applying $\mathbf{W}^{-1}$ to $\tilde{\mathbf{S}}$, $\tilde{\mathbf{X}}$ is obtained,

$$\tilde{\mathbf{X}} = \mathbf{W}^{-1} * \tilde{\mathbf{S}}.$$

6.3 Watermark Transmission

The watermarked document $\tilde{\mathbf{c}}$ is corrupted during transmission due to intentional/common signal processing distortions η, resulting in an attacked watermarked document $\hat{\mathbf{c}}$.

$$\tilde{\mathbf{c}} + \eta \to \hat{\mathbf{c}}.$$

η in the experiments refers to a compression attack. The compression of $\tilde{\mathbf{c}}$ is achieved by representing the one-dimensional signal as an AR process of order 6. The ICA based method requires the exact separating matrix $\mathbf{W}$ used by the encoder at the decoder to decompose the received $\hat{\mathbf{c}}$ into estimated sources [11]. The matrix $\mathbf{W}$ acts as a key and prevents illegal estimation of the embedded information. This indicates that the level of security for the embedded watermarks is higher in the ICA compared to the DWT.

6.4 Watermark Decoding

$\hat{\mathbf{c}}$ is transformed using DWT or ICA to obtain the wavelet decomposition/ estimated sources respectively. By applying the nearest integer level decoding technique an estimate of the embedded message is obtained.

The inverse of the watermark insertion and watermark generation process is applied sequentially to $\hat{\mathbf{c}}$ to obtain estimates of the embedded messages, $\hat{\mathbf{m}}_i$.

$$\mathcal{T}(\hat{\mathbf{c}}) \to \hat{\mathcal{C}}.$$

$$\mathcal{F}^{-1}(\hat{\mathcal{C}}(\mathbf{k})) \to \hat{WM}.$$

$$\hat{WM} \to \hat{\mathbf{m}}.$$

The verification of the decoding process is conducted by calculating the Hamming distance, *dist* between the binary watermarks embedded in $\mathbf{c}$ and the estimated watermarks from $\hat{\mathbf{c}}$. $dist = \sum(WM \oplus \hat{WM})$. In the absence of any knowledge of $\mathbf{m}$ at the decoder a predetermined threshold value is used to determine the presence of $\hat{\mathbf{m}}$.

7 Results

The watermark reconstruction error is calculated by comparing the retrieved watermarks at the decoder to the original embedded watermarks.

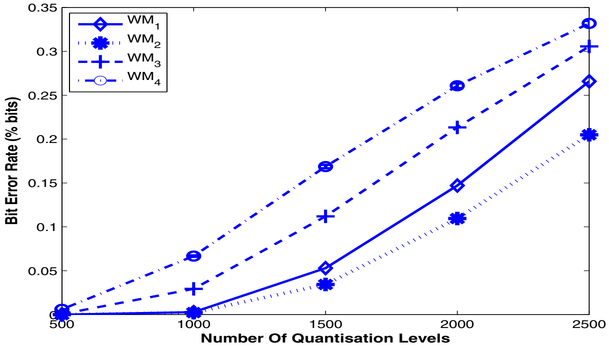


Fig. 1. Decoded error for each watermark using DWT

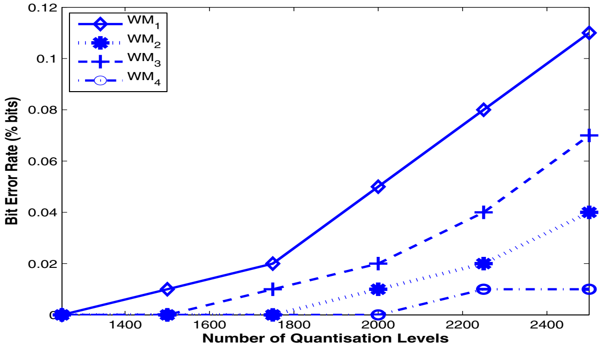


Fig. 2. Decoded error for each watermark using ICA

The signal ($\mathbf{c}$) to noise ($\mathcal{D}_{Emb}$) ratio of $\tilde{\mathbf{c}}$ using the ICA technique was 42.86dB and in the DWT method 48.79dB. This SNR was calculated for the largest value of δ required to decode the watermarks with zero error. The performance of both the ICA and DWT in terms of imperceptibility can be regarded to be acceptable.

Figures 1 and 2 depict the comparison of the performance of the DWT and ICA techniques for η. The value of $\mathcal{R}$ in both the methods of embedding information remaining equal totalling 5.62 bits per second of the EEG, the number of quantisation levels for both the DWT and the ICA being the same, the bit error rate for the ICA method is much lower compared to the DWT method.

The DWT based watermark embedding does not respond as designed against the LPC attack. If the attack is severe $\mathbf{c}$ will be compromised which renders it useless for diagnosis. As shown from our experiments, the ICA approach provides more control over security and distortion for a given data rate.

8 Conclusion

We have illustrated that hierarchical information-hiding approaches exist with differing levels of sensitivity. This mechanism can be used to permit patient-control over who has access rights to private parts of data. This is independent of other measures of data security such as encryption which could also be applied in tandem.

References

1. Working Document on the processing of personal data relating to health in electronic health records (EHR), written by the ARTICLE 29 Data Protection Working Party (adopted on February 15, 2007), `http://ec.europa.eu/justice_home/fsj/privacy/docs/wpdocs/2007/wp131_en.pdf`
2. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data OJ L 281 (November 23, 1995), `http://ec.europa.eu/justice_home/fsj/privacy/law/index_en.htm`
3. Patient and Citizens Task Force of EHTEL, Response to the Working Document on the Processing of Personal Data Relating to Health in Electronic Health Records as prepared by the European Data Protection Working Party (June 2007), `http://ec.europa.eu/justice_home/fsj/privacy/docs/health_records/others/ehtel_en.pdf`
4. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann Publishers, San Francisco (2002)
5. Kong, X.: Watermarking Medical Signals for Telemedicine. IEEE Trans. Inf. Tech. Biomedicine 5(3), 195–201 (2001)
6. Woo, C.S., Du, J., Pham, B.: Multiple Watermark Method for Privacy Control and Tamper Detection in Medical Images. In: Proc. APRS Workshop on Digital Image Computing (WDIC 2005), pp. 59–64 (2005)
7. Giakoumaki, A., Pavlopoulos, S., Koutsouris, D.: Multiple Image Watermarking Applied to Health Information Management. IEEE Trans. Inf. Tech. in Biomedicine 10(4), 722–732 (2006)
8. Chen, B., Wornell, G.W.: Quantization Index Modulation: a Class of Provably Good Methods for Digital Watermarking and Information Embedding. IEEE Trans. Inf. Theory 47(4), 1423–1443 (2001)
9. Antoniol, G., Tonella, P.: EEG Data Compression Techniques. IEEE Trans. Biomedical Engineering 44(2), 105–114 (1997)
10. Woon, W.L., Lowe, D.: Nonlinear Signal Processing for Noise Reduction of Unaveraged Single Channel MEG Data. In: Proc. Int. Conf. Artificial Neural Networks, pp. 650–657 (2001)
11. Matam, B.R., Lowe, D.: Steganography, BioPatterns and Independent Components. In: Proc. 7th Int. Conf. Mathematics in Signal Processing, pp. 206–209 (2006)

Bus Network Scheduling Problem: GRASP + EAs with PISA * Simulation

Ana C. Olivera[1,4], Mariano Frutos[1,2,4], Jessica A. Carballido[1,4],
Ignacio Ponzoni[1,3,4], and Nélida B. Brignole[1,3,4]

[1] Departamento de Ciencias e Ingeniería de la Computación
[2] Departamento de Ingeniería
[3] Planta Piloto Ingeniería Química (PLAPIQUI) Complejo CCT-UAT, CONICET
Camino la Carrindanga Km. 7, 8000 Bahía Blanca, Argentina
[4] Av. Alem 1253, B8000CPB Bahía Blanca, Argentina
`lidecc@cs.uns.edu.ar`

Abstract. In this work a memetic algorithm for the Bus Network Scheduling Problem (BNSP) is presented. The algorithm comprises two stages: the first one calculates the distance among all the pairs of bus stops, and the second one is a MOEA that uses a novel simulation procedure for the calculus of the fitness function. This simulation method was specially developed for the BNSP. The EA used for the second stage was selected between the IBEA, NSGA-II and SPEA2 by means of some PISA tools. As a result of this experimentation, the SPEA2 was preferred since it presents the more spread solution set.

Keywords: Memetic Evolutionary Algorithms, Bus-Network Scheduling Problem, Optimization, PISA.

1 Bus-Network Scheduling Problem

People travel everyday in the cities with the bus network system. The design of a bus network involves the satisfaction of the interests of the net's entities [4]: the **user**, who is a passenger of the buses, and the **operator** of the lines. These entities usually have different expectations for the bus network, which are generally confronted. The Bus Network Scheduling Problem (BNSP) deals with the task of finding a bus network, with the maximization of quality of the service and the maximization of the benefit for the transport operator [4]. The main challenging edges of this problem lie on its NP-complexity [3], economic and social interests, and technical difficulties. Furthermore, an extra hindrance is constituted by the need to consider temporal features in the model.

In this work, we have focused on the line-scheduling and route-design activities, and the entities, *user* and *operator,* are considered with the same level of importance. In essence, several stages were established for the memetic algorithm, by means of the combination of Multi-Objective Evolutionary Algorithms (MOEAs) based on the PISA platform [2], a Greedy Randomized Adaptive Search Procedures (GRASP) [14] and a Simulation tool, which was included in order to proportion representative time-related elements for the calculation of fitness values, which are necessary to carry off the dynamics of the real scenarios with precision.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1272–1279, 2009.
© Springer-Verlag Berlin Heidelberg 2009

This paper is structured in five sections. In section two, the equations that are used to model the entities that participate in the problem are introduced. Then, in the main section of the article, each stage of the memetic algorithm is presented. The fourth section contains the experimental studies based on an academic case of study, and an analysis of the results. Finally, the last section discusses the conclusions.

2 Entities of the BNSP

Due to its structural complexity, methods that have previously tackled the BNSP do not contemplate the randomness of real systems. The objectives of the problem, related to each of the entities, are modeled through equations, which will be used by the MOEA to guide the search. The equation associated to the **operator** is:

$$\sum_{L=1}^{M} FO_L = \sum_{L=1}^{M} \left(AF_L\ T_L - D_L\ K_L \right) \tag{1}$$

In the Equation 1, the rent for the transit network is represented with AF_L (the total client influx for line L) and T_L (the price for a journey in the line L). In contrast, the cost of the operator is relative to the total travel distance for each line L (D_L) and the unitary operative cost of operation per kilometer for the line L (K_L). FO_L corresponds to the economic benefit of the operator.

The equation associated to the **user** is defined as follows:

$$\sum_{L=1}^{M} FU_L = \sum_{L=1}^{M} \left[\sum_{i=1}^{N} \sum_{j=1}^{N} \left(\delta t^{A}_{ijL} + t^{J}_{ijL} + \eta t^{W}_{ijL} \right)\ VST_{ijL}\ V_{ijL} \right] \tag{2}$$

For a customer of line L, in the equation 2, t^{A}_{ijL} represents the access time (the time between the arrival of the bus and the payment of the journey), t^{J}_{ijL} is the journey time, and t^{W}_{ijL} characterizes the waiting time. VST_{ijL} denotes a subjective value for the time corresponding to each pair origin-destiny (i,j) that employs line L; δ and η are the relative weights between access time and waiting time with respect to the journey time. Finally, V_{ijL} is the number of journeys for each pair (i,j) that employs line L. In Equation 2, FU_L represents the cost related to transporting a client in line L.

From the formulation of these equations arises the need to model the attribute of time [5]. For this reason, in a later subsection this feature will be explained in the context of the calculation of the fitness function in the EA.

3 Memetic Multi-Objective Evolutionary Approach

The main stages of the algorithm comprise the initialization, which constitutes the estimation of the paths between each pair of bus stops and the corresponding distances; and the core, which yields the entire bus network. The memetic algorithm defines the routes and distances between any pair of bus stops by means of the GRASP method. Then, the MOEA performs the most important task assisted by the simulation tool (Figure 1).

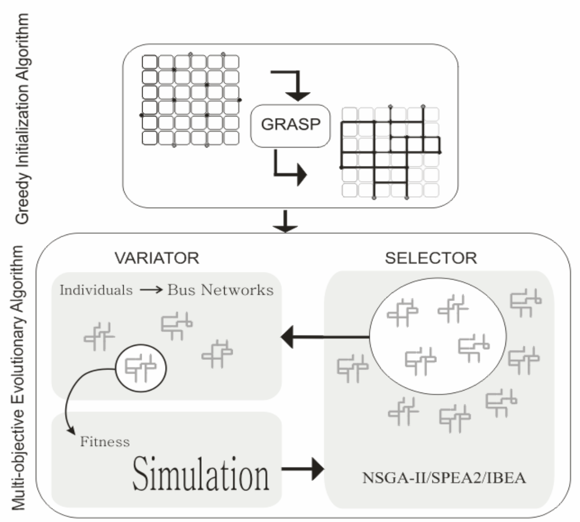

Greedy Initialization Algorithm
GRASP
Multi-objective Evolutionary Algorithm
VARIATOR
SELECTOR
Individuals → Bus Networks
Fitness
Simulation
NSGA-II/SPEA2/IBEA

Fig. 1. Memetic algorithm's layout

The MOEA stage was designed using the PISA platform [2], which is an interface for search algorithms that splits an optimization process into two modules. One module contains all the parts that are specific to the optimization problem, e.g., evaluation of solutions -simulation tool-, problem representation, and variation of solutions, and is called the *Variator*. The other module contains the parts of an optimization process which are independent of the problem, and it is called the *Selector*. For this work, we have designed a *Variator* specific for the BNSP, and we have combined it with the *Selectors* corresponding to the IBEA [16], NSGA-II [7] and SPEA2 [18] optimization algorithms. The reason for the selection of these MOEAs is that they are the most recommended evolutionary optimizers in the literature. In the following sections, we will describe the main features of the implemented *Variator*.

3.1 GRASP Stage

For the BNSP treatment, a classification of a node map was introduced. A node is any point in the map. In particular, **bus stops** are special nodes since it is important to know the distance between one bus stop and the others. When a line has the same bus stop in its route as another line, then this shared bus stop is called a **transfer point**. The difference between bus stops and transfer points is crucial for the understanding of the whole procedure. Greedy Randomized Adaptive Search Procedure (GRASP) is an exploratory local search method used to solve combinatorial optimization problems [14]. The GRASP constitutes the initialization stage and it works on two bus stops for a given graph that contains all the nodes. In the **constructive phase**, it generates a sub-optimal route of nodes between both bus stops, and it also yields the distance between them. The **local search procedure** takes the best route generated in the **constructive** GRASP step and examines the neighborhood of each node to find a better solution.

3.2 Multi-Objective Evolutionary Algorithm Stage

Several authors have shown that EAs succeed in obtaining sub-optimal solutions for NP-Hard problems [1, 8, 13]. In this work, a *Variator* for the BSNP was specifically implemented. It starts from a set of parameters, the distances between bus stops obtained through GRASP, and an initial population randomly generated. Each **individual** of the population represents a **bus network**. The EA evolves the population until several satisfactory bus networks are achieved in the last generation. The parameters are the following: number of bus stops with concentrated demand (B_I), information of each bus stop (position, time between arrivals), number of lines (M), fee for travelling with each line L (T_L), unitary cost of operation per kilometre for each line L (K_L), starting nodes B_S, return nodes B_F, and maximum capacity of buses C_{MAX}.

3.2.1 The Individuals

An individual represents a bus network by means of a **set of lines**. Each route of a line is modeled by an ordered list of integers, starting from an initial bus stop B_S and ending with a final bus stop B_F. Note that not all the integer strings constitute feasible solutions. Restrictions related to the feasibility of individuals are expressed according to the Set Theory. Consider that L_1, L_2,…, L_M are the lines represented by the individual *x*. A feasible solution for the BNSP should fulfill these constraints:

1. The solution should contain the same number of routes as the amount of lines (M) that were defined as input parameter of the algorithm (Equation 3).

$$x = \{L_1, \ldots, L_M\} \tag{3}$$

2. The number of transfer points should be consistent with the input parameter for the GA (Equation 4).

$$\forall L_i \exists L_j, i \neq j \Rightarrow L_i \cap L_j \neq \emptyset \tag{4}$$

3. All the bus stops should be present at least in one of the lines (Equation 5).

$$\{B_S\} \cup \{B_I\} \cup \{B_F\} = \{L_1\} \cup \{L_2\} \cup \ldots \cup \{L_M\} \tag{5}$$

3.2.2 The Genetic Operators

The genetics operators implemented in the *Variator* have a direct relationship with the BNSP and the individual's representation.

Crossover. A variation of the two-point crossover was adopted for this problem. By using the constraints defined in equations 3-5, if the crossover produces unfeasible children, they are discarded and new cut-points are selected.

Mutation. Two cases were implemented: edge and node mutation. Edge mutation randomly chooses two edges from two different routes and inverts them. There are two alternatives for node mutation: to insert the randomly selected node into another route or to introduce it into the same route. In both cases, if the resulting individual is unfeasible, it is discarded and the original individual is mutated again.

3.2.3 Fitness

Taking into account that FO_L and FU_L are two objectives of this problem, there is not only one solution for the problem. Moreover, both objectives are equally important. For this reason, in this paper we are proposing a multi-objective EA with the *Pareto frontier* concept to tackle this problem [9]. In this context, the FO_L objective in Equation 1 is redefined in order to minimize this goal, and Equation 6 shows the new formulation of this objective.

$$1 \Big/ 1 + \sum_{L=1}^{M} FO_L \tag{6}$$

Simulation. For the formal BNSP treatment, it was necessary to apply simulation techniques related to queue theory and access to resources. The static structure of a bus network is basically composed of different routes, the operator's fleets, the network users, and the transfer points. A transfer point is present in more than one route. This case may be associated to a user-change-of-unit center that connects the transport network and enables the user to access every point. It should be considered that a bus moves from the initial line stop to the final one, and then it travels back towards the initial point. During the simulation, each entity is associated to some information that should be obtained and that is later used to calculate the fitness. For each user, we are interested in its own waiting time (t_{WijL}), trip time (t_{JijL}), and access time (t_{AijL}). Each bus stop keeps two lists: a list of clients who are queuing, i.e. waiting to get into a mobile, and a list of the users that will arrive to that stop. It also stores information on the number of users that arrived to this center (V_{ijL}). The line's attributes include a list of the nodes the line goes through, its fleet and the total influx of trips (AF_L). The simulation time progresses in a discrete and synchronic way. For our problem instance, the planned event that advances the simulation clock is the arrival of a mobile to a node. Before beginning the simulation effectively for a mobile's working-day, the arrival of all the clients to the respective transfer centers is generated, and they are put on the customers' list that will potentially arrive to the stop. The simulation begins by generating each mobile's arrival to its route's initial point. The clock advances to the first arrival of the first transport of the fleet. Besides, the transport arrival to the next node is planned considering the present clock and the distance that has been traveled, which was obtained through GRASP. In this way, the clock moves forward until the simulation ends.

3.3 Working Out the Challenging Puzzle

In this section, we will explain how all the aforementioned pieces are assembled. Firstly, the GRASP calculates the different paths and distances between every pair of bus stops. Then, based on the "map" yielded by the GRASP stage, the *Variator* builds a random population of feasible bus networks. Later, in order to evaluate the fitness of the individuals in the population, it is necessary to calculate the value of each one of the objectives, FO_L and FU_L to be short. For this aim, the algorithm simulates a day of work, and returns the values for the variables that are necessary for the calculation of the eqs. 2 and 6. Afterwards, the *Selector* (IBEA, SPEA2 or NSGAII) chooses the parents for the next population. The parents are crossed and the children are mutated. This is performed in every generation until a given generation number is reached.

4 Experimental Results

In this section, an academic case study is presented, and the results achieved by the memetic algorithm are reported and analyzed. The experiments were carried out on the map of a hypothetical city that is represented with 100 nodes, 2 initial stops (B_S), 6 intermediate stops (B_I), and 2 final stops (B_F). The goal is to design the routes for two bus lines L_0 and L_1. There is room for 25 people in each bus, which is the maximum amount to be admitted. For this experiment, we modeled only one time frequency (the morning) that is set at the beginning of the run with 20 minutes.

For the EA stage of the memetic algorithm, IBEA, NSGA-II and SPEA2 have been evaluated with 50 runs and 100 generations over the academic case. For the experiment, α=100, μ=50 and λ=50. These values were selected from a few preliminary runs. In the case of IBEA algorithm we chose the *Additive Epsilon Indicator* and the rest of the parameters were set to the default values.

The metrics applied in the evaluation of the MOEAs are *Dominace Ranking* [12], the *Hypervolumen Indicator* I^{-}_{H} [15], the *multiplicative version of the Unary Epsilon Indicator* I^{1}_{E} [17] and the *R2 Indicator* I^{1}_{R2} [11]. If a significant difference can be demonstrated using the dominance ranking, the only purpose of the Quality Indicators is to characterize further differences in the approximation sets. On the other hand, if we cannot establish significant differences by means of the dominance rank, then the Quality Indicators can help us in the decision of which one of the optimizers is better.

For the parameter of the indicators, the default values of PISA's parameters were maintained, since the objectives are automatically normalized to the interval [1..2]. We have considered the algorithms in ordered pairs, and we have applied the one-tailed Mann-Witney [6] rank sum. None of the results is statistically significant regarding an overall significance level α=0.05. This shows that none of the MOEAs generates better approximation sets respect to the others. Next, unary quality indicators were applied using the normalized approximation sets and the reference set. The indicators were *Unary Epsilon Indicator* I^{1}_{E}, *Unary Hypervolume Indicator* I_H and *R2 Indicator* I^{1}_{R2}. The distribution of the resulting transformed approximation sets can be graphically visualized using conventional box plots (see Fig. 2).

A non-parametric statistical test was applied to the transformed approximation sets in order to obtain valid conclusions about the quality of the optimization methods. In this paper, Fisher test [16] has been chosen. The Table 1 shows that there are no

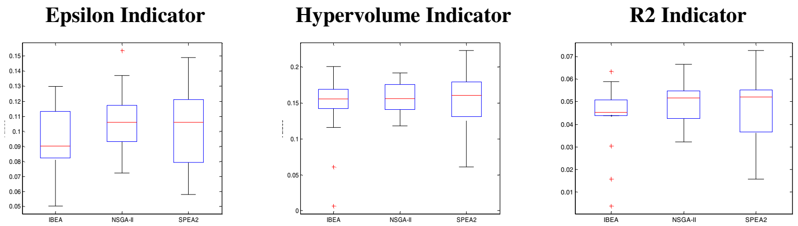


Fig. 2. Box plots for different unary indicators

Table 1. Fisher test over the Quality Indicators

Multiplicaticative Epsilon Indicator

	IBEA	NSGA-II	SPEA
IBEA	-	0.05998	0.08998
NSGA-II	0.93792	-	0.52518
SPEA2	0.91210	0.46918	-

Hipervolumen Indicator

	IBEA	NSGA-II	SPEA
IBEA	-	0.14782	0.24546
NSGA-II	0.85175	-	0.62742
SPEA2	0.75804	0.36888	-

R2 Indicator

	IBEA	NSGA-II	SPEA
IBEA	-	0.09732	0.16856
NSGA-II	0.90272	-	0.57412
SPEA2	0.83356	0.42332	-

significant differences between the results for the algorithms on the BNSP. Considering that several tests were applied and that none of them could establish a significant difference between the MOEAs, the last chance is to evaluate them considering the attainment surface plots generated by the MOEAs in each run.

The Figure 3 shows that there are not considerable differences between the MOEAs. However, the graphic shows that SPEA2 has the most convenient sets of solutions and the solution space in each run is largely more spread than the others. Due to that, decision makers have the last opinion on the most convenient MOEA for the resolution of BNSP. Therefore, since the SPEA2 exhibits the most spread solution set, this EA is preferred from the others.

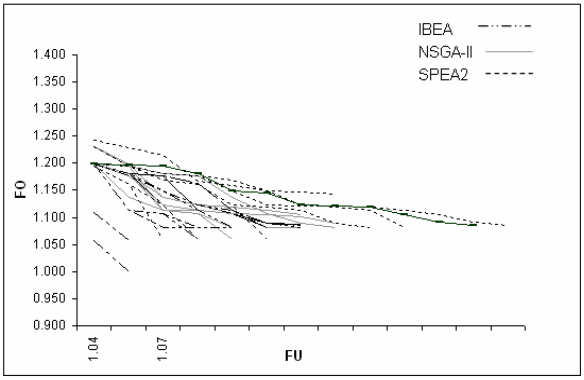


Fig. 3. Attainment surface plots for the first eight approximation sets generated by IBEA, NSGA-II and SPEA2

5 Conclusions

In this work we have presented the main features of a memetic stochastic technique built to solve the bus network scheduling problem. The technique gathers a map initialization tool implemented by a GRASP, with a route and scheduling method based on a MOEA with novel simulation features for the treatment of some dynamic aspects of the problem.

Several tests were carried out with the objective of comparing and choosing between three MOEAs, by means of the analysis tools provided by the PISA platform. SPEA2 was selected for the second stage of the memetic algorithm, since it presents the most spread set of solutions. In this manner, a complete memetic strategy for the bus network scheduling of real congested urban areas has been introduced.

Acknowledgments. Authors acknowledge the ANPCyT from Argentina, for Grant N°11-1652 and SeCyT (UNS) for Grants PGI 24/ZN15 and PGI 24/N019.

References

1. Bielli, M., Caramia, M., Carotenuto, P.: Genetic Algorithms in Bus Network Optimization. Transportation Research Part C: Emerging Technologies 10(1), 19–34 (2002)
2. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA - A Platform and Programming Language Independent Interface for Search Algorithms. In: Proceeding of Evolutionary Multi-Criterion Optimization, pp. 494–508 (2003)
3. Ceder, A., Wilson, N.H.M.: Bus Network Design. Transportation Research 20(4), 331–344 (1986)
4. Ceder, A., Israeli, Y.: User and Operator Perspectives in Transit Network Design. Transportation Research Record 1623, 3–7 (1998)
5. Chang, S.K., Schonfeld, P.M.: Analytical Optimization of Parallel Bus Routes with Time Dependent and Elastic Demand. Transportation Studies Center Working Paper 89-22, University of Maryland, College Park (1989)
6. Conover, W.: Practical Nonparametric Statistics. John Wiley & Sons, New York (1999)
7. Deb, K., Agarwal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
8. Desaulniers, G., Hickman, M.: Public transit. Handbooks in Operations Research and Management Science. In: Laporte, G., Barnhart, C. (eds.) Transportation, vol. 14, pp. 69–127. Elsevier, Amsterdam (2007)
9. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation. Discussion and Generalization, Forrest, 416-423 (1993)
10. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Massachusetts (1989)
11. Hansen, M., Jaszkiewicz, A.: Evaluating the quality of approximations to the non-dominated set. Technical University of Denmark (1998)
12. Knowles, J., Thiele, L., Zitzler, E.: A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. TIK Computer Engineering and Networks Laboratory (2005)
13. Pattnaik, S.B., Mohan, S., Tom, V.M.: Urban Bus Transit Route Network Design Using Genetic Algorithm. Journal of Transportation Engineering 124(4), 368–375 (1998)
14. Pitsoulis, L.S., Resende, M.G.C.: Greedy Randomized Adaptive Search Procedures. In: Pardalos, P.M., Resende, M.G.C. (eds.) Handbook of Applied Optimization. Oxford University Press, Oxford (2001)
15. Zhao, F.: Simulated Annealing–Genetic Algorithm for Transit Network Optimization. ASCE Journal of Computing in Civil Engineering 20(1), 57–68 (2006)
16. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
17. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. Evol. Comput. 3(4), 257–271 (1999)
18. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, T., Periaux, P., Fogarty (eds.) Evolutionary Methods for Design, Optimisations and Control, pp. 19–26 (2002)

Wine Classification with Gas Sensors Combined with Independent Component Analysis and Neural Networks

Jesús Lozano, Antonio García, Carlos J. García, Fernándo Alvarez,
and Ramón Gallardo

Pattern Classification and Image Analysis Group. University of Extremadura,
Elvas av., 06006 Badajoz (Spain)
`jesus@capi.unex.es`

Abstract. The aim of this work is to demonstrate the alternative of using Independent Component Analysis (ICA) as a dimensionality reduction technique combined with Artificial Neural Networks (ANNs) for wine classification in an electronic nose. ICA has been used to reduce the dimension of the data in order to show in two variables the discrimination capability of the gas sensors array and as a preprocessing tool for further analysis with ANNs for classification purposes.

Keywords: independent component analysis, neural networks, electronic nose, data processing.

1 Introduction

The quality of wine is influenced by different sensory characteristics. The most important is aroma [1]. This attribute has a 70 % weight in sensory panels with respect to texture and taste. Electronic nose (e-nose) technology has emerged as a possibility for aroma profile analysis. The e-nose basically consists of an array of gas sensors with different selectivity, a signal collecting unit, and pattern recognition software [2]. Pattern analysis constitutes a critical building block in the development of gas sensor instruments capable of detecting, identifying, and measuring volatile compounds, a technology that has been proposed as an artificial substitute of the human olfactory system. The successful design of a pattern analysis system for machine olfaction requires a careful consideration of the various issues involved in processing multivariate data (see fig. 1): signal preprocessing and feature extraction, dimensionality reduction, classification, regression or clustering and validation [2][3]. A variety of Pattern Recognition and Machine Learning techniques is available for each module and only preprocessing module is sensor dependent.

The initial block in the figure 1 represents the e-nose hardware. The data processing starts after the sensor signals have been acquired and stored into the computer. The first computational stage, called signal preprocessing, serves various purposes; including compensating for sensor drift, extracting descriptive

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1280–1287, 2009.
© Springer-Verlag Berlin Heidelberg 2009

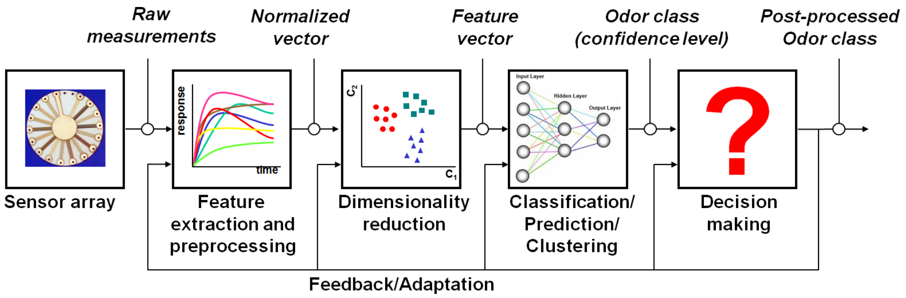


Fig. 1. Stages of typical signal processing in an electronic nose

parameters from the sensor array response and preparing the feature vector for further processing. A dimensionality reduction stage projects this initial feature vector onto a lower dimensional space in order to avoid problems associated with high-dimensional, sparse datasets. The resulting low-dimensional feature vector is then used to solve a given prediction problem, typically classification, regression or clustering.

Classification tasks address the problem identifying an unknown sample as one from a set previously learned odorant. In regression tasks, the goal is to predict a set of properties (e.g., concentration, quality) for an analyte, typically a complex mixture. Finally, in clustering tasks the goal is to learn the structural relationship among different odorants. A final step, sometimes overlooked, is the selection of models and parameter setting and the estimation of true rates for a trained model by means of validation techniques.

2 Material and Methods

A home-made and home-developed electronic nose was used to perform the measurements of wine samples. This section describes the e-nose and the data processing system.

2.1 Wine Samples

A total of 26 commercial wines (19 red and 7 white) from different regions were tasted by a sensory panel and analyzed by the electronic nose for the experiment. A list of wine samples is shown in table 1. Wine samples were acquired in specialized shops and supplied by collaborator wine cellars.

2.2 Electronic Nose

An array of 16 thin film tin oxide sensors was prepared in our laboratory by RF sputtering onto alumina and doped with chromium and indium. Sensor thickness varied between 200 and 800 nm. The array was placed in a $24cm^3$ stainless steel

Table 1. Wine samples measured by the e-nose and tasted by the sensory panel

Num.	Wine	Type of wine
W1	PIOZ (Guadalajara, Spain)	Young white
W2	Viña Esmeralda (Miguel Torres, Spain)	Young white
W3	Castillo de san Diego (B. Barbadillo, Spain)	Young white
W4	Marqués de Riscal (Rioja, Spain)	Young white
W5	Viñas del Vero (Somontano, Spain)	Young white
W6	Don Alvaro de Luna Albillo 2001 (Madrid, Spain)	Young white
W7	Albariño 2003 (B. Martín Códax, Spain)	Young white
R1	Puerta de Alcalá 2003 (Vinos Jeromín, Madrid, Spain)	Young red
R2	Cillar de Silos (Bodegas de la Villa, Spain)	Young red
R3	Castillo de Olite (Bodegas Artesanas, Spain)	Young red
R4	Cabernet Sauvignon Encin 2003 (Madrid, Spain)	Young red
R5	Ruchel Mencía 2002 (B. Majlu, Spain)	Young red
R6	Monasterio de las Viñas (Cariñena, Spain)	Young red
R7	Castillo de Aguarón (Cariñena, Spain)	Young red
R8	Don Luciano (B. Garcia Carrión, Spain)	Young red
R9	Sliven Merlot 2002 (Bulgary)	Young red
M1	Pablo Morate American Oak Barrel (Madrid, Spain)	Oak Barrel aged
M2	Pablo Morate French Oak Barrel (Madrid, Spain)	Oak Barrel aged
M3	Carlos Gosalbez American Oak Barrel 2002 (Madrid, Spain)	Oak Barrel aged
M4	Carlos Gosalbez French Oak Barrel 2002 (Madrid, Spain)	Oak Barrel aged
M5	Andrés Morate American Oak Barrel (Madrid, Spain)	Oak Barrel aged
M6	Andrés Morate French Oak Barrel (Madrid, Spain)	Oak Barrel aged
M7	Luis Saavedra American Oak Barrel 2002 (Madrid, Spain)	Oak Barrel aged
M8	Luis Saavedra French Oak Barrel 2002 (Madrid, Spain)	Oak Barrel aged
M9	Carlos Gosalbez American Oak Barrel 2003 (Madrid, Spain)	Oak Barrel aged
M10	Carlos Gosalbez French Oak Barrel 2003 (Madrid, Spain)	Oak Barrel aged

cell with a heater and a thermocouple. The operating temperature of sensors is controlled to 250°C with a PID temperature controller. Two sampling methods were available for analysis: purge and trap and static headspace followed by a dynamic injection [4][5]. The resistance of the sensors was measured with a Keithley 2700 71/2 digits digital multimeter (DMM) with a 40-channels multiplexer connected to the personal computer through a GPIB interface. More details of the system are shown in fig.2 and [6].

E-nose responses. Figure 3 shows the typical transient responses of four chemoresistive sensors of the array, operating at 250°C, exposed towards the headspace of one of the wine samples (PIOZ - W1). The response of the sensors corresponds to several pulses of 20 minutes of exposition to the tested wine flavour followed by a pure nitrogen purge for 40 minutes. These responses were stored in hard disk and processed later by the data processing system for prediction and classification purposes.

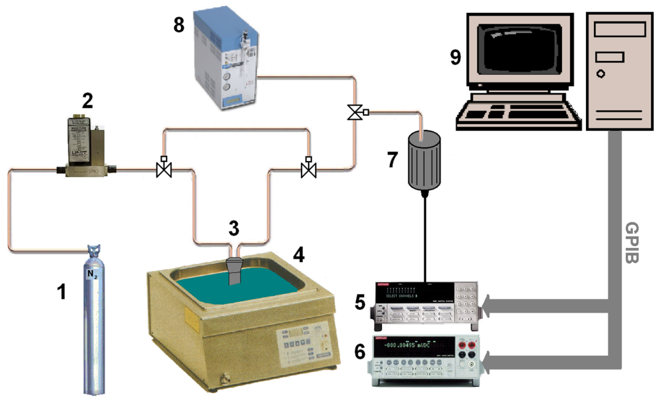


Fig. 2. Measurement setup: 1. Nitrogen bottle, 2. Mass Flowmeter Controller, 3. Dreschell bottle with sample, 4. Thermostatic bath, 5. Digital multimeter, 6. Multiplexer, 7. Sensors cell, 8. Tekmar 3100 purge and trap. 9. PC.

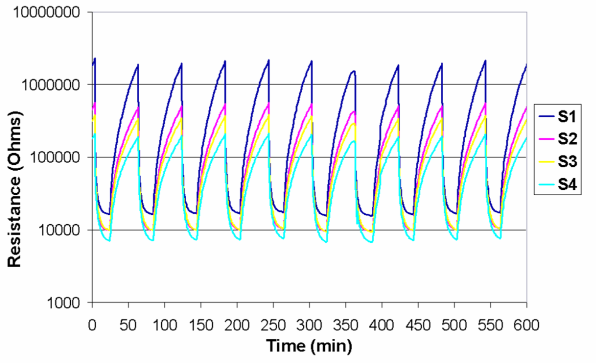
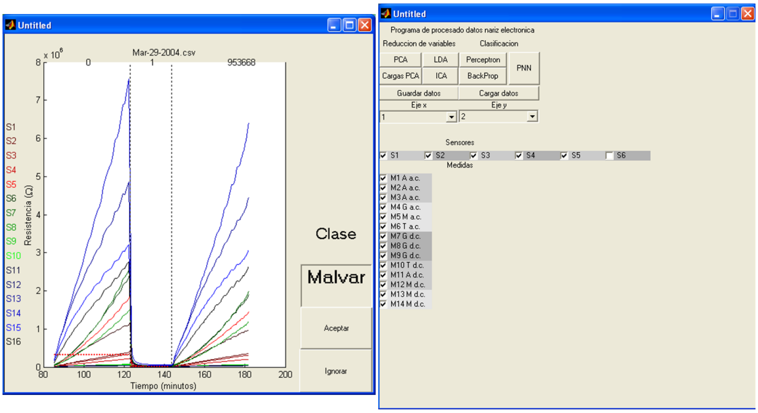


Fig. 3. Typical transient response of four sensors of the array

2.3 Data Processing System

This section shows the system designed for analyzing e-nose responses of the wine samples analyzed. The aim of this system is to perform a classification of the wine samples and prediction of the sensory panel attributes from the e-nose responses. The data collected were analyzed by means of pattern recognition techniques using commercial software packages. Matlab ® was used for linear methods like Independent Component Analysis (ICA) and non-linear methods based on Artificial Neural Networks (ANNs) like backpropagation and probabilistic neural networks.

Preprocessing and feature extraction. The goal of feature extraction is to find a low-dimensional mapping $f : x \in \Re^N \rightarrow y \in \Re^M (M < N)$ that preserves



Eje x: 1
Eje y: 2

☑ S1 ☑ S2 ☑ S3 ☑ S4 ☑ S5 ☐ S6

☑ M1 A a.c.
☑ M2 A a.c.
☑ M3 A a.c.
☑ M4 G a.c.
☑ M5 M a.c.
☑ M6 T a.c.
☑ M7 G d.c.
☑ M8 G d.c.
☑ M9 G d.c.
☑ M10 T d.c.
☑ M11 A d.c.
☑ M12 M d.c.
☑ M13 M d.c.
☑ M14 M d.c.

Fig. 4. Screenshot of the program designed and used for the preprocessing and feature extraction of measurement data (left). Screenshot of the program designed and used for the dimensionality reduction and classification with Neural Networks (right).

most of the information in the original feature vector x. In our case, a previous optimization of the method of feature extraction and selection were performed [9]; the responses of the individual sensors were defined relative to the minimum resistance to 12% (v/v) of ethanol for all the measurements:

$$r = \frac{R_{wine}}{R_{calibration}}$$

where R_{wine} was the minimum resistance of the sensor in the measurement of wine and $R_{calibration}$ was the minimum resistance of the sensor in a solution of 12% of ethanol. This calibration was performed once a week, to eliminate the sensors drift [3]. Data were centered and scaled for further analysis. Figure 4 (left) show a screenshot of the program designed for preprocessing and feature extraction. It automatically processed the files of measurements and generated the output vectors for dimensionality reduction.

Dimensionality reduction and classification. The traditional method for reduction in dimensionality and visualization of multivariate measurement data is Principal Component Analysis (PCA) [8]. This work introduces the higher order statistical method called Independent Component Analysis (ICA) [9] as an alternate method of data processing for an e-nose data. The main goal of ICA is to find a coordinate system that makes the original signals as independent as possible, using higher order statistics from the probability [10]. Another program was designed in Matlab using the neural networks toolbox. It performs the dimensionality reduction using PCA, LDA or ICA and classification with several types of Neural Networks (Perceptron, BackPropagation and Probabilistic Neural Networks). It generates the corresponding plots and the confusion matrix obtained with the validation of ANNs. Fig. 4 (right) shows a screenshot of this program.

3 Discussion and Results

3.1 Independent Component Analysis (ICA)

The Independent Components have been obtained using fastICA package of Matlab. The input data are the normalized sensor responses for each sample of wine. FastICA reduced dimension to 3, and estimated 3 independent components. The decorrelation approach used was deflation: it estimate independent component one-by-one like in projection pursuit. The nonlinearity used was power of 3. The ICs plot of young white, young red and aged red wines are shown in fig. 6, 7 and 8 respectively. As is shown in fig.6, the clusters are well separated except in some partial overlapping between samples of W7 and W5. The other wines are clearly

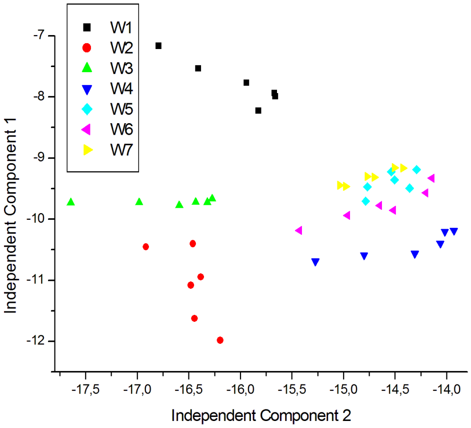


Fig. 5. ICA plot of the young white wine measurements

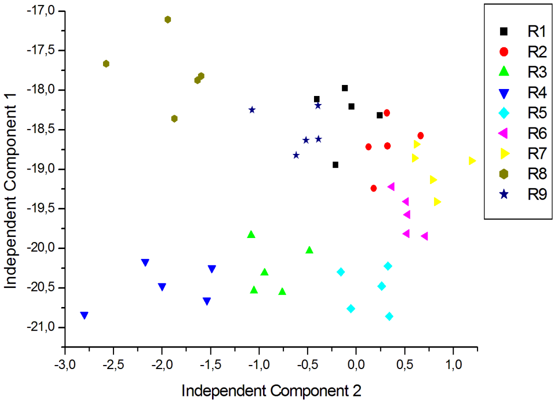


Fig. 6. ICA plot of the young red white wine measurements

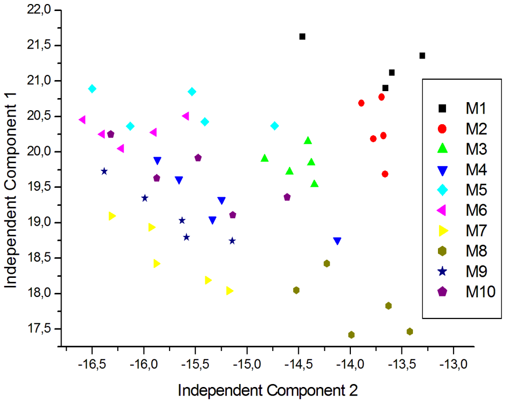


Fig. 7. ICA plot of the aged red wine measurements

separated among the others. Fig.7 shows the ICA plot for young red wines, a partial overlapping among some samples of R1, R2 and R9 can be observed. The clusters are well separated. Fig.8 show the ICA plot for aged red wines, there are several overlapping zones between some wines due to the similarity of the wines. Some of these zones correspond to measurements of the same wine but aged in a different type of oak barrel, e.g. M5-M6 and M9-M10 By the opposite, other wine samples, like M1-M2, M3-M4 and M7-M8 are well separated.

3.2 Artificial Neural Networks (ANNs)

Two types of neural networks were used for classification: Backpropagation and Probabilistic Neural Networks.

Backpropagation networks. In the training of feed-forward networks, several numbers of neurons in hidden layer has been tested. The optimal number turned out to be 14 neurons. The network was trained with the data obtained with the wine samples measurements. Leave one out (LOO) validation was performed in

Table 2. Confusion matrix obtained in validation of network with white young wines

	W1	W2	W3	W4	W5	W6	W7
W1	5	0	0	0	0	0	0
W2	0	5	0	0	0	0	0
W3	0	0	5	0	0	0	0
W4	0	0	0	5	0	0	0
W5	0	0	0	0	4	1	0
W6	0	0	0	0	0	5	0
W7	0	0	0	0	0	0	5

order to check the performance of the network. The classification success (percentage of well predicted number over total number of measurements) was 97% for young white wines, 87% for young red wines and 84% for aged red wines. As an example, confusion matrix obtained for white wines is shown in table 2.

Probabilistic networks. Probabilistic neural network were trained with the same data than backpropagation networks. Leave one out (LOO) validation was also performed. The classification success was 94% for young white wines, 84% for young red wines and 82% for aged red wines.

4 Conclusion

An electronic nose formed by gas sensors combined with a pattern recognition system based on Independent Component Analysis and Artificial Neural Networks was designed for classification of wine samples. Results showed classification successes higher than 87% for all types of wine. Furthermore, it can be concluded that is an alternate method for data processing of an e-nose.

References

1. Jackson, R.S.: Wine Science: Principles, Practice, Perception, 2nd edn. Academic Press, San Diego (2000)
2. Pearce, T.C., Schiffman, S.S., Nagle, H.T., Gardner, J.W.: Handbook of Machine Olfaction. Wiley-VCH, Germany (2003)
3. Gutierrez-Osuna, R.: Pattern analysis for machine olfaction: a review. IEEE Sens. J. 2(3), 189–202 (2002)
4. Lozano, J., Santos, J.P., Gutiérrez, J., Horrillo, M.C.: Comparative study of sampling systems combined with gas sensors for wine discrimination Sensors and Actuators B: Chemical, October 2007, vol. 126(2,1), pp. 616–623 (2007)
5. Lozano, J., Santos, J.P., Horrillo, M.C.: Enrichment sampling methods for wine discrimination with gas sensors. Journal of Food Composition and Analysis 21(8), 716–723 (2008)
6. Lozano, J., Santos, J.P., Horrillo, M.C.: Classification of white wine aromas with an electronic nose. Talanta 67, 610–616 (2005)
7. Aleixandre, M., Sayago, I., Horrillo, M.C., Fernández, M.J., Arés, L., García, M., Santos, J.P., Gutiérrez, J.: Analysis of neural networks and analysis of feature selection with genetic algorithm to discriminate among pollutant gas. Sensors and Actuators B: Chemical 103(1-2), 122–128 (2004)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, pp. 115–117. Wiley Interscience, Hoboken (2001)
9. Jutten, C., Hérault, J.: Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. Signal Processing 24, 1–10 (1991)
10. Comon, P.: Independent component analysis, a new concept? Signal Processing 36, 287–314 (1994)

Experiments and Reference Models in Training Neural Networks for Short-Term Wind Power Forecasting in Electricity Markets

Juan Méndez, Javier Lorenzo, and Mario Hernández

Institute of Intelligent Systems(SIANI)
Univ. Las Palmas de Gran Canaria
35017 Las Palmas, Canary Islands, Spain
`jmendez@iusiani.ulpgc.es`

Abstract. Many published studies in wind power forecasting based on Neural Networks have provided performance factors based on error criteria. Based on the standard protocol for forecasting, the published results must provide improvement criteria over the persistence or references models of its same place. Persistence forecasting is the easier way of prediction in time series, but first order Wiener predictive filter is an enhancement of pure persistence model that have been adopted as the reference model for wind power forecasting. Pure enhanced persistence is simple but hard to beat in short-term prediction. This paper shows some experiments that have been performed by applying the standard protocols with Feed Forward and Recurrent Neural Networks architectures in the background of the requirements for Open Electricity Markets.

1 Introduction

The economical and environmental advantages of the use of renewable energy sources (RES) in the electricity generation connected to public grids has a counterpart in the instability and unpredictability introduced into the electric systems. The electric generation from the more important renewable sources, wind and solar power, is mainly concerning with the weather in a local geographic area. However, the weather is a chaotic system whose predictability is limited. In each national system, the Transmission System Operator(TSO) deals in the management of the electric system in the different control and planning levels.

The statistical approach for short-term wind prediction has been used due to the system complexity of whether and the chaotic fluctuations of wind speed. The statistical models such as ARMA, ARX and Box- Jenkins methods have been used historically for short-term wind forecasting up to few hours ahead [1,2,3]. Giebel [4] reports some of the statistical state of the art models and methods for wind power forecasting which have been developed and used, such as time series models for up to a few hours by means of statistical approaches and neural networks, as well as models based on Numerical Weather Prediction(NWP).

The Electric Authorities of countries had included the power forecasting in its Regulatory Norms which goal is to preserve the quality of the electricity

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1288–1295, 2009.
© Springer-Verlag Berlin Heidelberg 2009

supply. The planning of an Electric System requires several levels related with different time scales as well as the whether forecasting. Very near short-term forecasting or nowcasting is the immediate prediction in time scale ranging from some minutes to several hours. Short-term forecasting concerns with time scale that ranges from one to three days, while middle-term forecasting covers from four days to several weeks time range.

The most interesting time scale in power predictions for using ANN is the nowcasting, which can be carried out by using the time series analysis. The short-term scale requires the cooperation between statistical and NWP tools, in regional and mesoscale weather models and cooperating with predictive systems as HIRLAN and MM5. The power forecasting for RES in Spanish Regulations is related to hourly periods of planning of the electricity market. All the power supplies and demands of the energy agents must be related to these hourly periods. The regulations for the short-term Spanish Electricity Market comprise two steps:

Short-term Forecasting. The RES producers, solar and wind farms, with power greater than 10 MW must provide 30 hours ahead the power forecasting for every hourly period of a full range of 24 hours.

Nowcasting. One hour ahead of each hourly period, corrections to the previous values can be sent to the Electricity Authority.

This means that in the nowcasting time scale, the computation of the predicted value must be carried out for the period covering two hours ahead. The second step can be carried out by using ANN, but the first requires the cooperation with NWP tools.

Artificial Neural Networks (ANN)[5] have been widely used for modeling and predictions in the field of renewable energy systems [6,7] because they are able to handle noisy, incomplete data and non-linear problems to perform predictions and classifications [8,9,10]. Hippert et al [11] have addressed the construction, and evaluation, of their performance of very large ANNs in electric systems to forecast the load profile. Recurrent ANN [12] have been used as generalizations of predictive systems as ARMA. Also, they can be used to generalize linear predictive systems as Kalman filter [13]. Recurrent and recurrence in each layer, called multilayer recurrent, architectures have been also used in wind power prediction[14].

Many studies about the use of ANN in wind power have been preformed, but the criteria to evaluate their performance have been mainly based on error parameters. Based on more modern standard protocol for forecasting[15], the published results will provide improvement criteria over the persistence or references models of its same place. Persistence forecasting is a simple model that is intrinsic to the data, that is, it is a no algorithm approach. Any new proposed algorithm is so good or bad as how much is able to overtake the persistence. The use of ANN can provide a suitable procedure to beat it and other reference model based on the Wiener predictive filter. An application is presented applying the standard protocols with Feed Forward(FNN) and Recurrent Neural

Networks(RNN) architectures in the background of the requirements for Open Electricity Markets. The paper contains a Methods section where the persistence and reference models are defined as well as the used ANN architecture and training procedures. The Results section shows the collected results of some experiments carried out by using wind power data series, and Conclusion section summarizes the main contributions and remarkable results.

2 Methods

Persistence is the simplest model for forecasting. It is based on the assumption of a high inertia in the subjacent physical model. If $y(t)$ is the value at time t of a time series, in persistence model the predicted value for k times ahead is: $\widehat{y}(t+k) = y(t)$. This kind of forecasting is really simple but can be very useful in practical, because it can be used as a reference model to compare different theoretical and practical applications. Any proposal of a new model or approach that requires some computational resource is required to have at least a better performance that this simple one. The level of improvement over this reference model must be a level of utility of the additional formal and computational cost. A high value in an error parameter, as MAE or RMSE, in a hardly predictable site can be a better result that a small value in a easily predictable site. However there are not a parameter to define what site has a hardly or easily predictable wind. A option is the use the own persistence as the reference to which compare the performance of proposed algorithms.

The pure persistence model can be overtaken by other model that involve persistence-like information. A reference model to compare different forecasting models has been proposed [16,15]. It is more advanced because it includes very short-term information, as persistence, and long-term information. This proposed reference model is an extension of the pure persistence as a linear expression: $\widehat{y}(t+k) = b + ay(t)$. A detailed analysis allows to show that is really the first order case of a more general linear predictive filter, as the Wiener filter with general expression:

$$\widehat{y}(t+k) = B + \sum_{i=0}^{m} A_i y(t-i) \tag{1}$$

where coefficients A_i and B can be computed from the matrix containing the cross correlation between $y(t+k)$ and $y(t-i)$. The constant parameter is $B = (1 - \sum_{i=0}^{m} A_i)\overline{y}$, where $\overline{y}$ is the large-term average value of $y(t)$. For the simplest case of first order filter: $\widehat{y}(t+k) = B + A_0 y(t)$, the value of the coefficient is:

$$A_0 = \frac{\int [y(t+k) - \overline{y}][y(t) - \overline{y}]dt}{\int [y(t) - \overline{y}]^2 dt} \tag{2}$$

In an Electricity Market we have two kind of power values, the spot power $P(t)$ and its hourly average P_h. For the TSO, the spot power is very important to assure the system stability at any time, but in the Electricity Market the hourly average is the required to RSE agents. The proposed reference model for wind power forecasting by Madsen [15], is applied for hourly average power in nowcasting as the required in the Spanish regulation as:

$$\widehat{P}_{h+2} = A_0 P_h + (1 - A_0)\overline{P} \tag{3}$$

where A_0 and $\overline{P}$ are parameters computed from large-term training information. This reference model, which we can call as improved persistence or Wiener persistence, is harder to beat because is based in the shortest-term information, P_h, and in the longest-term information, $\overline{P}$.

The basic theory for using ANN in prediction, its architectures and algorithms are in the area of adaptive and predictive linear filter[12]. The use of ANN has generated generalizations that has introduced improvements in the original linear models by allowing the construction of nonlinear predictive systems. The relationship between ANN, in special recurrent architectures, with linear predictive systems as ARMA allows nonlinear generalizations of previous statistical linear approaches.

A generalization of recurrent ANN is the multilayer recurrent[12,14]. In the wind power forecasting the problem can be formulated by using Feed Forward(FNN), without feedback, or Recurrent(RNN) ones:

$$\widehat{P}_{h+2} = F\left[V_h, \ldots, V_{h-n+1}, P_h, \ldots, P_{h-m+1}\right] \tag{4}$$

The used training procedure was the Bayesian regularization [17,18] which updates the weight and bias values according to the Levenberg-Marquardt [19,20] optimization procedure. It uses as goal function a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. The Bayesian regularization implementation that has been used is the implemented in the training function *trainbr* of the Neural Networks Toolbox of MATLAB[21]. The NARX architecture have been used for RNN with the same window size for input data, the wind speed, and feedback data, the wind power.

3 Results

We have used a wind data series acquired in Gran Canaria Island(Spain). The wind speed series comprise about 33 days data from a meteorological tower in time steps of one minute. Wind power series are obtained from the wind speed at 40 meters high and from a power transfer function with 5 and 12.5 m/sec cut-off values. Relative values about the nominal values, $P(t)/P_n$, are used in the power series.

The data set was split in two subset, the train and test. The train data is 2/3 of the global data. The standard protocol for performance evaluation suggested by Madsen[15] was used. It includes the definition of the Evaluation Criteria(EC)

Table 1. Comparative results for two hours ahead prediction by using several RNN configurations trained with Bayesian regularization. All Evaluation Criterion and their improvements over the reference model are in percent(%) normalize to the nominal power. The mean and standard deviation, $\mu \pm \sigma$, values are provided for 25 training trials.

	Pers. Ref.	RNN1	RNN2	RNN3	RNN4	RNN5
Delay		(2:3)2	(2:5)4	(2:7)6	(2:7)6	(2:7)6
Hidden Nodes		80	40	10	40	60
BIAS	0.6 0.9	0.5 ± 0.1	0.3 ± 0.1	0.1 ± 0.3	0.3 ± 0.4	0.3 ± 0.1
MAE	14.5 15.3	15.5 ± 0.2	15.3 ± 0.1	15.7 ± 0.5	15.3 ± 0.2	$15.3 \pm 0,1$
RMSE	23.7 22.3	22.3 ± 0.3	21.6 ± 0.1	22.5 ± 1.2	21.5 ± 0.1	21.6 ± 0.1
SDE	23.7 22.3	22.4 ± 0.3	21.6 ± 0.1	22.5 ± 1.2	21.6 ± 0.1	21.6 ± 0.1
Imp_MAE		$-0.4\pm, 1.2$	1.1 ± 0.5	-2.5 ± 3.3	0.6 ± 1.0	0.4 ± 0.9
Imp_RMSE		0.0 ± 1.2	3.3 ± 0.3	-1.0 ± 5.3	3.3 ± 0.6	3.2 ± 0.6
Imp_SDE		-0.1 ± 1.2	3.2 ± 0.3	$-1.1\pm, 5.3$	3.2 ± 0.6	3.1 ± 0.6

Table 2. Comparative results by using several FNN networks configurations. Additional data are the same as in Table 1.

	FNN1	FNN2	FNN3	FNN4	FNN5	FNN6
Delay	(2:4)3	(2:4)3	(2:6)5	(2:6)5	(2:11)10	(2:11)10
Hidden Nodes	3	6	5	10	10	20
BIAS	3.0 ± 1.8	4.0 ± 2.5	1.4 ± 0.3	1.4 ± 0.9	2.4 ± 2.4	3.2 ± 3.1
MAE	16.2 ± 1.0	16.8 ± 1.2	15.7 ± 0.4	16.0 ± 0.7	16.7 ± 1.3	17.4 ± 1.7
RMSE	22.7 ± 0.4	22.9 ± 0.6	22.2 ± 0.4	22.4 ± 0.5	22.6 ± 0.8	23.4 ± 1.3
SDE	22.5 ± 0.2	22.5 ± 0.3	22.2 ± 0.3	22.4 ± 0.5	22.5 ± 0.6	22.0 ± 1.1
Imp_MAE	$-4.8\pm, 6.5$	-8.6 ± 8.0	-1.3 ± 2.7	-3.1 ± 4.8	-7.4 ± 8.0	-12.3 ± 10.7
Imp_RMSE	-2.1 ± 2.0	-2.9 ± 3.0	2.7 ± 1.6	-0.6 ± 2.3	-1.5 ± 3.4	-4.7 ± 5.8
Imp_SDE	-1.0 ± 0.9	-0.8 ± 1.2	$0.4\pm, 1.5$	-0.4 ± 2.1	-0.5 ± 2.5	-2.9 ± 5.0

BIAS, MAE, RMSE and SDE, and also the improvement over the reference model which are computed in percent value as:

$$Imp_{ref,EC}(\%) = 100\frac{EC_{ref} - EC}{EC_{ref}} \tag{5}$$

Many training procedures of ANN use optimization procedures that run from initial random states. The optimization tries to reach a minimum value of some goal function, but the reached value and the trained network depend on the initial random state. In the practice, that means that the performance of a trained ANN has some random degree. To reduce the uncertainty in the results, we provide the mean and the standard deviation obtained from 25 training trials as: $\mu \pm \sigma$. Following the suggestion of Zang[22] that users should pay more attention to selecting the number of input nodes, we have cross correlated the power with itself and correlated it with the wind speed and concluded that the highest values are for offsets until the range of 4-6 hours back. It means that the size of the more useful data window must be around this range.

Tables 1 and 2 contain the results for several configurations of RNN and FNN respectively. Table 1 contains also the error values for the persistence and reference model. The computation of the reference model data was performed by using the train set, its parameters are: $A_0 = 0.82$ and $\overline{P} = 0.68$. The reported results are related to architectures including one hidden layer. The experiments have shown that more layers increases the computational cost and have no better performance. In both tables, the delays are taken in relation to the prediction time; they are represented as: $(h_1 : h_2)w$, where $w = h_2 - h_1 + 1$ is size of the time window. In all cases $h_1 = 2$ to met the regulations. Remark that the values of BIAS and MAE are related to the first moment of the error, therefore they are related to the generated power, but the values of RMSE and SDE are related to the second order moment and the variance of the error.

All the tested RNN architectures perform better on BIAS values, such as significatively reduce the level in relation to the reference model and the persistence. It means that the feedback of RNN architectures systematically corrects the biased offset in the prediction. The FNN architectures without such feedback are systematically biased. The inclusion of innovation filters can be needed for the FNN case but is no necessary for the RNN one. However, in MAE criterium the persistence value is not beaten neither reference nor any tested ANN architecture. The variance of the error provided by RMSE and SDE criteria are outperformed by some RNN architectures in relation to persistence, reference model and FNN. The range of parameters that provide better results are around values 4 and 6 for windows size, and around 40 for hidden nodes. The use of narrow windows or lower number of hidden nodes performs worse. There are not tradeoff between reducing the window size and increasing the hidden nodes as shows on the RNN1 case. The increasing of hidden nodes does not performs much better as is shown in RNN6 case. The FNN architectures are more unstable, eg. the FNN3 have a good improvement of 2.7 in mean value in the RMSE criterium, but has a big standard deviation value of 1.6. It is unstable if

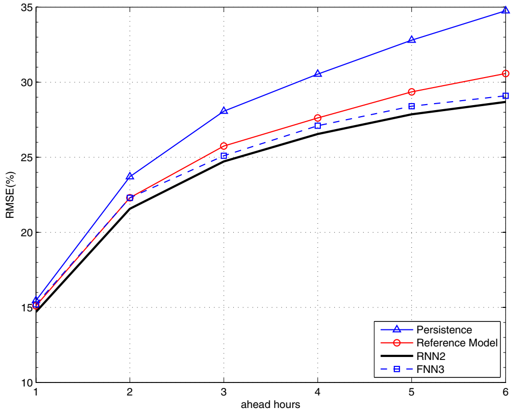


Fig. 1. Comparative RMSE of several models in the very short-term prediction

compared with the RNN2 case with 3.3 value in mean and 0.3 value in standard deviation.

Figure 1 shows the comparative performance in several hours ahead for the RMSE criterium. The included models are the persistence, the reference model the RNN2 and the FNN3 cases. It is shown that the reference model performs much better that the persistence and both ANN cases outperform the reference model. Also it is shown that the relative efficiency of the predictive models of ANN in relation to persistence increases when increases the ahead hours.

4 Conclusions

The short-term forecasting of wind power for Electricity Markets requires two kind of time scales prediction. The first requires detailed prediction for 1-2 days ahead, which needs the cooperation of some tools of NWP. The second is for the time scale of few hours ahead, which can be carried out by using time series analysis. In this time scale, ANN can be applied successfully for wind power forecasting useful in Open Electricity Markets.

This study has used the standard protocols to evaluate the performance of forecasting procedures that some authors have introduced. We have compared the results according these protocol. We have shown that the new reference model, based on the first order Wiener filter, perform better in variance criteria as RMSE and SDE, but it is worse in first order moment as BIAS and MAE. Some ANN architectures, as Recurrent and Feed Forward, have been tested. The main conclusion is that Recurrent architectures have better performance in first and second order statistical moments and can beat the reference model in the range of nowcasting useful in the Electricity Market.

Acknowledgment

This research was partially supported by the Spanish Ministry of Science and Innovation, grant TIN2008-06068. We acknowledge to the Instituto Tecnológico de Canarias (ITC) for the use of the wind speed series used in this paper that were acquired in its facilities in Pozo Izquierdo, Gran Canaria Island.

References

1. Nielsen, H.A., Madsen, H.: Wind power prediction using arx models and neural networks. In: Proceedings of the Fifteenth IASTED International Conference on Modelling, Identification and Control, pp. 310–313 (1996)
2. Landberg, L., Giebel, G., Nielsen, H.A., Nielsen, T., Madsen, H.: Short-term prediction–an overview. Wind Energy 6, 273–280 (2003)
3. Nielsen, T.S., Madsen, H., Nielsen, H.A., Pinson, P., Kariniotakis, G., Siebert, N., Marti, I., Lange, M., Focken, U., von Bremen, L., Louka, P., Kallos, G., Galanis, G.: Short-term wind power forecasting using advanced statistical methods. In: European Wind Energy Conference & Exhibition 2006 (2006)

4. Giebel, G.: EC Project ANEMOS: The state-of-the-art in short-term prediction of wind power a literature overview. Technical report, Risø National Laboratory (2003)
5. Haykin, S.: Neural Networs. Prentice Hall, Englewood Cliffs (1999)
6. Kalogirou, S.A.: Artificial neural networks in renewable energy systems applications: a review. Renewable and Sustainable Energy Reviews 5, 373–401 (2001)
7. Li, S., Wunsch, D.C., OHair, E., Giesselmann, M.G.: Neural network for wind power generation with compressing function. In: IEEE International Conference on Neural Network, vol. 1, pp. 115–120 (1997)
8. Alexiadis, M., Dokopoulos, P., Sahsamanoglou, H.S., Manousaridis, I.: Short-term forecasting of wind speed and related electric power. Solar Energy 63(1), 61–68 (1998)
9. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. Int. Journal of Forecasting 14, 35–62 (1998)
10. Kandil, N., Wamkeue, R., Saad, M., George, S.: An efficient approach for short term load forecasting using artificial neural networks. Electrical Power and Energy Systems 28, 525–530 (2006)
11. Hippert, H.S., Bunn, D.W., Souza, R.C.: Large neural netwoks for electricity load forecasting: Are they overfitted? Int. Journal of Forecasting 21, 425–434 (2005)
12. Mandic, D.P., Chambers, J.A.: Recurrent Neural Networks for Prediction. John Wiley and Sons Ltd., Chichester (2001)
13. Haykin, S.: Kalman Filtering and Neural Networks. John Wiley and Sons, Inc., Chichester (2001)
14. Li, S.: Wind power prediction using recurrent multilayer perceptron neural network. IEEE Power Engineering Society General Meeting 4, 225–230 (2003)
15. Madsen, H.: EC Project ANEMOS: A protocol for standardizing the performance evaluation of short-term wind power prediction models. Technical report, Technical University of Denmark (2004)
16. Nielsen, T.S., Joensen, A., Madsen, H., Landberg, L., Giebel, G.: A new reference model for wind power forecasting. Wind Energy 1, 29–34 (1998)
17. MacKay, D.J.C.: A practical framework for backpropagation networks. Neural Computation 4(3), 448–472 (1992)
18. Foresee, F.D., Hagan, M.: Gauss-newton approximation to Bayesian regularization. In: Proceedings of the 1997 International Joint Conference on Neural Networks, pp. 1930–1935 (1997)
19. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. The Quartely of Applied Mathematics 2, 164–168 (1944)
20. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics 11, 431–441 (1963)
21. Demuth, H., Beale, M., Hagan, M.: Neural Network Toolbox 6, User's Guide. The MathWorks, Inc. (2008)
22. Zhang, G.P., Patuwo, B.E., Hu, M.Y.: A simulation study of artificial neural networks for nonlinear time-series forecasting. Computers & Operations Research 28, 381–396 (2001)

Intrusion Detection Method Using Neural Networks Based on the Reduction of Characteristics

Iren Lorenzo-Fonseca[1], Francisco Maciá-Pérez[2], Francisco José Mora-Gimeno[2],
Rogelio Lau-Fernández[1], Juan Antonio Gil-Martínez-Abarca[2],
and Diego Marcos-Jorquera[2]

[1] Centre of Engineering and Systems,
José Antonio Echevarría Institute of Technology, Ciudad Habana, Cuba
`{ilorenzo,lau}@ceis.cujae.edu.cu`
[2] Department of Computer Technology,
University of Alicante, AP.99-03080, Alicante, Spain
`{pmacia,fjmora,gil,dmarcos}@dtic.ua.es`

Abstract. The application of techniques based on Artificial Intelligence for intrusion detection systems (IDS), mostly, artificial neural networks (ANN), is becoming a mainstream as well as an extremely effective approach to address some of the current problems in this area. Nevertheless, the selection criteria of the features to be used as inputs for the ANNs remains a problematic issue, which can be put, in a nutshell, as follows: The wider the detection spectrum of selected features is, the lower the performance efficiency of the process becomes and vice versa. This paper proposes sort of a compromise between both ends of the scale: a model based on Principal Component Analysis (PCA) as the chosen algorithm for reducing characteristics in order to maintain the efficiency without hindering the capacity of detection. PCA uses a data model to diminish the size of ANN's input vectors, ensuring a minimum loss of information, and consequently reducing the complexity of the neural classifier as well as maintaining stability in training times. A test scenario for validation purposes was developed, using based-on-ANN IDS. The results obtained based on the tests have demonstrated the validity of the proposal.

Keywords: Intrusion detection, Neural networks, Principal component analysis, Anomaly detection.

1 Introduction

Intrusion detection systems (IDS) have emerged to detect actions which endanger the integrity, confidentiality or availability of a resource [1] as an effort to provide a solution to existing security issues. This technology is relatively new, however, since its beginnings, an enormous number of proposals have been put forward to sort this situation out in the most efficient and cost effective of manners.

One detection approach used by IDS is based on the definition of normal behavior, classifying anything deviating from this state as an intrusion. This strategy is known as *anomalies detection* and has the advantage that it does not need to maintain attack patterns for classification.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1296–1303, 2009.
© Springer-Verlag Berlin Heidelberg 2009

For *anomalies detection*, the available literature reveals various research projects proposing IDS based on Artificial Neural Networks (ANN). One of the main downsides of this type of proposal is the basing of both, the specification of the architecture and the selection of input data on empirical criteria [2], [3], [4], [5], [6], making impossible to ensure the effectiveness of the selection, in other words, that the data will not cause any neural classifier malfunctioning. In fact, one of the criteria used in most of the works analyzed is to restrict selection of the input data according to size, which will ensure that the ANN used can support those data efficiently.

This work suggests a rather different approach: To freely analyze all the significant data involved in intrusion detection to later apply, in a secondary phase, compacting techniques which will pre-process and reduce entry space without any substantial loss of information. We propose to do so, in particular, Principal Component Analysis (PCA), one of the current top-performing algorithms in this area.

Having reviewed the state of the art (chapter 2) in related subjects, a method covering all the different phases implicated and the relationships between them is proposed (chapter 3). Subsequently (chapter 4) a test scenario is built using IDS based on a multilayer ANN. Finally, (chapter 5) the main conclusions deriving from research, as well as a proposal for appropriate future investigation lines are presented.

2 Related Works

An undeniable breakthrough in the field of Intrusion Detection has been the application of ANNs techniques. The neural networks have proven not only to be powerful classifiers but also to have remarkable generalization and learning abilities [7] as well.

Recent researches using ANN for IDS are based on their flexibility and adaptation to natural changes which may occur in the environment, and particularly, to its ability to detect patterns of previously unknown attacks.

Unsupervised learning techniques like the *Self-organizing map* (SOM) have been used to cluster the content of the network packets. In [8] and [9] appear two anomaly-detection systems that use SOM at network connection level, where each connection is defined with several features.

The research carried out in [3] is based on two different architectures: a *multilayer perceptron* (MLP) and a SOM, and its aim is to prove the efficiency of both ANNs for analyzing and classifying network packets as normal or dangerous. A sample of 29 fields was taken as ANN input data, and the efficiency was more than 90% for both ANN.

In [4] was developed *SADAM*. Its main objective was to explore the potential uses of a SOM as the core of the analysis system of IDS. *SADAM* does not analyze the content of the packets, and therefore it cannot detect attacks made at application level. Its purpose is the detection of attacks of ports scanning and denial of services, basing on the analysis of the packet head data. For the detection, 21 fields were taken into account, in addition to a series of statistical variables which show development of traffic in a certain window of time.

In [5] the performances of a BPL and a RBF networks were judged against. The input data used for both consisted on 41 fields of the HTTP protocol, without taking into account the remaining packet protocols. Both networks classification power levels topped the charts, but RBF outperformed its peer efficiency wise.

In [6] is just analyze the TCP data of the packets, leaving the likes of UDP and ICMP protocols out. The information contained in the data area of the packets was also taken into account. For this reason, in addition to using the most significant information of the TCP head, all characters of the packet content were selected. The obtained network had a detection power of more than 95 %.

Encouraging examined works' obtained results notwithstanding, important limitations still linger. Those works can be summarized into two large groups. The first of them refers to works which employ a high number of inputs for the neural network, which render the training as well as this system's usage inefficient. Conversely, the second group contains those in which little numbers of input data are used, with the subsequent reduction in detection's scope, leaving out some protocols or omitting to consider the data of the packet content. Therefore, the controversy between the optimum number of input selected for an adequate functioning of the network and the amount of TCP/IP packet fields required to detect the greatest number of intrusions, is one of the problems at the essence of this type of research. Due to this fact, there is another group of recent researches which attempts to select and reduce the amount of data required for classification based on TCP/IP packets and on traffic's general characteristics.

In [10] a work using a Support Vector Machine (SVM) as classifier was carried out. The inputs used are the 41 variables referring to characteristics of TCP/IP traffic. An interesting result coming out of this research is the identification of the most influential variables in order to consider that an intrusion has occurred. Other works aim at the use of essay-error techniques [11] or statistical techniques [12].

Previous researches make intensive use of feature selection algorithms where major weight variables for the classification are defined. The emergence of new attacks involving new variables changes the influence of the fields in the classification, reason why this solution is not general enough. Others like [13], [14], [15] use feature reduction algorithms (principally PCA). Their results are good, but they are only focused on TCP/IP traffic's general information, disregarding packet information and losing this way an important classification range.

3 Scalable Intrusion Detection Proposed Method

This work proposes a method, which is graphically represented in Figure 1, based on the combined efforts of application of PCA and ANN as pre-processing and classification techniques respectively. The application of PCA is intended to solve the discrepancy between the need to take sufficient data for classification and the performance of the classifier by keeping the number of necessary data at its minimum. Therefore, this method is looking for high levels of efficiency and scalability in ANN's application for intrusion detection, without sacrificing the effectiveness achieved in previous works.

The method initially proposes the analysis of the structure of the TCP/IP packet in order to identify the relevant characteristics for classification (*Selected TCP/IP Pattern*). The *TCP/IP data* responding to the selected pattern are processed by *PCA* to obtain the base change matrix (*TCP/IP Model*) which permits data to be re-expressed using fewer dimensions, thus achieving efficiency and scalability. This way a *PCA filter* is obtained which enables *TCP/IP traffic* to be compacted and used as input vector for the *ANN based IDS* in order to obtain the classification reports.

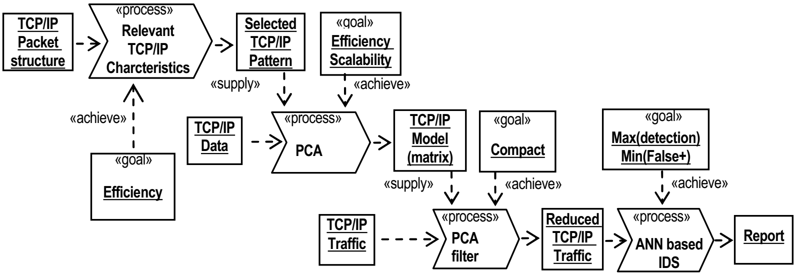


Fig. 1. Method proposed for Intrusion Detection based on the reduction of characteristics

3.1 Data Analysis and Selection

The first stage in the proposed method is *Identification of relevant TCP/IP characteristics* (Fig.1). This stage is primarily concerned with determining the characteristics of network traffic which are significant for the classification.

Each of the protocols which make up the TCP/IP includes a considerable amount of information required for its counterpart at the receiving machine [16]. Much of this data, due to its significance in communication, are used to identify a packet as intrusive. Therefore, for the purpose of obtaining an adequate classifier to detect anomalies in the TCP/IP traffic, it is necessary to carry out a detailed study of the protocols. Based on this study, a total of 419 different features for each TCP/IP packet were selected. These features form the *selected TCP/IP pattern* for the proposed method.

The time required for the ANN training increases with the square of the amount of selected input data [17]. Therefore, new input data should be incorporated to the ANN only if that represents a significant contribution to the improvement of the quality of classification.

3.2 Pre-processing Data

To reduce the input characteristics, a pre-processing method is applied during the second phase. The method used is PCA with the aim to obtain a data model which will facilitate the elimination of redundant and noisy dimensions [18].

To obtain the model, it is necessary to have an available set of previously acquired data responding to the *selected TCP/IP pattern* which will provide training for the model. This set should include both intrusive and normal packets. In this work is used the variant of capturing real packets in order to obtain training and test sets. *Nessus* [19] for the attack packets and *Ethereal* [20] as *sniffer* for packets capture were our tools of choice.

Having obtained an adequate set of test data, the 419 features taken into account in the *TCP/IP Fields* will be subjected to the re-dimensioning algorithm via PCA application looking for efficiency and scalability in the classification. The PCA objective is to compile a wide group of variables in a new set (fewer amounts) without losing any significant part of the original information. Algebraically, the PCA generates new variables (PC) by means of a linear combination of the original *p* variables. In this case, replacing the *p* variables with *m* components, the dimensionality of the problem is reduced, thus conserving almost all the information. Furthermore, in the stage subsequent to training, greater processing speed is achieved, which is a basic requirement for IDS based on neural networks.

A model corresponding to 20 PC was selected as it reduces significantly the initial vector (419 variables) and attempts to express a high percentage of the original information. With this new dimension it is possible to reduce 95.2% of the size of the original vector. At the same time, the 20 PC model is able to express over 95% of the original information.

In addition to the model's ability to express the data variability, it is important to assess its power of discrimination with respect to the different classes contained in the data. In our case there are two data classes: *Normal* and *Attacks*. Therefore, the existence of differences between these classes in the new information structure is convenient. The discriminatory power of the model is shown by means of the scores (Fig.2).

The fact that the model has a high capacity for expressing the variability of the data (above 95%) and that it contains principal components with discrimination power leads to the conclusion that we are in the presence of a good model of the original data.

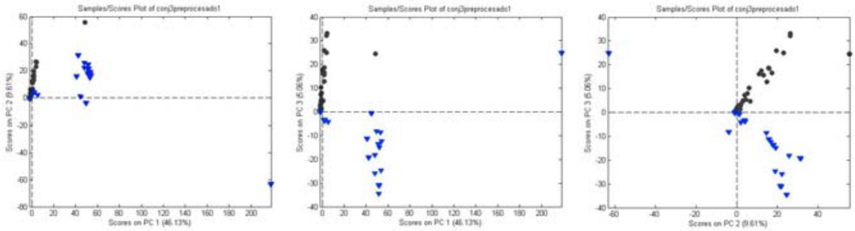


Fig. 2. (a) Scores of PC1 vs. PC2 (b) Scores of PC1 vs. PC3 and (c) Scores of PC2 vs. PC3

3.3 Application of the PCA Filter

The selected model of 20 PCs provides a data transformation matrix (*TCP/IP Model*). This matrix constitutes the main element of the *PCA filter* on the *TCP/IP Traffic* due to the fact that its application to the captured data is the source of the data expressed in the new base. The matrix (*M*) has as many columns as the set of data has variables (419) and as many rows as the number of PCs selected in the model (20). Each row of the *M* matrix is a PC expressed as a linear combination of the 419 original variables.

The *TCP/IP Traffic* data should be expressed as a matrix containing 419 rows corresponding to the variables of every packet and as many columns as packets captured.

With this transformation is obtained the reduced matrix, which expresses each of the TCP/IP packets based on the selected PC. This result corresponds to the *Reduced TCP/IP Traffic* which will serve as input to the neural classifier.

3.4 IDS Development Based on a Neural Classifier

When the traffic data has been reduced, they have to enter the *ANN based IDS* in order to obtain the classification reports.

The ANN used in this particular research is a MLP [21]. This choice is based on its usage's simplicity, allowing for attention to be focused on the results produced by the reduction of characteristics applied to this type of problem, which is the main objective of this work.

Having selected the type of ANN to be used, it is necessary to train it in order to define the optimal architecture. The first step consists on determining the number of ANN inputs and outputs. The inputs were defined by the 20 PC of the data model. The output is a binary value: *0* corresponds with packets classified as normal whilst *1* is assigned to the dangerous ones. MatLab was used for the ANN training considering the facilities it provides in this area.

Approximately 350 networks were trained; the top 5 behavior wise are shown in table 1, organized in ascending order according to their validation error value, thus the first shows the best result obtained.

Table 1. Neural Networks trained in 3 layers

Number of Neurones			**Transfer Function**			**Training Epochs**	**Validation Error**
Input layer	**Hidden layer**	**Output layer**	**Input layer**	**Hidden layer**	**Output layer**		
20	**36**	**1**	**tansig**	**tansig**	**logsig**	**12**	**6.47E-16**
20	62	1	logsig	logsig	logsig	8	8.42E-08
20	42	1	logsig	logsig	logsig	7	1.25E-07
20	12	1	logsig	logsig	logsig	5	1.64E-07
20	9	1	logsig	logsig	logsig	20	1.80E-07

Tests were also carried out adding a new hidden layer to the network, but experimental results showed that the three layers ANNs models always outperformed the four layers ones.

The ANN which proved to be the most efficient in the test stage was that of three layers with 20 neurons in the input layer, 36 in the hidden layer and 1 in the output layer, using the transfer functions *tansig* and *logsig* respectively (Figure 3).

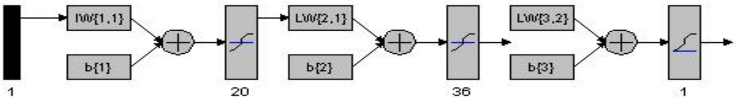


Fig. 3. Selected ANN architecture

4 Tests and Results

With the proposed method we have obtained: A filter able to compact captured TCP/IP traffic, a test network traffic data, and an IDS based on ANN. With these, we built our test scenario for the purpose of measuring the proposal's performance. Aiming to optimize the data set, a cross validation was used to test the method.

Once the selected ANN was trained, the results were judged against the test data in order to determine the amount of errors. With the testing data set, 100% effectiveness was obtained for normal packets classification, 99.33% for classification of attacks, which gave us a total effectiveness of 99.91%.

Performance wise, the use of PCA as pre-processing and re-dimensioning the input vector algorithm, provides undeniable advantages for the speed of ANN. The time required for the ANN training, increases with the square of the amount of input data. Therefore it can be said that the ANN implemented with 20 inputs is approximately 438 times faster in training time than another with 419 inputs.

In order to obtain results that can be compared with other IDS proposals, we need to use a standardized proving data. To date, DARPA intrusion detection evaluation data is the most comprehensive set known to be generated for the purpose of evaluating the performance of any given IDS [22]. The results with this dataset showed a promising performance: a detection rate of 90% with only one false positive per day.

5 Conclusion

This work has proposed a method for Intrusion Detection in which the ANN used does not limit the information regarded as important for detection purposes. To this end, the ANN's input vector is pre-processed and significantly reduced, suffering a minimum lost of relevant information. This way it provides a remedy for some of the most important problems associated with ANN's utilization in a growing information environment: exponential increases of learning times and poor detection performances. The results show the validity of the proposed method.

Furthermore, due to the application of the proposed method, it was possible to notably increase the input information used for detection without having an important impact on system's learning and detection times, obtaining much more higher levels of efficacy with a 0.09% of errors.

We are currently improving the proposed method in order to make the whole process to work unsupervised. In particular, we are concentrating our efforts in three directions: on a new model for the dynamic securing of PCA filters; on the qualitative study of the relevant parameters of TCP/IP traffic without restriction on size, and finally,on the identification of unsupervised neural network models which may be more effective for intrusion detection.

Acknowledgements

This work has been financed by the Spanish Ministry of Education and Science as research project TIN2006-04081, the Generalitat Valenciana as research project GV/2007175 and MAEC-AECID as Ph.D. scholarship.

References

1. Debar, H., Viinikka, J.: Introduction to Intrusion Detection and Security Information Management. In: Aldini, A., Gorrieri, R., Martinelli, F. (eds.) FOSAD 2005. LNCS, vol. 3655, pp. 207–236. Springer, Heidelberg (2005)
2. Ghosh, A., Michael, C., Schatz, M.: A real-time intrusion detection system based on learning program behavior. In: Debar, H., Mé, L., Wu, S.F. (eds.) RAID 2000. LNCS, vol. 1907, pp. 93–109. Springer, Heidelberg (2000)
3. Grediaga, A., Ibarra, F., García, F., Ledesma, B., Brotons, F.: Aplication of Neural Networks in Network Control and Information Security. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3973, pp. 208–213. Springer, Heidelberg (2006)
4. Cortada, P., Sanroma, G., Garcia, P.: IDS based on Self-Organizing Maps. Technical Report, RedIRIS (2002)
5. Zhang, C., Jiang, J., Kamel, M.: Comparison of BPL and RBF Network in Intrusion Detection System. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 460–470. Springer, Heidelberg (2003)
6. Zanero, S., Savaresi, S.: Unsupervised Learning Techniques for an Intrusion Detection System. In: ACM Symposium on Applied Computing SAC 2004, pp. 41–419 (2004)
7. Freeman, J., Skapura, D.: Neural Networks. In: Algorithms, Applications, and Programming Techniques. Addison-Wesley, Reading (1991)
8. Ramadas, M., Ostermann, S., Tjaden, B.: Detecting Anomalous Network Traffic with Self-Organizing Maps. In: Vigna, G., Krügel, C., Jonsson, E. (eds.) RAID 2003. LNCS, vol. 2820, pp. 36–54. Springer, Heidelberg (2003)
9. Lichodzijewski, P., Zincir-Heywood, A., Heywood, M.: Dynamic Intrusion Detection using Self-Organizing Maps. In: 14th Annual Canadian Information Technology Security Symposium (2002)
10. Mukkamala, S., Sung, A.: Feature Ranking and Selection for Intrusion Detection Systems Using Support Vector Machines. Technical Report, Institute of Minería y Tecnología, Nuevo México (2003)
11. Sung, A., Mukkamala, S.: Identifying important features for intrusion detection using support vector machines and neural networks. In: Proceedings of International Symposium on Applications and the Internet (2003)
12. Chebrolua, S., Abrahama, A., Thomasa, J.: Feature Deduction and Ensemble Design of Intrusion Detection Systems. Computers & Security 24, 295–307 (2005)
13. Xu, X., Wang, X.: An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS, vol. 3584, pp. 696–703. Springer, Heidelberg (2005)
14. Bouzida, Y.: Efficient Intrusion Detection Using Principal Component Analysis. Technical Report Departement RSM GET/ ENST Bretagne (2005)
15. Nguyen, D.: A Reconfigurable Architecture for Network Intrusion Detection using Principal Component Analysis. Technical Report, Northwestern University Evanston (2005)
16. Comer, D.: Internetworking with TCP/IP, vol. 1. Prentice Hall, Englewood Cliffs (2005)
17. Kauzoglu, T.: Determining Optimum Structure for Artificial Neural Networks. In: Proceedings of then 25th Annual Technical Conference and Exhibition of the Remote Sensing Society, Cardiff, K, September 1999, pp. 675–682 (1999)
18. Esbensen, K.: Multivariate Data Análisis – in practice. Camo Press AS (2002)
19. Tenable Network Security, `http://www.nessus.org`
20. Ethereal, `http://www.ethereal.com`
21. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
22. MIT Lincoln Laboratory: DARPA Intrusion Detection Evaluation, `http://www.ll.mit.edu/IST/ideval/index.html`

Evaluating the Performance of the Multilayer Perceptron as a Data Editing Tool

Mª-Dolores Cubiles-de-la-Vega[1], Esther-Lydia Silva-Ramírez[2],
Rafael Pino-Mejías[1], and Manuel López-Coello[2]

[1] Department of Statistics and Operational Research, University of Seville,
Av. Reina Mercedes s/n, 41012 Sevilla, Spain
cubiles@us.es, rafaelp@us.es
[2] Department of Languages and Computer System, University of Cadiz,
C/ Chile 1, 11003 Cadiz, Spain
esther.silva@uca.es, manuel.coello@uca.es

Abstract. Usually, the knowledge discovery process is developed using data sets which contain errors in the form of inconsistent values. The activity aimed at detecting and correcting logical inconsistencies in data sets is named as data editing. Traditional tools for this task, as the Fellegi-Holt methodology, require a heavy intervention of subject matter experts. This paper discusses a methodological framework for the development of an automated data editing process which can be accomplished by a general nonlinear approximation model, as an artificial neural network. We have performed and empirical evaluation of the performance of this approach over eight data sets, considering several hidden layer sizes and seven learning algorithms for the multilayer perceptron. The obtained results suggest that this approach offers a hopeful performance, providing a promising data cleaning tool.

Keywords: Multilayer perceptron, inconsistencies, data editing, learning algorithms, MATLAB.

1 Introduction

Large amounts of data are continuously collected for different purposes, but the resulting data files almost unavoidably contain errors in the form of missing or inconsistent values. The second error appears when the interviewee does not answer correctly to a question or the answer to the question is not correctly recorded. This second type of error is the target of this paper.

Data editing is the process whereby data are examined for inconsistent values and the errors that are identified are corrected. Data editing is an indispensable tool for improving the quality of information, but it is usually an expensive and time consuming process. The main data editing methodology is based on the specification of a set of edits. An edit can be defined as a set of logical conditions or restrictions on the values of data. Fellegi and Holt's contribution [1] offers a prominent methodological framework to identify the minimal set of values in a record to be changed to satisfy all edits, providing a way of generating all the edits that are implied

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1304–1311, 2009.
© Springer-Verlag Berlin Heidelberg 2009

by a set of explicitly defined edits. However, there exist some shortcomings in this approach, for example the heavy work of the subject matter experts, and other computational limits [2].

Machine learning models are a natural way for generating automatic models. Petrakos et al. [3] propose tree-based models for automatically specifying the functional form of edits. Another approach is based on Artificial Neural Networks (ANN in what follows). We propose a methodology based on ANN trying to reduce the time employed by experts. A possible architecture is the multilayer perceptron, given its well known properties, for example the universal approximate capability [4].

2 A Data Editing Methodology

We consider a data set S comprising n records with p variables, where the collecting data process may have produced a certain number of incorrectly recorded values. Therefore, the available data set S can be considered as an approximation to the true data set T, where the whole nxp data matrix would have been correctly recorded. A data editing model is as a set of rules and procedures to obtain an approximation T^* to T working on the available data set S. Therefore, T^* is intended to be a corrected and improved version of S, obtained through the editing process.

Data editing process is usually based on subject matter experts which exploit their knowledge to transform S into T^*, frequently with the aid of Fellegi-Holt methodology. However, the resulting process is generally slow and resource and time consuming. Our alternative approach reduces the set of records that must be reviewed by the experts. Thus, a subset S^1 of S is processed by the traditional way, providing corrected data set T^{*1}, and therefore a supervised machine learning model can be trained from the couple (S^1, T^{*1}), where the rows of S^1 are the inputs and the rows of T^{*1} are the targets. Later, this model can be applied to $S^2=S\setminus S^1$ and therefore the obtained outputs are the edited data set T^{*2}. Finally, T^* is the union of T^{*1} and T^{*2}.

Nordbotten [5] studied the performance of the ANN data editing procedure with data sets which were generated from an imaginary structure, and where a perturbation scheme to build the pair (S^1, T^1) was followed. In this paper, we have performed a wider study where real and simulated data sets are exposed to a perturbation experiment.

There are many machine learning models which could be used in advance. However, with ANN only one model is required, whereas other techniques as decision trees or support vector machines need to build a model for each one of the variables in the data set. This way, an ANN model receives an input vector of variables corresponding to a record and it outputs an edited version of the whole input record. We have considered a three layered perceptron: an input layer with p nodes which receives values corresponding to the p variables existing in S, an output layer with p outputs which provide the p edited variables, and a hidden layer with H neurons. The hyperbolic tangent activation function $g(u)=(e^u-e^{-u})/(e^u+e^{-u})$ has been chosen in the hidden layer while the identity function is the activation function for the output layer in our experiments. We must remark that each one of the categorical variables must be codified by a vector formed by dummy variables 0-1, one for each class, so the number of inputs and outputs p is usually larger than the number of variables in the data file. For obtaining the edited value of a categorical variable, the largest predicted dummy variable provides the associated category as prediction.

3 An Empirical Evaluation

We have performed an empirical study of this data editing approach. Data sets with both quantitative and qualitative variables have been analyzed, and we have also studied the effect of the number of the hidden units, the learning rule and the number of epochs.

3.1 Data Preprocessing

For each data set, several preprocessing tasks are performed, according to the usual practical applications of ANN. First, all variables that are constant across the data file are deleted. Second, each quantitative variable has been categorized into four classes, defined by the three cutpoints corresponding to the first quartile, the median and the third quartile. Third, the values of categorical variables are codified with dummy variables 0-1. Finally, each data set is randomly split into training (70%) and test (30%) sets, in order to obtain reliable measures of the performance of the ANN models. We must remark that 70% is also the percentage used to fix the size of S^1 in our study. Future works will also consider other sizes.

3.2 Data Perturbation

We assume that each available dataset is complete and correct. Therefore, we can generate a perturbed version by introducing random errors of interchange. Thus, for each original variable Y, a perturbed variable Y^d is defined. We have considered 0.05 as the probability of error. This error rate can be considered high, for example for a data set with 10 variables, approximately 40 percent of the records of the perturbed data would be incorrect.

For a categorical variable Y with K categories $c_1, c_2,\ldots,c_K$, its associated perturbed variable is defined as follows: $Y^d=Y$, with probability equal to 0.95, while with a probability equal to 0.05, Y^d is defined by randomly choosing a value from the set of remaining values $\{c_1, c_2,\ldots,c_K\}\backslash Y$.

Over the correct data set T the previous perturbations are performed, and then a data set perturbed T^d is obtained. Each correct data set T and its associated perturbed version T^d are split into training (70%) and test (30%) sets. Therefore, two files, T^1 and T^2, were obtained with associated perturbed sets T^{d1} and T^{d2}. The inputs to the MLP are defined by the rows of T^{d1}, while the rows of T^1 are the target records. As it is explained in section 3.3, M=357 different parameter configurations for the MLP were considered, so a 10-fold validation procedure was followed to select the best configuration. The rows of T^{d2} were fed to the fitted MLP and the output records of the MLP, contained in T^{*2}, were compared with the true records contained in T^2. Moreover, the perturbation procedure is independently repeated 15 times.

3.3 Multilayer Perceptron Fitting

The MLP fitting requires several decisions as the learning algorithm, the random initialization of the MLP weights, the number of hidden units or the number of iterations (epochs) of the learning algorithm. Thus, different alternatives have been considered, as it is discussed in the following.

Table 1. Learning algorithms for the multilayer perceptron

Abbreviation	Matlab function	Algorithm
GD	Traingd	Gradient Descent
GDM	traingdm	Gradient Descent with Momentum
BA	traingda	Gradient Descent with an Adaptive Learning Rate
BE	trainrp	Resilient Backpropagation (RProp)
GC	traincgf	Conjugate Gradient Fletcher-Reeves Update
QN	Trainbfg	BFGS Quasi-Newton
LM	Trainlm	Levenberg-Marquardt

We have chosen seven of the main learning algorithms, as it is shown in table 1, where the name of the corresponding Matlab Neural Network Toolbox function appears in the second column.

The initial random configuration of weights may lead to very different solutions, thus, the training algorithm has been run 5 times, from 5 randomly generated initial weights vectors.

We have considered three sizes of hidden layer: 5, 10 and 15 nodes. The selection has been done from previous studies for these experiments, in which we observed that the use of larger sizes frequently led to worse results.

It was also observed that for 300 and up epochs the error decreased very slowly. So, the 17 studied training epochs are: 5, 10, 15, 20, 25, 30, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275 and 300.

The total number of weights for each MLP is $(2p+1)H+p$, where p is the number of inputs and outputs and H is the number of hidden units. For example, for the Soybean database with 35 qualitative attributes, $p=95$ when they are codified with dummy variables, and for $H=15$, the ANN model comprises 2960 weights.

The total number of different ANN models which have been studied for each data set is 15x10x7x3x17x5 = 267.750, that is, 15 perturbed data sets, the 10 training/test splits of the cross validation process, 7 learning algorithms, 3 sizes of hidden layers, 17 values for the epochs and 5 different initial weights.

For each data set perturbation, M=7x3x17=357 configurations are generated. For each one of these configurations, 10 values of the *GCD* criterion (explained in section 3.5) are available through the 10-fold procedure, and their mean value is computed. The minimum mean *GCD* guided us to the selected configuration.

The source code employed in this work has been written in Matlab 6.0, and Neural Network Toolbox was used for the construction of the MLPs.

3.4 Data Sets

Our study has been performed on eight data sets extracted from UCI-Repository [6]. A brief description of each one is showed in table 2, presenting the number of records (size), the number p of inputs/outputs for the multilayer perceptron, and the number of quantitative and qualitative variables.

Table 2. Data sets used in the empirical study

Name	Size	Number of inputs/outputs	Number of quantitative vars.	Number of cualitative vars.
Cleveland	303	25	6	7
Heart	270	25	6	7
Zoo	101	31	1	15
Hayes-Roth	132	15	0	4
Led7	500	14	0	7
Lymphography	148	60	0	18
Monk	432	17	0	6
Soybean	47	95	0	35

3.5 Evaluation Criteria

Several performance measures have been computed to evaluate the performance of MLP editing data model on the previously presented eleven data sets. A first criterion is the percentage of inconsistency errors that have been corrected, namely *COR*, defined as follows.

For a given variable X_j, we define x_{ij} as its correct value for the record i, d_{ij} the corresponding perturbed value and e_{ij} the estimated value that is obtained by the editing data model. The number of inconsistent values for X_j is $A=\#\{x_{ij}\neq d_{ij}\}$. The number of inconsistent values correctly edited by the neural net model will be denoted by $B==\#\{x_{ij}\neq d_{ij} \cap x_{ij}=e_{ij}\}$. Therefore, the percentage of inconsistency errors that have been corrected for X_j will be computed by $COR(X_j)=100*B/A$. Thus, the percentage of inconsistency errors correctly edited, *COR*, is obtained by averaging over the variables existing in the data set.

However, the number of inconsistency errors could increase after the data editing process, so we have also computed the percentage of inconsistency errors before (*BEF*) and after (*AFT*) the application of the MLP, computed with regard to the total of items (number of records multiplied by the number of variables) in the data set. The reduction of the percentage of inconsistency errors have been computed as a percent, namely $RPI=100(1-(AFT/BEF))$.

Finally, a global criterion of the whole data editing process (GCD) has been computed as the mean of *COR* and *RPI*. *GCD* is the measure that we have used in the 10-fold validation procedure to select the best configuration of learning algorithm, number of hidden units and number of epochs, for each perturbed data set.

4 Results

Table 3 shows the test mean of each criterion computed over the 15 test sets for each one of the 8 data sets, containing the distribution of the 10-fold selected learning algorithms, hidden layer size and number of epochs, computed as percentages over the 357 perturbed data sets (section 3.3). Fig. 1 exhibits the corresponding mean values of the criterion *BEF* y *AFT*.

Table 3. Mean values of the criteria for the test sets

Data set	COR	RPI	GCD
Cleveland	58.61	57.84	58.23
Heart	58.54	58.23	58.39
Zoo	86.08	86.56	86.32
Hayes-Roth	40.53	41.63	41.08
Led7	54.85	57.85	56.35
Lymphography	67.00	68.48	67.74
Monk	37.12	37.75	37.44
Soybean	69.27	69.99	69.63

From table 3 we can observe that the percentage of inconsistency errors correctly edited *COR* raises from 37.12% to 86.08%, while the reduction of the percentage of inconsistency errors *RPI* presents similar values, ranging from 37.75% to 86.56%. The column RPI shows a clear trend to improve the quality of the perturbed data sets, as it can also be observed in Fig. 1. Moreover, a Mc Nemar test was applied to perform a statistical comparison between the probability of inconsistency errors before and after the application of the MLP model. The significance value over the different repetitions was computed, being for the eight data sets and the different repetitions lower than 0.01 for the test sets, and therefore a statistically significant reduction of the number of inconsistency errors is usually achieved. Last column of table 3 shows the global criterion GCD. It is greater than 56% for six of the eight data sets, being HayesRoth and Monk the exception.

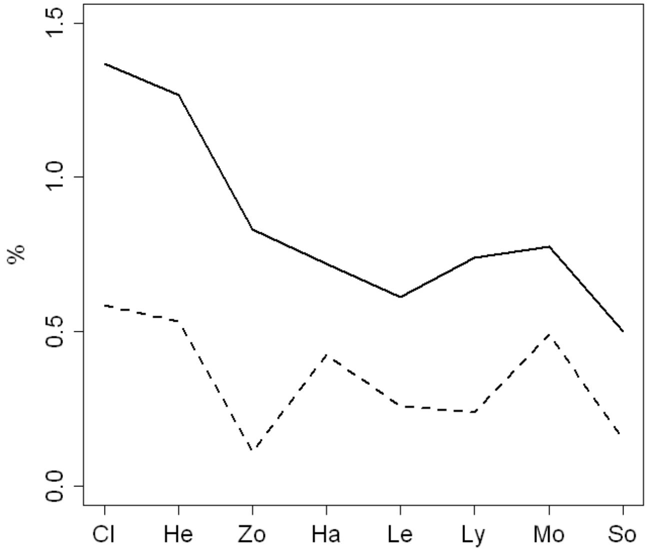


Fig. 1. Mean test values of the criterion BEF (upper line) and AFT (dashed line)

Table 4. Distribution of the selected parameters

Learning rule	%	Hidden layer size	%	Number of epochs	%
GD	27.6	5	82.3	5	37.0
GDM	9.3	10	8.9	10	8.5
BE	8.7	15	8.8	15	9.0
GC	17.1			20	18.4
QN	37.3			100	8.9
				125	9.3
				225	8.9

Table 4 contains the distribution of the 10-fold selected learning algorithms, hidden layer size and number of epochs, computed as percentages over the 8x15=120 perturbed data sets.

QN (BFGS Quasi-Newton) tends to be the preferred learning algorithm, being GD and QC the main alternatives. Two learning algorithms, BA and LM, are never selected. This table also suggests that simpler MLP are usually preferred, and indeed 5 hidden units are sufficient in the majority of the data sets. From last column we can observe that in almost 82% of the repetitions the number of epochs is not greater than 100, and therefore it is not recommended to allow for too many epochs in the training process.

5 Conclusions

A methodology for correction inconsistencies in data sets by artificial neural networks has been proposed. Ten data sets have been exposed to a perturbation experiment, and several architectures and training algorithms for the multilayer perceptron have been tested. Several criteria for evaluating the performance of the data editing model have also been computed.

The main conclusion is that multilayer perceptron models offer a hopeful performance as automatic data editing techniques. There has been a trend to prefer simpler models, with not many hidden units nor many epochs, while there is not any clearly preferred learning algorithm, although a quasi-newton procedure tends to be selected.

Acknowledgments. This work was partially funded by Institute of Statistics of Andalusia, grants 2007/00001428 and 2008/00000771.

References

1. Fellegi, I., Holt, D.: Systematic Approach to Automatic Edit and Imputation. J. Am. Stat. Assoc. 71(353), 17–35 (1976)
2. Manzari, A.: Combining editing and imputation methods: an experimental application on population census data. J. R. Stat. Soc. Ser. A-Stat. Soc. 167(2), 295–307 (2004)

3. Petrakos, G., Conversano, C., Farmakis, G., et al.: New Ways of Specifying Data Edits. J. R. Stat. Soc. Ser. A-Stat. Soc. 167(2), 249–274 (2004)
4. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (2005)
5. Nordbotten, S.: Editing Statistical Records by Neural Networks. Journal of Official Statistics 4(11), 391–411 (1995)
6. UCI Repository of Machine Learning Databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html`

A.N.N. Based Approach to Mass Biometry Taking Advantage from Modularity

Kurosh Madani, Abdennasser Chebira, and Véronique Amarger

Images, Signals and Intelligence Systems Laboratory (LISSI / EA 3956), PARIS-EST / PARIS 12 – Val de Marne University, Senart-FB Institute of Technology, Bât.A,
Av. Pierre Point, F-77127 Lieusaint, France
`madani@univ-paris12.fr`

Abstract. Over the recent years, new public security tendency to fit up public areas with biometric devices has emerged new requirements in biometric recognition dealing with what we call here "mass biometry". If the goal in "individual biometry" is to authenticate and/or identify an individual within a set of favored folks, the aim in "mass biometry" is to classify a suspect individual or behavior within a flow of mass customary information. In this case, the ability of handling relatively poor information and the skill of high speed processing become chief requirements. These antagonistic requests make the "mass biometry" and related applications among the most challenging frames. In this paper we present an ANN based system in a "mass biometry" context using facial biometric features. The proposed system takes advantage from kernel functions ANN model and IBM ZISC based hardware. Experimental results validating our system are presented and discussed.

Keywords: Mass Biometry, Artificial Neural Networks, Real-Time, kernel functions, Parallel Implementation.

1 Introduction and Problem Stating

Over the last decade a particular attention has been devoted on biometry based security issues and related applications, leading to a large variety of available products. However, the major efforts as well as the most of products have concerned individual authentication or identification: what we will call here "individual biometry". The main goal of "individual biometry" is to authenticate and/or identify an undesired individual within a set of favored folks. Individual biometry assumes a precise biometrical characterization of concerned individuals, even if requirements differ between authentication and identification tasks. In fact, in authentication (confirmation of desired individual) the precision concerns exclusively desired individuals, which should be reliably recognized as those belonging to the set of authorized individuals. An example is biometric access control, where the system identifies the authorized persons only (on the basis of their biometrical features: fingerprint, iris, etc…). In identification (recognition of incriminated individual) the precision concerns reliable recognition of incriminated (suspected) or hunted (pursued, tracked) individuals. An example is criminal investigations based on DNA, fingerprint, or other biometric features. If computing delay is a major requirement, it occupies a second rank comparing to the recognition accuracy's weight.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1312–1319, 2009.
© Springer-Verlag Berlin Heidelberg 2009

On the other hand, over the recent past years, new tendency to fit up public areas with biometric devices has emerged new requirements in biometric recognition dealing with what we call here “mass biometry”. Contrary to “individual biometry”, the main goal in “mass biometry” is to authenticate and/or identify an unusual (suspect) behavior within a flow of mass customary behaviors. An example is matching the presence (or identification) of an individual with a heavy “police record” within a flow of passengers in a rail station or in an airport. In fact, here the processing delay remains the chief requirement. That’s why, between biometric features’ precision and processing’s speed the preference goes to the second one. Additionally, due to the technological lake of mass oriented biometric devices, the biometric information involved in “mass biometry” remains relatively poor (lower quality) comparing to the case of the “individual biometry”. The most accessible (available) devices are those which fit up the aforementioned public areas: essentially digital cameras installed in streets, in rail stations or in airports.

On the side of biometric features, even if the fingerprint [1] remains the most popular biometric feature, over the past decade, an increasing number of works have concerned other biometric features (and issued biometric systems): voice biometry using speech processing issues [2], hand geometry based biometry [3], cognitive or mental characteristics extraction [4], those involving human psychology [5] or those dealing with multiple biometric traits using data fusion issues [6]. However, the above-listed reflections arguments lead to state that the most promising issue for “mass biometry” related applications remains the facial biometry based approach. In fact, it is reliable and realistic enough to state today that the near future years will be those of fitting up the public areas with digital cameras (came scopes) than to imagine that they will be those of public areas’ entrapment with more complex apparatuses.

In this paper we present an Artificial Neural Network (ANN) based face recognition system in a “mass biometry” context using facial biometric features. Our motivation to investigate an ANN based solution has been stirred by three points: their successes in solving nontrivial problems (especially those dealing with decision making and classification), their learning and generalization capabilities (extrapolation of learned tasks to unlearned situation) and availability of parallel electronic implementation for some of these models (as the IBM ZISC-036, offering reliable functional and programming skills). The paper has been organized as follows: section 2 will present a brief overview of kernel functions based ANN focusing their structure and their implementation within the IBM ZISC architecture. Section 3 will present the proposed system, its implementation, experimental protocol and validation results. Finally, the last section will conclude the paper.

2 Brief Overview of Kernel Functions Based ANN and ZISC Architecture

This kind of neural models belong to the class of “evolutionary” learning strategy based ANN ([7], [8], [9]). Including a unique hidden layer, the neural network’s structure is completed during the learning process. Fig.1 gives the bloc-diagram of such neural net (the left picture). The number of neurons in input layer corresponds to the problem’s feature space dimension. The output layer represents a set of categories associated to the input data. Connections between hidden and output layers are established dynamically during the learning phase.

A neuron from hidden layer is characterized by its "centre" representing a point in an N dimensional (N-D) feature space and some decision function, called also neuron's "Region Of Influence" (ROI). The solution is mapped in problem's N-D feature space thank to learning examples (centres) and associated ROI (e.g. regions around prototypes where generalization is possible). The right picture of Fig.1 shows example of the learning mechanism's principle for a 2-D feature space. When a prototype is memorized, ROI of neighbouring neurons are adjusted to avoid conflict between neurons and related categories. The neural network's response is obtained from relation (1) where C_j represents a "category", $V = [V_1 \quad V_2 \quad \ldots \quad V_N]^T$ is the input vector, $P^j = [P_1^j \quad P_2^j \quad \ldots \quad P_N^j]^T$ represents the j-th "prototype" learned as the neuron j in the hidden layer and λ_j the ROI associated to this neuron (neuron j). F(.) denotes the neuron's activation (decision) function, which is a kernel like function.

$$\begin{array}{lll} C_j = F\left(dist\left(V, P^j\right)\right) & \mathit{If} & dist\left(V, P^j\right) \le \lambda_j \\ C_j = 0 & \mathit{If} & dist\left(V, P^j\right) > \lambda_j \end{array} \tag{1}$$

The choice of the distance calculation (choice of the used norm) is one of the main parameters in the case of the kernel functions based neural models. The most usual function to evaluate the distance between two patterns is the Minkowski function expressed by relation (2), where V_i is the i-th component of the input vector and P_i^j the i-th component of the j-th learned pattern. Manhattan distance ($n = 1$, called also L1 norm) and Euclidean distance ($n = 2$) are Minkowski function's particular cases.

$$dist = \sqrt[n]{\sum_i \left| V_i - P_i^{\,j} \right|^n} \tag{2}$$

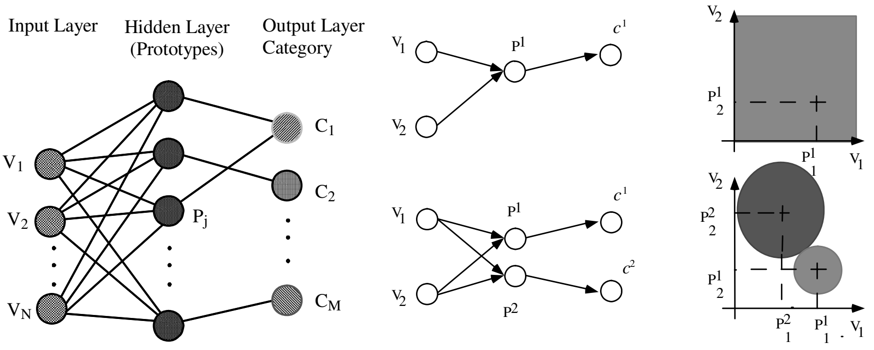


Fig. 1. Kernel Functions based ANN's bloc-diagram (left) and an example showing its learning mechanism solving a classification task in 2-D feature space (right)

The IBM ZISC-036 ([10], [11], [12]) is a parallel neural processor implementing kernel functions based ANN. Each chip is able of performing up to 250000 recognitions per second. It also integrates an incremental learning algorithm. This circuit is very easy to program in order to develop applications; a very few number of functions (about ten functions) are necessary to control it. Each ZISC-036 like neuron implements two kinds of distance metrics called L1 and LSUP respectively. Relations (3) give the above-mentioned distance metrics.

$$\text{L1: } dist = \sum_{i=0}^{n} \left| V_i - P_i \right| \qquad \text{and} \qquad \text{LSUP: } dist = \max_{i=0 \ldots n} \left| V_i - P_i \right| \tag{3}$$

Including 36 neurons, ZISC-036 chip is fully cascadable allowing the use of as many neurons as the user needs (several ZISC PC boards). A neuron is able to:

- memorize a prototype (64 components coded on 8 bits), the associated category (14 bits), an influence field (14 bits) and a context (7 bits),
- compute the distance, based on the selected norm (norm L1 or LSUP) between its memorized prototype and the input vector (the distance is coded on14 bits),
- compare the computed distance with the influence fields,
- communicate with other neurons (in order to find the minimum distance, etc.),
- adjust its influence field (during learning phase).

Association of a context to neurons is an interesting concept, which allows a versatile implementation of several neural nets by dividing ZISC-036 neurons in distinguished subsets of neurons, committing only neurons relative to a same context. the network to be divided in several subsets of neurons. Up to 127 contexts can be defined ([10], [11]).

3 Modular Kernel Functions ANN Based Face Recognition System

The facial recognition based mass biometric system we propose includes three main stages. The first one is a video (image flow) acquisition device, which could be a standard digital video camera. The second stage is essentially an image processing stage, which on the one hand, performs a set of image pre-processing operations, and on the other hand, extracts a number of facial biometric features. The last one is a kernel functions based ANN stage carrying out classification and decision operations.

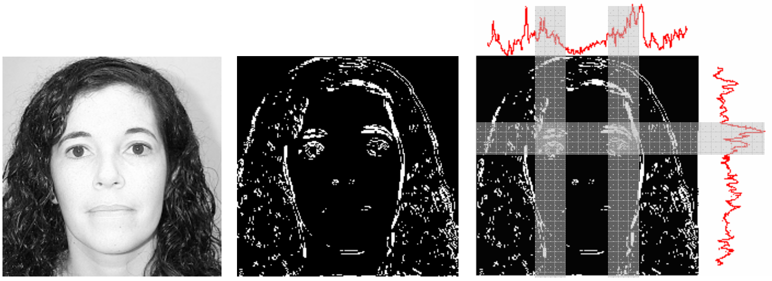
It is pertinent to notice that one of the chief goals in present work was realization of an effectively operational prototype. That is why our implementation choices have been guided by real-time computation capability.

3.1 Image Flow Acquisition

Image flow acquisition could be done by standard or specialized (dedicated) video devices. Most of market available standard video devices offer “face tracking” function which is proposed as associated software option or is implemented as hardware function. On the side of the dedicated (specialized) camera, several products offer sophisticated functions allowing face tracking and a number of face related features extraction or measurements [13].We have preferred to consider a standard Web camera including a face tracking standard function simulating limited resolution and noisy nature of input information conformably to the mass biometry related context and conditions.

3.2 Preprocessing and Features Extraction Stage

In principle, this stage could be a software based stage or a hardware module. However, either software or hardware, the processing should be done in real-time. In our case, the second stage has been realized as a software module on PC. As the face tracking function is performed by the first stage, this second stage has essentially been dedicated to face characteristic areas (mouth, nose and eyes) detection and to the associated biometric features extraction.



(a) **(b)** **(c)**

Fig. 2. Eyes area detection example: input image (a), filtered image (b) and the result (c)

The above-mentioned features' extraction is performed in two steps. During the first step the face image is converted to a binary image and filtered using a conventional Sobel [14] transformation. During the second step, the white pixels spatial histogram in each dimension, representing number of white pixels for all columns (dimension X) and for all rows (dimension Y), is realized [11]. These histograms are used as some kind of signatures to detect mouth, nose and eyes areas.

Fig.2 shows the eyes' area detection using such approach. Detection of each area (mouth, nose and eyes) using the above-described approach takes less than 60 ms remaining compatible with real-time requirements. The obtained images are then used to generate biometric features composing the ANN stage's inputs. The biometric features are a set of 64-pixels images ("8 by 8" or "4 by16"). Different policies could be used to obtain such features [15]. The size (64-pixels) of the images is conditioned by the size of ZISC-036 input vector: a 64 components vector.

3.3 Modular ANN Based Face Recognition System

The third and last stage of the proposed "mass biometry" oriented face recognition system is a kernel functions ANN based stage including three neural nets conceived according to a parallel architecture. Fig. 3 gives the bloc diagram of the whole proposed face recognition system, detailing this third stage.

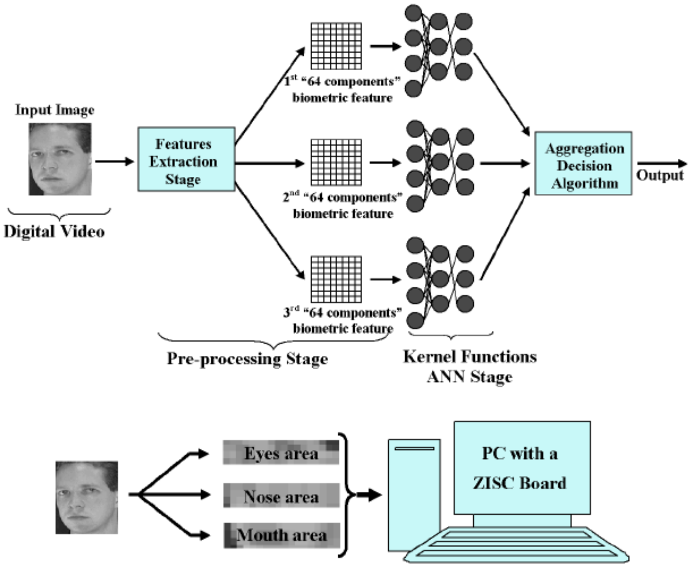

Input Image
Digital Video
Features Extraction Stage
1st "64 components" biometric feature
2nd "64 components" biometric feature
3rd "64 components" biometric feature
Pre-processing Stage
Kernel Functions ANN Stage
Aggregation Decision Algorithm
Output
Eyes area
Nose area
Mouth area
PC with a ZISC Board

Fig. 3. Face recognition system's bloc diagram (upper) and its implementation bloc diagram

Each ANN is specialized in processing of a specific kind of biometric feature extracted from the input image. Then a decision logic based procedure performs (on the basis of classification results relative to each biometric feature) the identification of the concerned individual. The appealing ability of "context" definition in ZISC-036 has been used to implement the three above-mentioned dedicated neural networks.

A first prototype of the proposed "mass biometry" oriented face recognition system has bee, realized using a ZISC-036 based board including 16 chips making available approximately 700 neurons. The second stage has been implemented as software module on PC.

3.4 Experimental Validation

The experimental validation has been done using the ORL (Olivetti Research Laboratory, Cambridge) faces database. This database is composed of 400 images representing 40 different individuals. In other words, the database offers 10 different pictures, for each individual, each one representing a different situation: different mimics, with and without glasses, different degrees of rotation, etc.. An example of images set corresponding to an individual is given in Fig. 4.

Two different kinds of biometric features have been considered. In the first kind, called "global biometric features", the three "64 components" biometric features are "8 by 8" images involving the whole face. The first and second "8 by 8" images are obtained from median operation performed on input image's rows and columns. While, the third one, representing some face morphology related global feature, has been obtained from median operation performed on 64 equal slices of the input image. In the second kind, called "local biometric features", the three "64 components" biometric features are "4 by 16" images involving specific face's areas.

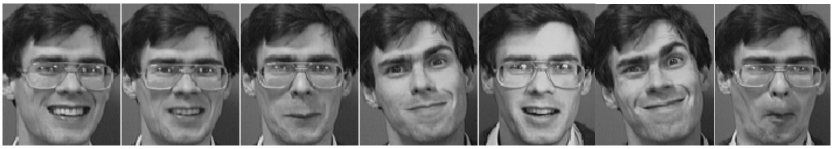


Fig. 4. Example of images set offered by ORL database for a same face (individual)

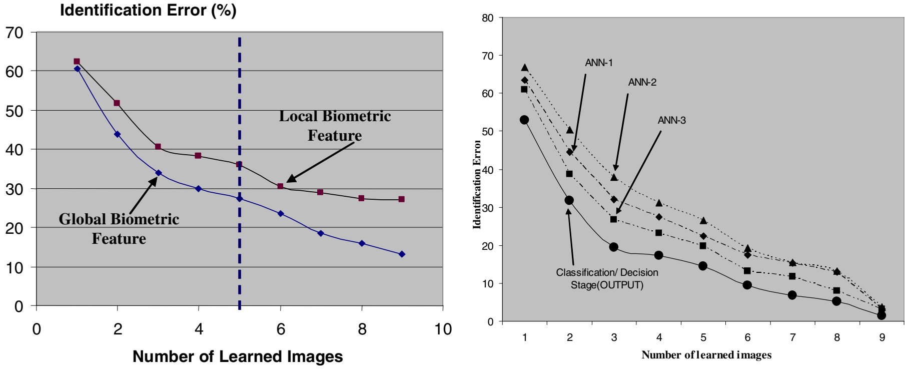


Fig. 5. "Identification error" versus number of learned samples using first fusion based decision logic (left) and using the second decision policy (right)

The first one is obtained from "mouth area", the second fashioned from "nose area" and the last one constructed using the "eyes area". Both of these biometric features take reasonable processing times: less than 100 ms for whole face's image.

Two fusion based decision logic have been implemented. The first (the simplest one) averages the results of the three ANN while the second valorizes the third neural net's response (e.g. the face morphology related global feature and the "eyes area" local feature). Fig. 5 gives identification error versus number of learned samples from the aforementioned database. Reported results correspond to the use of ROI mode with L1 metrics. As it could be observed from the left diagram, using "global biometric feature" based strategy and the first fusion based decision policy a correct identification of 73% is obtained for 5 (among 10) learned examples. While using "local biometric features", the same experimental protocol leads to 64% of correct identification. Also, the "local biometric features" remain less sensible to the effect of additional samples' learning within this decision policy. This shrink of performances could be explained by the effect of the used fusion policy which smoothes the relevance of each local feature in correct typifying of different individuals (faces). Results obtained using "local biometric features" within the second fusion based decision policy confirm this explanation (right diagram of Fig.5).

In fact, identification rate in this case is significantly improved reaching 85% for 5 leaned samples and more than 98% for 9 learned samples. Another interesting point concerns the whole system's "identification error", which remains lower than the error affecting each specialized neural network composing the system, especially when the number of learned samples remains fairly small (3 to 5 samples).

4 Conclusion

Taking into account antagonistic requirements inherent to “mass biometry” context, an ANN based facial recognition biometric system has been proposed and implemented using ZISC architecture and skills. The proposed solution takes advantage at the same time from kernel functions based ANN’s image processing ability and from the massively parallel architecture of ZISC-036 which allows very high processing speed. A first prototype of the proposed system has been realized combining software (biometric features extraction stage) and hardware (ZISC-036 neuro-processor’s based “Classification / Decision” stage) modules. The obtained very promising results show feasibility and effectiveness of the proposed solution. These promising results open a number of auspicious perspectives concerning as well the proposed solution as the “mass biometry” related applications in general. We are working now on two directions. On the one hand we are investigating new “64 components biometric features” (representations), and on the other hand we develop more sophisticated learning strategies on ZISC-036.

References

1. Faundez-Zanuy, M., Fabregas, J.: Testing report of a fingerprint-based door-opening system. IEEE Aerospace and Electronic Systems Magazine 20(6), 18–20 (2005)
2. Samborska, A.: Feature Space Reduction and Classification in Automatic Voice Quality Estimation. Image processing and Communication Journal, IP&C Journal (2006)
3. Carlos, M., Travieso-González, J.B., Alonso, S., David, M.A.: Ferrer-Ballester Optimization of a biometric system identification by hand geometry. In: Complex systems intelligence and modern technological applications, Cherbourg, France, pp. 581–586 (2004)
4. Valchuk, T., Wyrzykowski, R., Kompanets, L.: Mental Characteristics of Person as Basic Biometrics. In: Tistarelli, M., Bigun, J., Jain, A.K. (eds.) ECCV 2002. LNCS, vol. 2359, pp. 78–90. Springer, Heidelberg (2002)
5. Kompanets, L.: Counterterrorism-Oriented Psychology and Biometrics Techniques Based on Brain Asymmetry, Eyes “Fingerprints”, Face Asymmetry and Person Psyche. In: Proc. of SCI 2003, Orlando, Florida, USA, July 2003, pp. 18–21 (2003)
6. Faundez-Zanuy, M.: Data fusion in biometrics. IEEE Aerospace and Electronic Systems Magazine 20(1), 34–38 (2005)
7. Reyneri, L.M.: Weighted Radial Basis Functions for Improved Pattern Recognition and Signal Processing. Neural Processing Let. 2(3), 2–6 (1995)
8. Haykin, S.: Neural nets, 2nd edn. A comprehensive foundation. Prentice Hall, Englewood Cliffs (1999)
9. Handbook of Brain Theory and Neural Networks, 2nd edn. MIT Press, Cambridge (2003)
10. ZISC/ISA ACCELERATOR card for PC, User Manual, IBM France (February 1995)
11. De Tremiolles, G.: Contribution to the theoretical study of neuro-mimetic models and to their experimental validation: a panel of industrial applications., Ph.D. Report, University of PARIS XII (in French) (March 1998)
12. Madani, K., De Tremiolles, G., Tanhoff, P.: Image processing using RBF like neural networks: A ZISC-036 based fully parallel implementation solving real world and real complexity industrial problems. Applied Intelligence (18), 195–231 (2003)
13. `http://seeingmachines.com`
14. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall, Englewood Cliffs (2002)
15. Zhao, W., Chellapa, R., Rozenfeld, A., Phillips, P.J.: Face recognition: A Literature Survey, Tech. Report, Univ. of Maryland (2003), `http://www.cfar.umd.edu/ftp/TrsfaceSurvey.ps.gz`

Thresholded Neural Networks for Sensitive Industrial Classification Tasks

Marco Vannucci, Valentina Colla, Mirko Sgarbi, and Orlando Toscanelli

Scuola Superiore S.Anna, PERCRO lab.
Viale Rinaldo Piaggio, 56025 Pontedera, Italy
`{mvannucci,colla,sgarbi,tojfl}@sssup.it`
`http://www.sssup.it`

Abstract. In this paper a novel classification method for real world classification tasks is proposed. The method was designed to overcome the difficulties encountered by traditional methods when coping with those real world problems where the key issue is the detection of particular situations - such as for instance machine faults or anomalies - which in some frameworks are hard to be recognized due to some interacting factors that are analyzed within the paper. The method is described and tested on two industrial problems, which show the goodness of the proposed approach and encourage its use in the industrial environments.

Keywords: Classification, Industrial databases.

1 Introduction

In many real world industrial problems which are focused on the detection of particular situations, some classification problem can arise. For instance within industrial manufacturing process, to detect as soon as possible machine faults or defective products is a key issue which allows to save time and money. It is clear that, in such cases, the correct identification of these situations is more important than the identification of other situations and, on the other hand, the misclassification of pattern related to not anomalous situations (the so–calles *false alarms*) is more tolerable than the opposite error. Moreover in the industrial framework this kind of events are often rarer and not simply discernible from the others ones, therefore the identification task is quite complex. The inherent difficulty of the classification task combined with the different importance of class recognition are at the basis of the development of the proposed method which is based on a thresholded neural network. This method aims at the identification of pattern corresponding to arbitrary situations which are interesting for the specific applications.

The paper is organized as follows: in section 2 the method is described together with a general analysis of the problem of the recognition of rare and sensitive situations; in section 3 and 4 the method is tested on two industrial problems related to metal industry: the first one concerns defect identification on the surface of metal sheets while the latter one refers to fault diagnosis; finally conclusion and future developments are discussed in section 5.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1320–1327, 2009.
© Springer-Verlag Berlin Heidelberg 2009

2 The Proposed Method

The proposed method aims at the detection of noticeable events related to sensitive problems which are hard to correctly classify. Typical examples within the industrial field are fault diagnosis and defect recognition. In these classification problems the difficulty can be due to several factors: the overall low number of samples to be used for training the classifier; a relative low number of samples corresponding to the interesting situation; the non–separability of the classes within the dataset. In many practical cases these factors are combined so that their effect on the learning process is highly detrimental. In facts traditional learning algorithms, when coping with classification tasks which present these characteristics tend to ignore the less represented classes as the classifiers are designed in order to optimize the overall performance without taking into account the relative distributions of each class [4]. This behavior is not desirable in a context where the main purpose is to correctly identify the less numerous situations. In order to overcome the drawbacks encountered by traditional methods, a novel approach based on a thresholded neural network is proposed. This method will be suitable for all those classification problems where: 1) the main aim of the classifier is the detection of pattern belonging to a particular class despite of the misclassification of patterns belonging to other classes (*false alarms*); 2) the dataset used for the training of the classifier is not separable.

The above listed features are frequently encountered when coping with real industrial problems and decrease the performances of standard methods. The main problem of standard algorithms when coping with data characterized by these features is that, once trained, they tend to output values corresponding to the most represented class. In this paper only classification problems where two possible classes are present will be considered. For the sake of simplicity, and without modifying the generality of the treated problem, the two classes will be labeled as follows: *class-0* will represent the class corresponding to the more frequent instances, while *class-1* will be correspond to the situations whose detection is fundamental. In this context both standard decision trees [9] and standard multilayer perceptron feed–forward neural network (MLP-FFNN) [8] fail in the detection of *class-1* patterns.

In classification tasks, a M-outputs network is normally used where each output neuron is associated to a possible class: when an input pattern is presented the class associated to the output neuron returning the highest value is selected [1]. Another approach involving MLP-FFNN consists in the use of a network with a single output neuron: low output values are associated with *class-0* and high output values correspond to *class-1*. The basic idea exploited by the proposed method is to increase the sensitivity of this latter MLP-FFNN architecture to the salient patterns, so as to rise a *class-1* classification as soon as the presented input pattern is suspected to belong to that class. This idea is implemented by means of the use of a threshold operator at the output of a standard MLP-FFNN and to tune the threshold for the rounding in order to maximize the performance of the classifier on the basis of the specific problem. The general structure of the thresholded artificial neural network (TANN) is depicted in figure 1. The input of the threshold operator will be the output of the feed-forward neural network while its output will be *class-1*, if the processed value is greater than the fixed threshold, *class-0* otherwise. This adjustment will have the effect of increasing the number of *class-1* classifications for all those input pattern whose FFNN output will be higher *enough* to suppose that they belong to *class-1*.

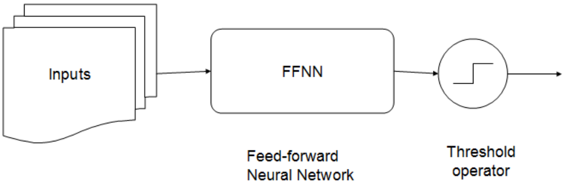


Fig. 1. A schematic representation of the architecture of the proposed method

The training algorithm of the TANN is divided into two subsequent steps: the first one aims at the training of the neural network sub-system while the second one selects the optimal threshold value. To this purposes, the training set is divided into two parts: the first parts includes the 60% of the training set (*AnnTrainSet*) while the remaining samples are used for the threshold determination (*ThrTrainSet*). The exploited neural network is a standard two layers feed-forward network with one output neuron which feeds the threshold operator. The algorithm employed for the training of the ANN is a variation including Bayesian regularization of the Levenberg-Marquardt [5] training algorithm which grants stability and generalization capabilities to the network [2] [3] . Once this part of the training is complete, the threshold value has to be determined. At this stage the output returned by the neural network when a new input pattern is presented is a real number in the range $[0;1]$ and the distribution of such output will be heavily unbalanced in favor of an higher number of values close to zero . The optimal threshold is determined by means of the evaluation of the performance of the proposed structure for a set of candidate thresholds according to the following criteria:

Correct classifications: the overall rate of correct classifications

Detections: the rate of detected important situations by respect to their number

False alarms: the rate of false alarms risen (predicted *class-1* but actual *class-0* observation)

A global figure of merit involving these criteria is calculated for each candidate threshold t by means of equation 1 where *Det* represents the rate of main events detected, *FA* the rate of false alarms and *Corr* the overall rate of correct classifications. In the equation γ and μ are two empirical parameters. Equation 1, whose trend is shown in figure 2, encourages the detection of the salient events

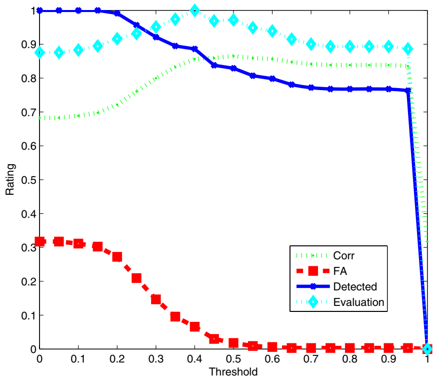


Fig. 2. In this picture the trend of the evaluation function is shown by respect to the others performance measures involved in its calculation

rather than the correctness of the overall classification and, at the same time, aims at avoiding false alarms.

$$evaluation(t) = \frac{\gamma Det(t) - FA(t)}{Corr(t) + \mu FA(t)} \tag{1}$$

The evaluation of the candidate thresholds (whose set is identified by T in equation 2) is separately performed for the two distinct training sets *AnnTrainSet* and *ThrTrainSet* according to equation 2, by selecting for each set the threshold which obtains the highest evaluation. The optimal threshold is then calculated by averaging the thresholds selected for the two groups (namely Th_{ann} and Th_{thr}); such double calculation is done in order to increase the generalization capabilities of the classifier, as the calculation of the final threshold on the basis of a set of observation unknown to the ANN (namely the set *ThrTrainSet*) simulates the real operation of the TANN and improves its performance on the validation set. Once the final threshold has been established, the TANN is completely trained.

$$Th_{ann,thr} = argmax_{t \in T} \left(evaluation(t) \right) \tag{2}$$

3 Quality Control of Metal Sheets

During the production of metal sheets some surface controls on the products are necessary in order to assess the product quality and avoid the presence of defects. In particular the surface of the sheets is inspected by means of a vision system which is able to detect several defects (such as scratches, patchiness...). Once a map collecting all the information on the observed defects is compiled by the vision system, human operators must decide, on the basis of the characteristics of the reported defects, whether to discard the examined sheet or to accept it. The decision on the sheet is taken by globally considering the number of observed defects and, for each defect, some features such as its type, extension, position and shape. Currently the decision is taken in two stages by two human operators: the first one examines the defects map, signals to the subsequent operator those sheets which are suspected to be defective and classifies the others as non defective. The second operator examines the signaled sheets and takes the final decision on the basis of both the defect map and some further indications provided by the first operator. This process is very time consuming due to the high number of sheets processed and suspected to be defective which need a second control. The accomplishment of this latter control needs a considerable amount of time as the examination pursued on the sheets is very accurate. Moreover the decision is left to human operators and can lack of objectivity as the performance of the operator can vary with time. The TANN method was used in this framework in order to support the decision of the second human operator, as depicted in figure 3.

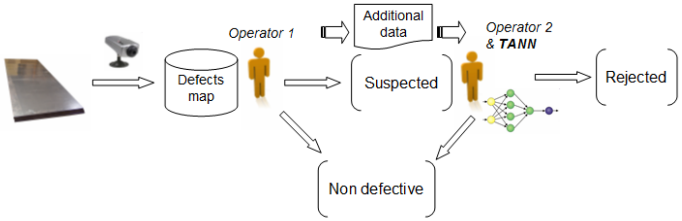


Fig. 3. Diagram describing the metal sheet production line highlighting the integrated TANN system

The described problem is suitable to be faced by the TANN method as the rate of defective sheets is significantly lower than non–defectives and the key issue of the task is the detection of defective products because it is important not to put into the market products containing surface defects while, on the other hand, false alarms (i.e. to signal as defective a sheet which is non defective) are not so detrimental. The exploited TANN system was thus used to identify defective sheets among those signaled by the first human operator. The neural network takes as input both the map provided by the vision system and the indication provided by the first operator. A subset of defect types which mostly affect the final product quality have been taken into account and an analysis of their presence on the examined sheet, together with some sheet features and the report of the first operator, have been exploited to form the input vector of the TANN. Such vector includes both real and discrete variables concerning the gravity of the defects signaled by human operators and some features for the selected defects (such defect width, position, area and shape) that are extrapolated from the maps which are provided by the vision system. The TANN exploits a two layers perceptrons feed-forward neural network with 13 inputs, 8 hidden units and one output. It was trained by using about 9000 observation provided by the metal sheet producer. The 20% of these observations are related to devective sheets. Once the system was trained, a new dataset composed by about 2000 observations (20% defective) was used for the validation. The same training and test data have been exploited in order to apply other methods and make a comparison of the different approaches. The tested methods are: decision trees based on the C4.5 training algorithm [11] [10], support vector machines (SVM) [6] [7] and feed-forward neural networks. The results obtained by these methods are shown in table 1.

Table 1. Results obtained by the tested methods on the metal sheet problem. For each method several configurations were tested but only the results of the best performing one are shown.

Method	Correct	Detected	False alarms
Decision tree	83%	68%	7%
FFNN	72%	29%	4%
SVM	87%	65%	1.5%
TANN	83%	77%	3%

The results confirm the goodness of the proposed methods, in facts, the TANN is able to detect a high number of defective products (77%). The rate of detected defective products is higher than those obtained by the other methods: SVM and decision trees detect less than 70% of these cases while plain FFNN only the 29%. On the other hand TANN rises a reasonable rate of false alarms (3%). The overall performance of the proposed method is satisfactory and TANN significantly overcomes the other methods, also according to the indication of manufacturers. The integration of the TANN in the production line could sensibly improve the performance of the quality control process in terms of speed and reliability.

4 Fault Detection in the Casting Process

In common steel making practice the liquid material produced in the blast furnace is cast, after some manufacture, into the ladle and, subsequently, into the tundish. On the bottom of the tundish, the submerged entry of some nozzles is located, through which the liquid steel passes into the mould or strip casters. The section of such nozzles is far smaller with respect to the tundish dimensions and internal surface of nozzles can be covered by precipitates which slow or even completely block the flow of the liquid steel. This phenomenon is commonly known as *clogging* and is highly detrimental to casting reliability and quality of the cast products. The clogging phenomenon is still not deeply understood [14] due to the very high number of chemical and process factors affecting its formation thus many models have been developed aiming at its early detection. The detection of clogging is important as it allows the early use of appropriate countermeasures that can avoid material waste and can increase of product quality. The TANN methods and several other algorithms have been tested on this problem. The dataset available for the training and the test of the classifiers is made up of about 4000 observations (3% of them corresponding to clogging situations) which include information both on the chemical properties of the produced steel and on the process and machine parameters. The tested method, whose results are shown in table 2, are: FFNN, Learning Vector Quantization (LVQ) [13], Rectangular Basis Function Networks (RecBF) [12], SVM and decision trees.

Table 2. Table summarizing the results obtained by the tested methods on the clogging detection problem

Method	Correct	Detected	False alarms
Decision tree	98%	17%	2%
FFNN	98%	17%	2%
LVQ	97%	0%	0%
RecBF	72%	44%	27%
SVM	97%	0%	0%
TANN	77%	67%	21%

Table 2 shows that the detection of clogging is difficult for all the tested methods. This behavior is probably due to the low number of data corresponding to such event with respect to normal ones. Nonetheless the TANN method is able to detect a satisfactory number of occlusions rising 20% of false alarms. The performance of TANN is satisfactory by considering that false alarms are not detrimental for the production process and product quality. The other methods, except RecBF which seem to recognize a certain rate of cloggings, tend to classify all the patterns as related to the *normal situation* as they aim to maximize the overall number of correctly classified patterns. Also in this case the integration of the TANN in the process could sensibly improve the prevention of harmful situations.

5 Conclusions and Future Work

In this paper a new method for classification tasks within sensitive problems is presented. This method is suitable for those situations where it is particularly important to detect particular events by respect to others - such as, for instance, the machine faults and product defects in the industrial framework. In real world applications this kind of problems are often hard to solve, especially due to the effect of many factors. Traditional methods do not achieve satisfactory results when facing those kind of problems. The proposed method, that is based on a thresholded neural network, was tested on two real world problems and obtained satisfactory results, by overcoming the criticalities encountered by other methods. In particular TANN is able to detect the notable situations by generating a reasonable number of false alarms. The obtained results encourage the integration of the proposed approach in the examined production processes. Future work will concern a refinement of the algorithm for the threshold determination as well as further tests on case studies extracted from industrial fields.

References

1. Zhang: Neural networks for classification: a survey. IEEE trans. on Systems, man and cybernetics. Part C 30(4) (November 2000)
2. MacKay: Neural Computation 4(3), 415–447 (1992)
3. Foresee, H.: In: Proceedings of the International Joint Conference on Neural Networks (June 1997)
4. Estabrooks.: A combination scheme for inductive learning from imbalanced datasets. MCS thesis, Faculty of computer science, Dalhousie university
5. Marquardt: Analgorithm for least square estimation of non linear parameters. SIAM Journal of applied mathematics 11, 164–168
6. Boser, Guyon, Vapnik: A training algorithm for optimal marginclassifiers. In: Fifth annual workshop on computational learning theory, San Mateo, CA (1992)
7. Vapnik: Statistical learning theory. Wiley, Chichester (1998)
8. Haykin: Neural Networks: A Comprehensive Foundations. Mc Millan, New York (1994)
9. Breiman, F., Olshen, S.: Classification and regression trees. Wadsworth, Belmont (1984)
10. Quinlan: C4.5: programs for machine learning. Morgan Kaufman, San Francisco (1993)
11. Breiman, et al.: Classification and Regression Trees. Chapman and Hall, Boca Raton (1993)
12. Soler, V., Prim, M.: Rectangular basis functions applied to imbalanced datasets. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 511–519. Springer, Heidelberg (2007)
13. Kohonen: Improved versions of learning vector quantization. In: Proc.of the International Joint Conference on Neural Networks, pp. 545–550 (June 1990)
14. Heesom: Physical and chemical aspects of nozzle blockage during continuous casting. In: Proc. of 1st Int. Calcium Treatment Sym., London (1988)

ANN Based Solutions: It Is Time to Defeat Real-World and Industrial Dilemmas

Kurosh Madani, Véronique Amarger, and Christophe Sabourin

Images, Signals and Intelligence Systems Laboratory (LISSI / EA 3956), PARIS 12 – Val de Marne University, Senart-Fontainebleau Institute of Technology, Bât.A, Av. Pierre Point, F-77127 Lieusaint, France
madani@univ-paris12.fr

Abstract. Over past decades, Artificial Neural Network (ANN) area has been the focal point of an ever-increasing number of research works and a very active pivot of interdisciplinary research activity. It is now time to state if ANN are ready to defeat nowadays' real-world and industrial challenges. The main goal of this paper is to present, through some of main ANN models and based techniques, their capability in real world industrial dilemmas solution. Examples of real world and industrial applications have been presented and discussed.

Keywords: Artificial Neural Networks, Real-Time, Real-world, Industrial, Solutions, Implementation.

1 Introduction and Context Stating

Real world dilemmas, and especially industry related ones, are set apart from academic ones from several basic points of views. The difference appears since definition of the "problem's solution" notion. In fact, academic (called also sometime theoretical) approach often begins by problem's constraints simplification in order to obtain a "solvable" model (here, solvable model means a set of mathematically solvable relations or equations describing a behavior, phenomena, etc…) [1]. If the theoretical consideration is a mandatory step to study a given problem's solvability, for a very large number of real world dilemmas, it doesn't lead to a solvable or realistic solution. Examples illustrating the above-mentioned fact are numerous and may concern various areas. As first example, one can emphasize difficulties related to industrial applications dealing with system identification, industrial processes control, systems and plants safety, manufacturing regulation and optimization. Another illustrative example concerns the delicate class of dilemmas dealing with economical and financial modeling and prediction, where the large number of parameters, on the one hand, and human related factors, on the other hand, make related real world problems among the most difficult to solve.

Overcoming a number of conventional approaches' limitations thank to their learning and generalization capabilities, Artificial Neural Networks (ANN) made appear a number of expectations to design "intelligent" information processing systems. They have been, over the past decades, central stream of elaboration of many

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1328–1335, 2009.
© Springer-Verlag Berlin Heidelberg 2009

original techniques covering a large field of applications ([2] to [4]). If much is still to discover about how the animal's brain trains and self-organizes itself in order to process so various and so complex information, a number of recent advances in "neurobiology" allow already highlighting some of key mechanisms of this marvels machine and could already be sources of inspiration for designing new approaches defeating the above-mentioned difficulties.

The main goal of this paper is to present, through main ANN models and based techniques, the effectiveness of such approaches in real-world and industrial problems' solution. Two examples real-size and real-complexity applications have been reported and discussed. The paper has been organized as follows: after a brief overview of industrial applications' specificities, the next section will present two ANN based applications. One of them will deal with an industrial problem and the other with a real-world challenging purpose. Finally, the last section will conclude the paper.

2 ANN Serving Real-World and Industrial Applications

Several specificities distinguish the industrial world and related constraints from the others. In the context of the present paper, the word "specificity" intends characteristic or criterion channelling industrial preference as an alternative to the others. As a first specificity one could mention the "reproducibility". That means that an industrial solution (process, product, etc…) should be stable. A second specificity is "viability", which means implementation (realization) possibility. That signifies that an industrial solution should be adequate to available technology and achievable in reasonable delay. Another criterion is "saleability", which means that an industrial solution should wrap a well identified field of needs. Finally, an important specificity is "marketability" making a proposed solution attractive and concurrent (regarding price-quality ratio, etc…) to other products (or solutions) concerning the same area.

Moreover, it is not always possible to put away the lower degree phenomena's influence or to neglect secondary parameters when dealing with real world environment and related realities. That's why a well known solved academic problem could appear as an unachieved (unbearable) solution in the case of an industry related dilemma. In the same way a viable and marketable industrial solution may appear as primitive from academic point of view. The above-listed specificities have guide the spirit of reported applications.

2.1 Intelligent Automated Defects' Detection and Diagnosis in High-Tech Optical Devices' Production

Fault diagnosis in industrial environment is a challenging but crucial task, because it ensures products' nominal specification as well as the manufacturing processes control. Concerning High-Tech optical industry, a major step for high-quality optical devices' faults diagnosis concerns scratches and digs defects detection and characterization in such products. These kinds of aesthetic flaws, shaped during different manufacturing steps, could provoke harmful effects on optical devices' functional specificities, as well as on their optical performances by generating undesirable scatter light, which could seriously damage the expected optical features. A reliable diagnosis of these defects becomes therefore a crucial task to ensure products' nominal (expected) specification. Moreover, such diagnosis is strongly motivated by manufacturing process correction requirements in order to guarantee mass production quality with the aim of maintaining acceptable production yield.

The diagnosis is performed on the basis of a human expert based visual inspection of the whole production. However, this conventionally used solution suffers from several acute restrictions related to human operator's intrinsic limitations. The main limitations are: reduced sensitivity for very small defects, detection exhaustiveness alteration due to attentiveness shrinkage, operator's tiredness and weariness due to repetitive nature of fault detection and fault diagnosis tasks. Figure 1 gives an example of High-Tech optical product, showing four optical filters. The same figure shows an example of visual inspection process of the aforementioned defects requiring expert knowledge and leading to a consequent inspection delay ([5] and [6]).

To construct an automatic diagnosis system, the proposed approach has been based on three main operations: detection, classification and decision. The motivation to dissociate detection and diagnosis is based on industrial production's specificities. In fact, in the case of mass production it is not always necessary to diagnose whole manufactured products, but it is crucial to detect the presence of defects in order to state if the number of defects is conform to the process' intrinsic limitations. On the other hand, for High-Tech optical products, additionally to systematic defects' detection it is crucial to state on nature of detected defects in order to reach high-quality specifications. Fig. 2 gives the complete bloc diagram of the proposed solution. Four stages compose the automated diagnosis chain: "defect detection", "defects' isolation" then defects' "features extraction" and "classification" [6].

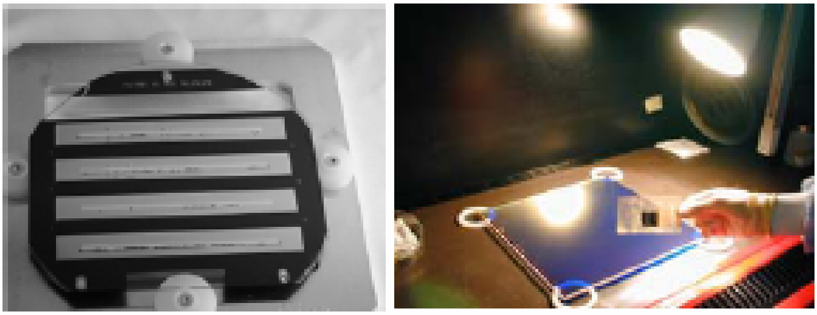


Fig. 1. Example of High-Tech optical devices performing optical filtering (left) and the visual fault detection, performed by an expert (right)

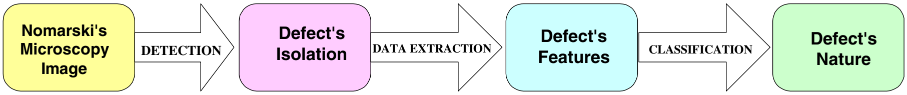


Fig. 2. Block diagram of the proposed defect diagnosis system

The aim of defect's detection stage is to find defects from "Differential Interference Contrast" (DIC) microscopy based detector' issued digital images. The proposed method [5] includes four phases:

- Pre-processing: DIC issued digital image transformation in order to reduce lighting heterogeneity influence and to enhance the aimed defects' visibility,
- Adaptive matching: adaptive process to match defects,
- Filtering and segmentation: noise removal and defects' outlines characterization.
- Defect image extraction: complete defect's images' construction.

The defects' isolation stage sets sights on defects' isolation. In other words, it generates an image for each detected defect isolating it from other items (e.g. depicts the defect in its immediate environment). Figure 3.gives examples of isolated defects after their DIC microscopy based detection. This raw data (isolated defects' images) can not be directly processed and has first to be appropriately encoded, using some transformations. Fourier-Mellin transformation is used as it provides invariant descriptors, which are considered to have good coding capacity in classification tasks [7]. Finally, the obtained features are normalized, using the centring-reducing transformation, providing a set of 13 components vectors [6].

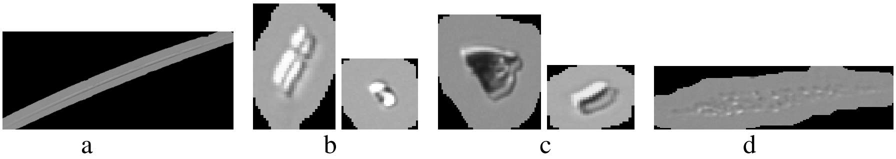


Fig. 3. Images of characteristic items: a) scratch; b) dig; c) dust; d) cleaning marks

In order to reduce the problem dimensionality (e.g. the processing's complexity), we use Curvilinear Distance Analysis (CDA) [8]. This technique is related to Curvilinear Component Analysis (CCA), which aims to reproduce the topology of an n-dimension original space in a new p-dimension space (where $p<n$) without fixing any configuration of the topology [9]. To do so, a criterion characterizing the differences between original and projected space topologies is processed. Finally, the classification stage is a MLP based classifier.

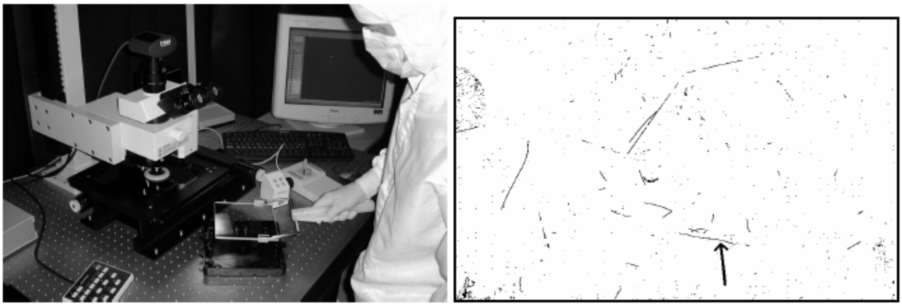


Fig. 4. Industrial system's prototype (left) and Cartography of a 100mm x 65mm optical device

Table 1. Description of the two databases used for validation experiments

Database	Optical Device	Number of microscopic fields	Corresponding area	Total items number	Class 1 items number	Class -1 items number
A	1	1178	28 cm²	3865	275	3590
B	2	605	14 cm²	1910	184	1726

An industrial prototype performing automatic control has been implemented. It involves an Olympus B52 microscope combined with a Corvus stage, which allows scanning an entire optical component. The proposed image processing method is applied on-line. Figure 4 shows the realized industrial prototype and an example of defects' detection (and visualization).

In order to validate the above-presented concepts and the industrial prototype two experiments called (that we will call A and B) were carried out, using two different optical devices. Table 1 shows the different parameters corresponding to these experiments. Using this experimental set-up, classification experiment was performed. It involved a multilayer Perceptron with n input neurons, 35 neurons in hidden layer, and 2 output neurons (n-35-2 MLP topology). First this artificial neural network was trained for discrimination task between classes 1 and -1, using database B. This training phase used BFGS (Broyden, Fletcher, Goldfarb, and Shanno) with Bayesian regularization algorithm, and was repeated 5 times. Subsequently, the generalization ability of the proposed neural network was tested using database A. Since databases A and B are issued from different optical devices, the obtained generalization results are significant. Correct classification rate reaches 97% with 0.87% (e.g. less than 1%) standard deviation. The obtained test results show viability of the implemented solution.

2.2 Modular Neuro-Fuzzy Adaptive Controller for Biped Walking Robots

One of the most challenging topics, over the recent decades, in the field of robotics concerned the design and the control of biped robots. Several potentialities make this foremost research area particularly appealing in the frame of middle and long term projection. On the fundamental side, advances in this research area can lead to a better comprehension of the human locomotion mechanisms. From the applicative point of view, it could concern a wide spectrum of applications among which: the design of more efficient prosthesis and the construction of more sophisticated (human-like) humanoid robots for interventions in hostile environments.

Two main control strategies are generally used in the field of biped robots' locomotion: one is based on a kinematics and dynamic modeling of the whole robot's mechanical structure, and another takes advantage from soft-computing techniques (fuzzy logic, neural networks, genetic algorithm, etc…) and heuristically established rules resulting from the expertise of the human's walking.

Investigating soft-computing based fully autonomous biped robot's walking, we proposed a new approach taking advantage simultaneously from local and global generalization. Our approach [10] is based on a modular Fuzzy-CMAC architecture: a set of CMAC ANN ([10] and [11]) based modules and a fusion stage. The fusion is carried out by using Takagi-Sugeno FIS (Fuzzy Inference System). The main task of Fuzzy-CMAC based modular part of the system is to compute the swing leg's trajectory (using a Fuzzy Inference System fusion of several CMAC neural networks' outputs). The proposed control strategy allows regulating the average velocity from a modification of the desired pitch angle at each new step. Fig. 5 gives the bloc diagram of the proposed hybrid control architecture (the left figure) and shows the training strategy (right bloc diagram).

The trajectories of the swing leg (in terms of joint positions and velocities) are learned by four "single-input/single-output" $CMAC_k$ with k= 1,..,4 neural networks (four trajectories to learn). The learned trajectories are joint angles q_{i1} and q_{i2}, and the two corresponding angular velocities $\dot{q}_{i1}$ and $\dot{q}_{i2}$. q_{i1} and q_{i2} are respectively the measured angles at the hip and the knee of the leg i. In the same way, $\dot{q}_{i1}$ and $\dot{q}_{i2}$

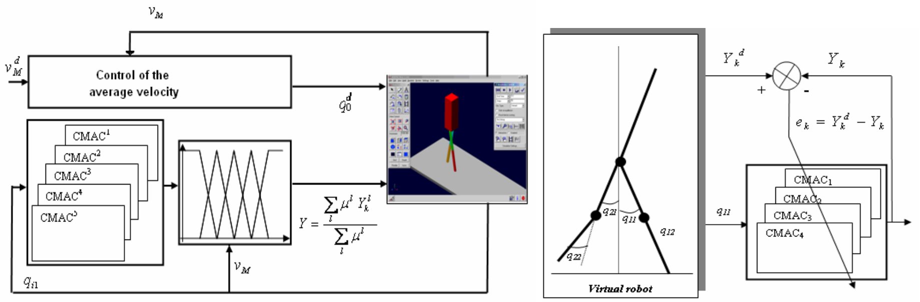


Fig. 5. Bloc- diagram of the proposed modular Neuro-Fuzzy adaptive control strategy (left) and learning strategy principle's bloc diagram (right)

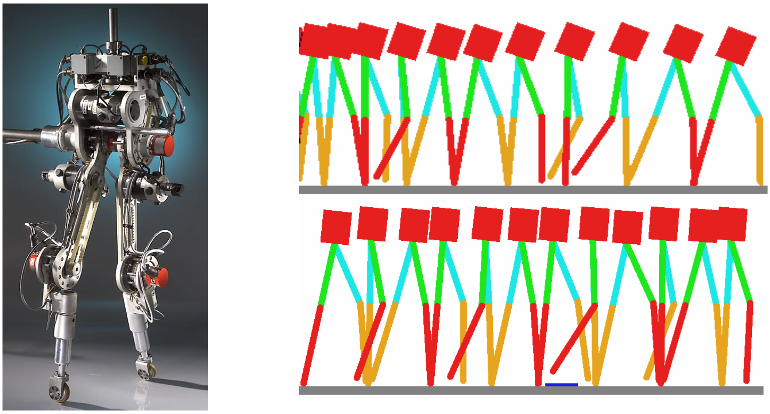


Fig. 6. RABBIT biped robot (left) and robot's walking stick diagrams corresponding to "velocity increasing" case (right-upper) and to "obstacle crossing" case (right-lower)

are respectively the measured angular velocities at the hip and the knee of the leg i (see figure 5). During the training stage, five trajectories corresponding to five different average velocity values (V_M measured in m/s) included in [0.4, 0.8] interval are learned by five CMAC based modules. Each module (labelled $CMAC^l$, with $l \in \{1, 2, 3, 4, 5\}$) includes four $CMAC_k$ neural networks (corresponding to the four above-mentioned robot's trajectories). V_M is computed by using relation (2) where L_{step} is the distance between the two feet at the moment of double impact and t_{step} is the duration of the step (from takeoff to landing of the same leg).

$$V_M = \frac{L_{step}}{t_{step}} \tag{1}$$

The Fuzzy Inference System is obtained from the five following rules, where Y^l corresponds to the output of $CMAC^l$ with $l \in \{1, 2, 3, 4, 5\}$:

- IF V_M IS Very Small THEN $Y = Y^1$
- IF V_M IS Small THEN $Y = Y^2$
- IF V_M IS Medium THEN $Y = Y^3$
- IF V_M IS Big THEN $Y = Y^4$
- IF V_M IS Very Big THEN $Y = Y^5$

The validation of proposed approach has been done using skills of an under-actuated robot: RABBIT ([12] and [13]). This robot has been the central point of a project, within the framework of CNRS (Centre Nationale de la Recherche Scientifique) [13]. This robot is composed of two legs and a trunk and has no foot as shown on Fig. 6 (left picture). A numerical model of the previously described robot has been implemented within the ADAMS software [14]. This software allows to simulate RABBIT's dynamic behavior and namely to calculate the absolute motions of the platform and the relative motions of the limbs when torques are applied on the joints by the virtual actuators. The right picture of Fig.6 shows the stick diagram of the biped robot's walking sequence when the desired average velocity increases. The same figure gives also the stick diagram corresponding to robot's walking sequence avoiding obstacle. The main interest of this modular approach is to proffer to the walking robot autonomy and robustness. The obtained results show the adaptability of the walking step length. Furthermore, the modular Fuzzy-CMAC approach allows decreasing the memory size in comparison to the traditional multi-input CMAC ANN.

3 Conclusion

The main goal of the present paper was focused on ANN based techniques and their application to solve real-world and industrial problems. The reported examples (works) show that today, conjunction of ANN' learning and generalization ability with recent computational technologies offers attractive potential for design and implementation of real-time intelligent industrial solutions. If this paper doesn't pretend to give an exhaustive state of art concerning the huge potential offered by such approaches, it reports, through above-presented ANN based realizations, a plenty idea of promising capabilities of ANN based solutions to solve difficult nowadays' industrial challenges.

Acknowledgements. The research work reported in section 2.1 has been a joint research project involving SAGEM Group and LISSI laboratory. Authors wish thank SAGEM Group and especially Mr. Joel Bernier from this industrial group. We would also thank Dr. M. Voiry and Miss W. Yu involved in reported works respectively.

References

1. Madani, K.: On ANN Based Solutions for Real-World Industrial Requirements. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 79–90. Springer, Heidelberg (2004)
2. Egmont-Petersen, M., de Ridder, D., Handels, H.: Image Processing with Neural Networks - A Review. Pattern Recognition 35, 2279–2301 (2002)
3. Reyneri, L.M.: Weighted Radial Basis Functions for Improved Pattern Recognition and Signal Processing. Neural Processing Let. 2(3), 2–6 (1995)
4. Madani, K., DeTremiolles, G., Tanhoff, P.: Image processing using RBF like neural networks: A ZISC-036 based fully parallel implementation solving real world and real complexity industrial problems. J. of Applied Intelligence (18), 195–231 (2003)
5. Voiry, M., Houbre, F., Amarger, V., Madani, K.: Toward Surface Imperfections Diagnosis Using Optical Microscopy Imaging in Industrial Environment. In: IAR & ACD Workshop 2005 Proceedings, pp. 139–144 (2005)
6. Madani, K., Voiry, M., Amarger, V., Kanaoui, N., Chohra, A., Houbre, F.: Computer Aided Diagnosis Using Soft-Computing Techniques and Image's Issued Representation : Application to Biomedical and Industrial Fields. Computing, Special Issue on Intelligent Computing 5(3), 43–53 (2006)
7. Choksuriwong, A., Laurent, H., Emile, B.: Comparison of invariant descriptors for object recognition. In: IEEE International Conference on Image Processing (ICIP), pp. 377–380 (2005)
8. Demartines, Hérault, J.: CCA: Curvilinear Component Analysis. In: Proceedings of 15th workshop GRETSI (1995)
9. Lee, J.A., Lendasse, A., Donckers, N., Verleysen, M.: A Robust Nonlinear Projection Method. In: European Symposium on Artificial Neural Networks - ESANN 2000 (2000)
10. Sabourin, C., Madani, K., Bruneau, O.: "A Fuzzy-CMAC Based Hybrid Intuitive Approach for Biped Robot's Adaptive Dynamic Walking". In: Proc. ICNNAI 2006 Conference Proceedings, Brest, Belarus, May 31 - June 2, pp. 78–83 (2006)
11. Albus, J.S.: A new approach to manipulator control, the Cerebellar Model Articulation Controller (CMAC). Journal of Dynamic Systems, Measurement and Control, 220–227 (1975)
12. `http://robot-rabbit.lag.ensieg.inpg.fr/`
13. `http://www.laas.fr/robea/`
14. `http://www.mscsoftware.com/products/adams.cfm`

Pollution Alarm System in Mexico*

M.G. Cortina-Januchs[1], J.M. Barrón-Adame[2], A. Vega-Corona[2],
and D. Andina[1]

[1] Universidad Politécnica de Madrid, Group for Automation in Signals and Communications, Spain
januchs@salamanca.ugto.mx
[2] Universidad de Guanajuato, Laboratory of Computational Intelligence, Mexico

Abstract. Air pollution is one of the most important environmental problems. The prediction of air pollutant concentrations would allow taking preventive measures such as reducing the pollutant emission to the atmosphere. This paper presents a pollution alarm system used to predict the air pollution concentrations in Salamanca, Mexico. The work focuses on the daily maximum concentration of PM_{10}. A Feed Forward Neural Network has been used to make the prediction. A database used to train the Neural Network corresponds to historical time series of meteorological variables (wind speed, wind direction, temperature and relative humidity) and air pollutant concentrations of PM_{10} along a year. Our experiments with the proposed system show the importance of this set of meteorological variables on the prediction of PM_{10} pollutant concentrations and the neural network efficiency. The performance estimation is determined using the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

1 Introduction

In recent years, the city of Salamanca has been catalogued as one of the most polluted cities in Mexico [1]. The main causes of pollution in this city are fixed emission sources, such as chemical industry and electricity generation, of largest pollutant in air: Sulphur Dioxide (SO_2), measured in Part Per Billion (PPB) and Particulate Matter less than 10 microns in diameter (PM_{10}). Currently, an Automatic Environmental Monitoring Network (AEMN) is established in Salamanca. The AEMN reported five criteria pollutant [2] and seven meteorological variables: Wind Direction (WD), Wind Speed (WS), Temperature (T), Relative Humidity (RH), Atmospheric Pressure(AP), Precipitation (P) and Solar Radiation (SR). This article focuses on forecasting of PM_{10} daily maximum concentration.

---

* This research has been partially supported by National (MICINN) and Madrid (CAM) Spanish institutions under the following projects: Technical University of Madrid AL09-P(I+D)-12 , PTFNN (MCINN ref: AGL2006-12689/AGR); the Computational Intelligence Laboratory (LABINCO) of Guanajuato University, Mexico; Consejo Nacional de Ciencia y Tecnología (CONACyT) of Mexico; Secretaría de Educación Pública (SEP) and Gobierno Mexicano.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 1336–1343, 2009.
© Springer-Verlag Berlin Heidelberg 2009

1.1 Particulate Matter

Particulate Matter (PM) is a complex mixture of airborne particles that differ in size, origin and chemical composition. PM is released from natural and anthropogenic sources, such as volcanic eruptions, soils, car exhausts, industry, and power plants therefore increasing PM's concentrations in many locations. PM has been linked to a range of serious respiratory and cardiovascular health problems [3].

The Environmental Protection Agency (EPA) divide particle pollution in two categories: Coarse Particles that include particles less than 10 microns in diameter (PM_{10}) and Fine Particles referring to particles smaller than 2.5 microns ($PM_{2.5}$). The PM presence and dispersion in the atmosphere depends on a wide rage of factors including meteorological (T, rain, WD, WS, etc) [4]. The key effects associated with exposure to ambient PM include: premature mortality, aggravation of respiratory and cardiovascular disease. If it would be possible to predict high PM concentrations in advance, more efficient actions could be taken in order to protect the population.

1.2 Artificial Neural Network in Pollutant Prediction

Classical statistical methods and Artificial Neural Networks (ANN) have been used by several authors for short terms prediction of gas and PM pollution. Prediction models have been proposed where average pollutant concentrations are given between one hour and 48 hours in advance. Some these models use the pollution data from the past plus some meteorological information, traffic information, day of the week [5].

The Multilayer Perceptron (MLP), were largely applied in last decade for prediction of gas and PM pollution. The MLP have been applied for the prediction of SO_2, O_3 and $PM_{2.5}$. Kukkonen *et al.* [7] used five ANN models, a linear statistical model and deterministic one for the prediction of urban NO_2 and PM_{10} concentrations at two stations in the central Helsinki. Their models used traffic flow and meteorological variables as input data to forecast the two pollutants on a 24-h horizon. The results show that the non-linear ANN models performed slightly better than the deterministic model and linear statistical model.

Pérez *et al.* [8] predicted average $PM_{2.5}$ concentration for up to 24 hours ahead . They compared the forecasting of air quality for fine PM produced by three different methods: a MLP, linear regression and persistence, obtaining the best result using a MLP with three layers. Their experiment was improved taking into account related meteorological variables.

Kurt *et al.* [9] used a Feed Forward Neural Network (FFNN) to predict future (+1, +2 and +3 days) pollutant levels (SO_2, PM_{10} and CO). Their prediction of +2 and +3 days are made cumulatively using previous days predicted values. This model consists of six inputs: general condition and some meteorological variables (WD, AP, T, RH, WS). Experiments presented in their paper show that quite accurate predictions of pollutant levels are possible with a simple ANN.

Brunelli *et al.* [10], presented a recurrent Elman ANN for the prediction, two-days ahead, of daily maximum concentrations of some pollutants (SO_2, O_3, PM_{10}, NO_2 and CO), the city of Palermo, Italy, using a meteorological predictors: WD, WS, AP and T.

Ibarra *et al.* [11] made predictions of up to predicted 8 ahead for five pollutants (SO_2, CO, NO_2, NO and O_3), using MLP, Radial Basis Function (RBF) and Generalized Regression Neural Network. Their results show that in some cases, the GRNN and RBF can perform as well or even better than MLP.

Kumar *et al.* [12] proposed a scheme based on a MLP to predict O_3 concentrations, using an algorithm to search the nearest neighbor in prediction patterns. They obtained that the new scheme provides better prediction than the standard scheme with relatively small prediction error.

Coman *et al.* [13], predicted ozone concentrations for a 24-h horizon in Paris. They proposed two models: a dynamic model used structure that involve a cascade of 24 MLP and with 24 outputs, and static model with a classical MLP with 24 outputs. In the experiments, they used a combination of variables such as: T, NO_2 concentration, RH, SR. Generally both models the generalization was better for the first half of prediction horizon (for the first 8 hours).

2 Methodology

The air quality in cities varies depending on the industrialization, population and traffic density and meteorological and topographical properties of the region [14]. Because of this air quality models require meteorological data to correctly predict air pollutant concentrations. The requisite meteorological inputs can vary by air quality model, but typically involve information such as: wind vectors, vertical mixing, temperature, and atmospheric moisture. In previous papers authors shown WD, WS, T and RH [6,9,10,15] have the most influence on PM_{10} concentration. For this reason our experiment were made with this meteorological variables.

The proposed system is based on a Feed Forward Neural Network (FFNN). After experimenting with some other ANN structures, a FFNN structure with two hidden layer was selected. The FFNN model consists of five inputs (maximum daily concentration of PM_{10}, WS, WD, T and RH), two hidden layers of 20 neurons and 10 neurons respectively (as shown in figure 1), obtaining as the output the next-day maximum concentration. The activation function used for hidden layers' neurons was sigmoid function and for the output neuron the linear function was used. The FFNN with the same structure was used for all our experiments in order to find the time window (number of days) necessary to make the best prediction using meteorological variables. All the mathematical computations were performed using Neural Network Toolbox in Matlab©.

This paper presents two experiments. In the first experiment, we find the optimum training data set size comparing the performance within different time windows (from 1 day to 15 days). In the second experiment, we use the first

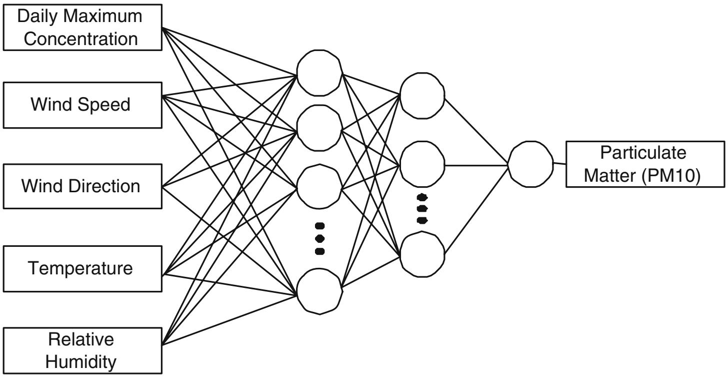


Fig. 1. ANN Model implement to predict the maximum daily concentration of PM_{10}

experiment results to compare the prediction using different meteorological variables in order to show the importance of this variables.

2.1 Database

The data base obtained by the AEMN, represents the essential information to be used for the determination and prediction of environmental situations. Due to the large amount of variability and conditions involved in air pollutants measurements it is necessary to revise and refine the gathered information from the AEMN. The data base used corresponds to the information of the years 2003 and 2004. The validation of data base was done according to the INE manual [16]. Daily maximum of PM_{10} pollutant concentrations, WD, WS, T, and RH were obtained to form the training and test set. The training set was formed with 75% of the available data and the test set consisted of 25% remaining of the data.

In addition, training and test sets were normalized in the range [0 1], using the following transformation:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

where X' is the normalized value, X is the old value, X_{max} and X_{min} are the minimum and the maximum, respectively, of the considered data set.

2.2 Experiment 1

The objective of this experiment is to find the optimum training data set size. In this experiment, the performance for different number of days (from 1 day to 15 days) is compared.

In the experiment we form fifteen training sets named P, each of them containing the maximum daily concentration of PM_{10}, WS, WD, T, and HR. In order to evaluate the performance the training sets are varied from 1 day to 15 days, forming them as follow:

$$P = \{x_{PM_{1\ldots i}}, x_{ws_{1\ldots i}}, x_{wd_{1\ldots i}}, x_{t_{1\ldots i}}, x_{rh_{1\ldots i}}\} \tag{2}$$

where i is the number of days necessary to make the prediction, x_{PM} is PM_{10} concentration, x_{wd} is wind direction , x_{ws} is wind speed, x_t is temperature and x_{rh} is humidity relative, and all values are daily maximum.

2.3 Experiment 2

This experiment was conducted to verify the importance of meteorological variables. The first experiment results were used. We formed five training sets, named P_0, P_1, P_2, P_3 and P_4, their forming as follow:

$$P_0 = \{x_{PM}\} \tag{3}$$

$$P_1 = \{x_{PM}, x_{ws}\} \tag{4}$$

$$P_2 = \{x_{PM}, x_{ws}, x_{wd}\} \tag{5}$$

$$P_3 = \{x_{PM}, x_{ws}, x_{wd}, x_t\} \tag{6}$$

$$P_4 = \{x_{PM}, x_{ws}, x_{wd}, x_t, x_{rh}\} \tag{7}$$

where x_{PM} is PM_{10} concentration, x_{wd} is wind direction, x_{ws} is wind speed, x_t is temperature and x_{rh} is humidity relative, as the first experiment, all values were for PM_{10} concentration daily maximum.

2.4 The Prediction Accuracy

The prediction accuracy was evaluated through two parameters Mean Absolute Error *(MAE)* and Root Mean Square Error *(RMSE)*, defined by the following equations:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|X_i - Y_i| \tag{8}$$

$$RMSE = \left[\frac{1}{N}\sum_{i=1}^{N}(X_i - Y_i)^2\right]^{1/2} \tag{9}$$

where X_i is the observed value at time i, Y_i is the predicted value at time i and N is the total number of observations.

Table 1. Experiment 1: The prediction accuracy for different time windows (from 1 to 15 days)

No. of Days	MAE	RMSE
1	0.249	0.510
2	0.240	0.493
3	0.216	0.327
4	0.108	0.150
5	0.152	0.205
6	0.088	0.117
7	0.219	0.297
8	0.105	0.133
9	0.138	0.174
10	0.131	0.167
11	0.171	0.232
12	0.131	0.160
13	0.145	0.192
14	0.130	0.160
15	0.103	0.134

Table 2. Experiment 2: The prediction accuracy obtained using different meteorological variables

No. of Days	MAE	RMSE
P_0	0.276	0.357
P_1	0.233	0.375
P_2	0.164	0.210
P_3	0.117	0.171
P_4	0.088	0.117

3 Experimental Results

- The first experiment was done using the training sets with meteorological variables (WS, WD, T and RH) and PM_{10} daily maximum concentration changing time span. The table 1 shows the results obtained from the first experiment. The MAE values show that the best perform were obtained with training set that contained from the previous data of six days with MAE=0.088 and RMSE=0.117. The worse results were obtained with the training sets of 1, 2, 3, 7 days.
- In the second experiment the first experiment results were used. An ANN was trained using the 6-day training set. Five new training sets were formed. They included the meteorological variables in order to show the importance of these variables. Table 2 shows the obtained results of the second experiment. In these results we observe that the use of meteorological variables allow us to obtain more prediction accuracy. The worst results were obtained with the training set of PM_{10} concentration.

4 Conclusions

In the case presented, we propose and justify to use four meteorological variables (WD, WS, T, HR) to predict daily maximum concentration of PM_{10} applying a FFNN. The best results were obtained with a time window of six days. In previous works, applying the same ANN, the authors used five meteorological variables (WD, WS, AP, T, RH) to predict the pollutant level, obtaining the best results with a time window of three days.

According to our results, large time windows are necessary in order to predict the PM_{10} concentrations. The advantages of these time windows are the reduction of meteorological variables necessaries to predict the PM_{10} concentrations. Moreover, with the obtained prediction it is possible to directly calculate the pollutant level.

The results show the importance of the meteorological variables to predict the maximum daily concentration of PM_{10}. Meteorological variables permit to obtain more information about the next maximum concentration of PM_{10}. It is important to note that it is necessary to analyze the study zone and variations in climate to obtain the relation between meteorological variables and pollutants concentrations. These experiment can be extended in the future, using other variables such as traffic flow, day of the week, etc.

References

1. Zamarripa, A., Sainez, A.: Medio Ambiente: Caso Salamanca. Instituto de Investigación Legistativa (2007)
2. INE, Instituto Nacional de Ecología, Dirección General de Investigación sobre la Contaminación Urbana y Regional. Investigación sobre la calidad del aire: contaminantes criterio (2008), `http://www.ine.gob.mx`
3. Acken, A.H., Lewis, R.: Final Particulate Matter National Ambient Air Quality Standards. ENRLS update 4(15) (2006)
4. Sousa, S.I.V., Martins, F.G., Pereira, M.C., Alvim-Ferraz, M.C.M., Ribeiro, H., Oliveira, M., Abreu, I.: Influence of atmospheric ozone, PM_{10} and meteorological factors on the concentration of airborne pollen and fungal spores. Atmospheric Environment 42, 7452–7464 (2008)
5. Gardner, M.W., Dorling, S.R.: Artificial Neural Networks (The multilayer perceptron) - A review of applications in the atmospheric sciences. Atmospheric Environment 32, 2627–2636 (1998)
6. Kukkonen, J., Salmi, T., Saari, H., Konttinen, M., Kartastenpaa, R.: Review of urban air quality in Finland. Borial Environment Research 4, 55–65 (1999)
7. Kukkonen, J., Partanen, L.: Extensive evaluation of neural network models for tha prediction of NO_2 and PM_{10} concentrations, compared with a deterministic modelling system and measurements. Atmospheric Environment 37, 4539–4550 (2003)
8. Pérez, P., Trier, A.: Prediction of $PM_{2.5}$ concentrations several hours in advance using neural networks in Santiago, Chile. Atmospheric Environment 34, 1189–1196 (2000)
9. Kurt, A., Gulbagci, B.: An online air pollution forecasting system using neural networks. Environmental International 34, 592–598 (2008)

10. Bruneli, U., Piaza, V., Pignato, L., Sorbello, F., Vitabile, S.: Two-days ahead prediction of daily maximum concentrations of SO_2, O_3, PM_{10}, NO_2, CO in the urban area of Palermo, Italy. Atmospheric Environment 41, 2967–2995 (2007)
11. Ibarra-Berastegui, G., Elias, A., Barona, A., Saenz, J., Ezcurra, A., Diaz de Argandoña, J.: From diagnosis to prognosis for forcasting air pollution using neural network: Air pollution monitoring in Bilbao. Environmental Modelling and Software 23, 622–637 (2008)
12. Kumar Gautam, A., Chelani, A.B., Jain, V.K., Devotta, S.: A new scheme to predict chaotic time series of air pollutant concentrations using artificial neural network and nearest neighbor searching. Atmospheric Environment 42, 4409–4417 (2008)
13. Coman, A., Ionescu, A., Candau, Y.: Hourly ozone prediction for a 24-h horizon using neural networks. Environmental Modelling and Software 23, 1407–1421 (2008)
14. Celik, M.B., Kadi, I.: The relation between meteorological factors and pollutants concentration in Karabuk City. G.U. Journal of science 20(4), 87–95 (2007)
15. Bessagnet, B., Hodzic, A., Blachard, M., Lattuati, M., Le Bihan, O., Marfaing, H., Rouil, L.: Origin of particulate matter pollution episodes in wintertime over the Paris Basin. Atmospheric Environment 39, 6159–6174 (2005)
16. Vega-López, E.: Gaceta Oficial del Distrito Federal, Órgano de Gobierno del Distrito Federal (2006)

Author Index

Abad, José-Alfredo II-831
Abad-Grau, María M. II-1097
Abe, Takashi I-829
Abella, Fernando Martínez I-1184
Abrahão, Silvia II-134
Acosta, Leopoldo II-21
Aguiar, Vanessa I-917
Aguilar, Luis Joyanes II-284
Aguilar, Wendy I-789
Aguirre, Carlos I-9
Aizpurua, Gotzone II-1264
Alaiz-Rodríguez, Rocío I-122
Alamán, Xavier I-1098, II-772
Alberola, Juan M. I-522
Alcañiz, Mariano II-894
Aldana-Montes, José F. II-67
Alfonseca, Manuel I-472
Alique, José Ramón II-1282
Alonso, Alonso A. II-839, II-863
Alonso, Ricardo S. II-116, II-663
Alonso-Betanzos, Amparo I-456
Álvarez, B. Rosario Campomanes II-718
Álvarez-Estévez, D. I-997
Alvarez, Fernándo I-1280
Álvarez, I. I-941, I-949, I-965, I-973
Álvarez, José A. II-50, II-710
Álvarez, Juan Antonio II-843
Álvarez-Vellisco, Antonio I-174
Amanuma, Toshiyuki II-404
Amaral, Ronaldo S. I-1053
Amarger, Véronique I-1312, I-1328
Amigo, Víctor Castro II-483
Aminaiee, Abdol Hossein II-634
Andina, Diego I-174, I-1029, I-1336
Andrade, Angel G. II-702
Andrés, César I-343
Andrés-Gutiérrez, Juan J. I-1070
Anguita, M. I-861
Angulo, Cecilio I-773, I-1137
Annicchiarico, R. I-635
Annicchiarico, Roberta I-1083, I-1145, I-1161
Antón-López, Alfonso I-1037
Antúnez, E. I-781
Aparicio, José Paz II-718
Aramendi, Gorka II-1264
Arana, Ramón II-1264
Araujo, Álvaro II-695, II-752, II-800, II-914
Araujo, Carmen Paz Suárez I-981
Aravena, Francisco II-1248
Arazoza, Héctor de I-909
Arenas, M.G. I-383
Arenas-García, Jerónimo I-440
Argente, Estefanía II-154, I-642
Arias, Juan II-124
Arnaiz, Aitor II-1264
Arnay, Rafael II-21
Arquero, Agueda I-196
Arrais, Joel II-1101
Arrieta, Angélica González II-621
Asami, Katsushi II-404
Asensio, Ángel I-1113
Atemezing, Ghislain I-545
Atencia, Miguel I-886, I-901
Aznar, Fidel II-547

Baets, B. De I-302
Baeyens, Enrique II-240
Báez, Patricio García I-981
Bahrammirzaee, Arash I-610
Bajdik, Chris I-989
Bajo, Javier I-220, I-512, I-1106, I-1256, II-475, II-964
Ball, Graham R. I-877
Bandera, A. I-399, I-781
Baños, R. II-515
Barbero, Álvaro I-73
Barella, Antonio I-522
Barrenechea, E. I-302
Barrios, Jorge I-893, I-909
Barrón-Adame, J.M. I-1336
Barros, Lorena II-1147
Barros-Velázquez, Jorge II-1147
Barrué, A. Federici C. I-635
Barrué, Christian I-1121, I-1145
Baruque, Bruno I-165
Bastos, Hugo II-930

Bel-Enguix, Gemma I-464
Belle, V. Van I-65
Bello, Rafael II-458
Belmonte, Luis J. II-710
Benghazi, Kawtar II-211
Berghe, Greet Vanden I-1248
Berlanga, Antonio I-714
Bernardino, Anabela Moreira II-179
Bernardino, Eugénia Moreira II-179
Bernier, J.L. I-820
Berrocal, Javier I-1184
Bertini, Ilaria II-1195
Bertol, Douglas W. II-1256
Bertoni, Francesco II-1109
Bessant, Conrad II-1118
Biehl, M. I-933
Biganzoli, Elia I-989
Billhardt, Holger I-561
Blaciotti, Mona I-1083, I-1161
Blanco, Roberto II-1139
Blanes, David II-134
Blankertz, Benjamin I-651, I-674
Blasco-Marín, Rubén II-906, I-1113
Blasco, Xavier I-375
Blesa, Javier II-695
Blum, Christian II-1035
Bono-Nuez, Antonio II-906
Borda, Monica II-988
Börgermann, Chris II-248
Borges, Cruz E. I-1
Borrajo, Lourdes II-926, II-950
Bosch, Guillermo I-1121
Botia, Juan A. I-1090, II-75, II-819
Botti, Vicente II-146, II-154, I-594
Bourke, Alan I-1129
Bouveyron, Charles I-1021
Bravo, Jose II-870
Brehm, Nico II-366
Brezmes, Tomas II-796
Brignole, Nélida B. I-1272
Bringas, Pablo G. II-554
Bruque, S. I-286
Bueno, Jesús M. Miró II-1170
Buiza, Cristina II-756
Burrieza, J. Escuadra II-424
Bustillo, Andres II-1282
Bustince, H. I-302

Caballero, Armando II-1143
Caballero, Juan Antonio II-1084
Cabestany, Joan I-1137
Cabrera, Inma P. I-236
Cadenas, Jose M. I-1090
Calo-Mata, Pilar II-1147
Caltagirone, Carlo I-635, I-1145
Calvo, J.C. I-845, I-853, I-861
Calvo-Estévez, Rosa M. I-456
Calvo-Palomino, Roberto II-808
Calvo-Rolle, José Luís I-212
Camacho, Rui II-1001, II-1043
Cámara, Mario I-760
Campana, F. I-635
Campoy, Pascual I-1168
Canabal, Jose Manuel II-466
Cañas, Benito II-1147
Carballedo, Roberto II-626
Carballido, Jessica A. I-1272
Carcangiu, Sara I-744
Cárdenas, Clemente II-240
Carmona, Cristóbal I-752
Carneiro, Davide II-788
Carneiro, Sónia II-954
Carrascosa, C. I-642
Carreira, Rafael II-954
Carrera, Luis II-1248
Carrera, Pablo V. II-950
Carvajal-Rodríguez, Antonio II-1028
Carvajal-Rodríguez, Fernando II-1028
Carvalho, Omar S. I-1053
Carvalho, Paulo II-1178
Casado, Francisco II-163
Casañ, Gustavo A. II-264
Casanovas, Enrique II-458
Casas, David Martínez II-487
Casas, Roberto I-1113
Casas-Nebra, Roberto II-906
Castaño, M^a. Asunción II-264
Castelan, Eugênio B. II-1256
Castellanos, Juan II-586
Castellanos-Garzón, José A. I-367
Castillo, José R. Fernández del II-992
Castillo, Pedro A. II-163, I-383, I-925
Castillo S., José Luis II-992
Castro-Bleda, M.J. I-33
Català, Andreu I-1129
Catoira, Alba I-1184
Cecotti, Hubert I-706
Celda, Bernardo I-1005
Centeno, Roberto I-561
Ceruelo, Victor Pablos I-270

Cervantes, Alejandro I-728
Cetina, Carlos II-740
Chakeri, Alireza II-634
Chaves, R. I-941
Chebira, Abdennasser I-1312
Chen, Shu-An I-869
Cheong, Chi Hong II-107
Chicote, Juan C. II-890
Chikha, Soukeina Ben I-204
Chippendale, Paul II-687
Chohra, Amine I-610
Christiansen, Henning I-488
Cid-Sueiro, Jesús I-122
Cisneros, Francisco J. II-586
Civit, A. I-97, I-797
Claros, M. Gonzalo I-845, II-1084
Coarasa, Victorián I-1113
Coelho, Helder I-569
Colla, Valentina I-89, I-1320
Conte, G. I-585
Corbacho, Fernando I-448
Corchado, Emilio I-165, II-1282
Corchado, Juan M. II-116, I-165, I-220, I-512, I-561, I-594, I-642, II-663, II-964, II-972
Cordero, Pablo I-236, I-261
Correa, Maritza II-1282
Cortes, Ulises I-635, I-1083, I-1145, I-1161
Cortina-Januchs, M.G. I-1336
Costa, Rafael S. II-1063, II-1079
Costa, Ricardo II-788
Costa, Vítor Santos II-1001, II-1043
Cotelo, Carmen II-491
Cotos, José M. II-491, II-531
Cotrina, Josep II-796
Couto, Francisco II-930, II-934, II-942
Couto, P. I-302
Crespo, Rubén González II-284, II-292, II-299, II-323, II-330
Cruz, José Bulas II-566
Cubero-Atienza, Antonio J. II-1240
Cubero-Torres, Sergio II-562
Cubiles-de-la-Vega, M^a-Dolores I-1304
Cubo, Oscar II-979
Cunha, J. Boaventura II-1222

Dahl, Veronica I-480
Danciu, Daniela I-24
Danóczy, Márton I-651
Daouzli, Adel I-140
Daryl, Essam II-171
Delgado, Miguel II-764
Delgado, Soledad I-196
Delicato, Flávia C. II-851
De Paz, Juan F. I-220
Derderian, Karnig I-327
Deris, Safaai II-187, II-495
Destexhe, Alain I-140
Deusdado, Sérgio II-1178
Diaz, Antonio F. II-462
Díaz, David II-1084
Díaz, Fernando II-954, II-972
Díaz, José E. II-475
Dirkmorfeld, Matthias II-248
Dominguez, Enrique II-1274
Domínguez, Rubén II-1092
Dorado, Gabriel II-1084
Dorado, Julian I-391
Dorronsoro, José R. I-73
Dossou, Paul-Eric II-374
Duran, L. I-1153
Dutra, Luciano V. I-1053

Encina, Alberto de la I-318
Enciso, M. I-261
Endo, Toshiro II-397
Engelbrecht, Andries P. II-450
Escolano, F. I-399, I-407
España-Boquera, S. I-33
Espinosa, Agustin I-522
Espinosa, Ainhoa Montero de II-752
Estrada, Griselda Cobos II-621
Estudillo, Miguel A. II-722
Etchells, Terence A. I-989
Etxaniz, Aitziber II-756
Evangelista, Pedro II-1071
Expósito, J.E. Muñoz I-286

Fagundes, Moser I-561
Falgueras, Juan II-1084
Fanni, Alessandra I-744
Farshchian, Babak A. II-922
Faundez-Zanuy, Marcos I-1216
Fazli, Siamac I-651
Fdez-Riverola, Florentino II-954, II-972, II-1092
Felipe, Jonatán II-21
Félix, Paulo II-1020
Fernández-Blanco, Enrique I-391

Fernandez-Carmona, M. I-1153
Fernández-Espejo, Blanca I-1145
Fernández-Fernández, Héctor II-315
Fernández-López, Mariano II-91
Fernández-Lorenzo, Santiago II-195
Fernández-Montes, Alejandro II-843
Fernandez, A. I-642
Fernandes, Ana S. I-989
Fernández, Alberto I-57, I-294, I-561, I-594
Fernández, Fernando I-577
Fernández, Gloria García II-284, II-292, II-323, II-330
Fernández, J. I-302
Fernández, José-Jesús II-227
Ferreira, Eugénio C. II-954, II-1063, II-1071, II-1079
Ferreiro, Susana II-1264
Fertsch, Marek II-358
Finat, Javier II-727
Fiol-Roig, Gabriel II-902
Floreano, Dario I-768
Flórez-Revuelta, Francisco II-244
Florido, J.P. I-845, I-853, I-861
Fons, Joan II-740
Fonseca, José M. I-989
Fonseca, Nuno A. II-1001, II-1043
Fontecha, Jesús II-870
Fontenla-Romero, Oscar I-157
Fraga, David II-800
Fraile, Juan A. I-602, I-1106
Franco, Enrique Torres II-284, II-292
Franco, Samuel II-839
Freire, Ana I-917
Freitas, Corina C. I-1053
Fresnadillo, María J. II-1055
Frutos, Mariano I-1272
Fuentes, Carmen II-870
Fuentes, Lidia II-851
Fuentes-Fernández, Rubén I-514, I-537
Fujikawa, Kazutoshi II-655
Fujita, Masanobu II-590

G-Bustelo, B. Cristina Pelayo II-299
Gabaldón, Toni II-1125
Gadea, Rafael I-359
Galán, S. García I-286
Galiano, V. II-99
Gallardo, José M. II-1147
Gallardo, Ramón I-1280
Gallinas, Roberto Berjón I-1232, I-1240
Gálvez, Sergio II-1084
Gámez, Nadia II-851
Galuszka, Adam II-438
Galván, Inés M. I-728
Garbin, Riccardo II-1195
García-Bernal, M.A. I-188, I-1208
García-Bustelo, Cristina Pelayo II-292, II-323
García-Chamizo, Juan Manuel II-244
García-Díaz, Vicente II-284, II-299, II-315
García-Fernández, Gloria II-315
Garcia-Fornes, Ana I-522
García-Garaluz, Esther I-886, I-901
García-Gómez, Juan Miguel I-1005
García-Herranz, Manuel II-772, I-1098
García-Lagos, Francisco I-901
García-Macías, J. Antonio II-827
García-Magariño, Iván I-514, I-537, I-545, I-619
García-Morate, Diego I-1037
García-Pereira, María Jesús II-1143
García-Ramos, Sara II-91
García-Rodríguez, José II-244
García-Sánchez, P. I-925
Garcia-Sanchez,Pablo II-163
García-Sebastián, Maite I-957
García-Sola, Alberto II-75
Garcia-Vazquez, Juan P. II-702
García, Alfredo S. Alguero II-483, II-718
García, Ana Fermoso I-1232, I-1240
García, Antonio I-1280
García, Begoña II-671, II-812, II-859
García, Carlos Armando I-367
García, Carlos J. I-1280
García, Enrique II-1055
García, Inmaculada II-219
García, Javier I-577
García, Jesús II-13
García, José E. II-1055
García, Óscar II-663
García, Sandra I-728
Gayarre, L. I-813
Gestal, Marcos I-805
Geven, Arjan II-756
Ghani, Nor Azura Md II-604
Ghita, Camelia I-1083, I-1161
Gil, C. II-515

Gil-Martínez-Abarca, Juan Antonio I-1296
Giner, Pau II-740
Giralda, D. Boto I-149
Giret, A. I-642
Glez-Peña, Daniel II-926, II-950, II-954, II-972, II-1092
Godo, Lluís I-1053
Golińska, Paulina II-358
Golinska, Paulina II-366, II-382
Gómez-López, Gonzalo II-1092
Gómez-Pulido, Juan Antonio II-179
Gómez-Rodríguez, Alma II-29
Gómez-Rodríguez, F. I-97
Gomez-Ruiz, Jose Antonio I-41
Gómez-Ruiz, J.A. I-188, I-1208
Gómez-Sanz, Jorge J. I-514, I-619
Gómez-Verdejo, Vanessa I-440
Gómez, Daniel II-240
Gómez, J. II-515
Gómez, M. I-813
Gómez, Pedro II-988
Gómez, Yudel II-458
Goñi, Ángel II-586
González-Moreno, Juan C. II-29
González, Carina S. II-582
González, Evelio J. II-21, II-582
González, J. I-925
González, José Ángel Taboada II-487, II-727
González, Santiago II-979
Gonzalez, José II-1248
Gonzalo, Consuelo I-196
Gorricho, Juan-Luis II-796, II-831
Górriz, J.M. I-941, I-949, I-965, I-973
Goyeneche, Juan-Mariano de II-695, II-752, II-800
Graña, Manuel I-957
Grandinetti, Lucio II-223
Gräser, Axel I-706
Grassi, Marco I-1216
Gray, Philip II-748
Grego, Tiago II-934, II-942
Grolmusz, Vince II-1162
Grzenda, Maciej II-1213
Gu, Baohua I-480
Guerrero-Curieses, Alicia I-122
Guerrero, José Luis II-13
Guger, Christoph I-690
Guijarro-Berdiñas, Bertha I-157, II-195
Guillén, Alberto I-49, I-335, I-1224
Guimarães, Ricardo J.P.S. I-1053
Guixeres, Jaime II-894
Gutiérrez, Celia I-537, I-619
Gutierrez, Encarnacion Beato I-1240
Gutierrez, German I-720
Gutiérrez, M. Encarnación Beato I-1232
Gutierrez, Unai II-626
Guzmán, I.P. de I-261

Haber, Rodolfo E. II-338
Hamano, Yuta I-829
Hamilton, Alberto II-21
Haritou, Maria II-878
Harumoto, Kaname II-646
Haya, Pablo II-772
Haya, Pablo A. I-1098
Hema, Banati II-25
Heras-Quirós, Pedro de las II-808
Hernández-Capel, Dolores M. II-710
Hernández, Mario I-1288
Hernández, Pilar II-1084
Herrera, Francisco I-57, I-81, I-294
Herrera, L.J. I-49, I-853
Herrero, Jesús García I-714
Herrero, Pilar I-114
Hervás, Ramón II-870
Hidalgo-Herrero, Mercedes I-318
Hierons, Robert M. I-327
Higa, Masaki II-590
Higuera, J.F. Díez I-149
Hiramatsu, Ayako II-397, II-416
Hirofuchi, Takahiro II-651
Holdyk, Andrzej II-438
Hornos, Miguel J. II-211
Howe, Franklyn A. I-1005
Huerta-Cepas, Jaime II-1125
Huffel, S. Van I-65
Hutter, Marcus II-1109

Ibarz, Alejando I-1113
Iborra, Emilio II-819
Ibrahim, Zuwairie II-446, II-450
Iglesias, Eva L. II-926, II-950
Ikebe, Minoru II-655, II-659
Ikemura, Toshimichi I-829
Ilarri, Sergio I-627
Imamura, Makoto II-404
Inomata, Atsuo II-655
Insfran, Emilio II-134

Ioancio, Ioana I-1083, I-1161
Ipiña, Diego López de II-870
Isea, Raúl II-1130
Isern, David II-232
Ito, Akira II-397
Itoh, Satoshi II-651
Izquierdo-Cortázar, Daniel II-808
Izquierdo, Alberto II-675

Jara, Antonio J. II-882
Jarman, Ian H. I-989
Jemain, Abdul Aziz II-604
Jesus, María José del II-37, I-57, I-294, I-383, I-752
Jevtić, Aleksandar I-174
Jhun, Min A. I-837
Jiang, Jianmin II-735, II-780
Jimenez-Fernandez, A. I-797
Jiménez-López, M. Dolores I-464
Jiménez, A. I-97
Jiménez, C. II-515
Jiménez, G. I-97
Jiménez, José Paz II-718
Jiménez, María I. II-675, II-839
Jimenez, G. I-797
Jimeno, Antonio II-612
Johnson, Samuel I-17, I-228
Joya, Gonzalo I-886, I-901
Julià-Sapé, Margarida I-1005
Julián-Iranzo, Pascual I-245
Julián, P. I-253
Julián, Vicente II-154, I-553, I-561, I-594, I-642

Kanaya, Shigehiko I-829
Kanbara, Masayuki II-659
Kang, Dongshik II-590
Kappen, Hilbert J. I-17
Karanik, Marcelo I-41
Kawa, Arkadiusz II-346
Kawaguchi, Nobuo II-638
Kawanabe, Motoaki I-674
Kayal, Siavash II-634
Keilwagen, J. I-933
Keller, Laurent I-768
Kelly, Rafael I-530
Khalid, Marzuki II-446, II-450
Khalid, Noor Khafifah II-446, II-450
Khelifi, Fouad II-735
Kleffe, Jürgen I-1061
Kohigashi, Kohichi II-646
Komoda, Norhisa II-397
Koutsouris, Dimitris II-878
Krause, Jens-Uwe I-1061
Kurniawan, Tri Basuki II-446, II-450
Kwee, Ivo II-1109

Lackes, Richard II-248
Lago-Fernández, Luis F. I-448
Laguna, Javier II-870
Laguna, Miguel A. II-727
Lamas-Rodríguez, Julián II-124
Lancashire, Lee J. I-877
Lancho, Ángel Luis Sánchez Lázaro y Belén Pérez II-621
Lanz, Oswald II-687
Laredo, Juan L.J. II-163
Lau, Nuno I-805
Lau-Fernández, Rogelio I-1296
Laza, Rosalía II-926, II-950
Lemetre, Christophe I-877
Lendasse, Amaury I-359, I-1224
León, Coromoto II-41
Levorato, Paolo II-1195
Libal, Vit II-687
Lima, Luís II-566
Linares-Barranco, A. I-97, I-797
Liong, Choong-Yeun II-256, II-604
Lisboa, Paulo J.G. I-989
Llinás, Pablo II-772
Lluch-Ariet, Magí I-1005
Lopes, Pedro II-1101
Lopez-Arevalo, Ivan II-539, II-574
López-Bueno, Agustín II-562
López-Bueno, Iván I-577
López-Coello, Manuel I-1304
López-García, Hilario I-212
Lopez-Molina, C. I-302
López-Rodríguez, Domingo I-432
López-Rubio, Ezequiel I-106, I-180
Lopez, Juan C. II-130
Lopez, Miguel Angel I-666, I-683
López, M. I-941, I-949, I-965, I-973
López, M.F. II-280
López, Pedro L. I-752
López, Silvia II-870
López, Sira II-926
López, Vivian I-220
Lorenzo, Javier I-1288
Lorenzo-Fonseca, Iren I-1296

Loukanova, Roussanka I-496
Lourenço, Anália II-954
Lousado, José P. II-1047
Lowe, David I-1264
Lozano, Jesús I-1280
Lozano, M.A. I-407
Lucas, Caro II-634
Luengo, Julián I-81
Lujan-Martinez, C. I-797
Luque, Cristóbal I-728
Luque, Rafael M. I-432, II-1274

Macarro, Antonia I-602
Machado, Catia II-930
Machado, Daniel II-1063, II-1079
Machado, J.A. Tenreiro II-1231
Machón-González, Iván I-212
Maciá-Pérez, Francisco I-1296
Macukow, Bohdan II-1213
Madani, Kurosh I-610, I-1312, I-1328
Madeira, Sara C. II-1013
Maervoet, Joris I-1248
Maharshak, Erez I-480
Malagón, Pedro II-695, II-800, II-914
Malcata, Francisco X. II-1118
Malutan, Raul II-988
Maña, Manuel J. II-1205
Mana, Nadia II-687
Marcano-Cedeño, Alexis I-174
Marco, Álvaro I-1113
Marcos-Jorquera, Diego I-1296
Marfil, R. I-399, I-424
Margaliot, Michael I-131
Marques, Paulo I-569
Márquez, Juan L. II-475
Marrero, Aymée I-909
Marro, Joaquín I-228
Martín-Gago, J.A. II-280
Martín-del-Brío, Bonifacio II-906
Martín, Ángel II-1055
Martin, Mariana I-1083, I-1161
Martín, V. II-280
Martinez-Gil, Jorge II-67
Martínez-Marín, T. I-813
Martínez-Ortigosa, Pilar II-219
Martínez-Rego, David II-195
Martínez-Zarzuela, M. I-149
Martínez, J. II-280
Martínez, M. II-99
Martinez, Estibaliz I-196
Martínez, Oscar Sanjuán II-284
Martins-Bedé, Flávia T. I-1053
Martins, Nardênio A. II-1256
Marzal, Andrés II-1154
Marzouki, Kirmene I-204
Mata, Aitor I-165
Mata, Jacinto II-1205
Matam, B.R. I-1264
Mateo, Fernando I-359
Matsuyama, Kengo II-307
Mayo, Rafael II-1130
Mayordomo, Elvira II-1139
McBryan, Tony II-748
McKay, Bob II-171
Medina, Jesús I-278
Mejias, Jorge F. I-17, I-228
Mena, Eduardo I-627
Mendes, Luís II-1231
Mera, David II-491
Merayo, Mercedes G. I-327
Mercaş, Robert I-472
Merelo, J.J. I-383
Merelo, Juan J. II-37, II-163
Méndez, Amaia II-671, II-812, II-859
Méndez, José R. II-954, II-972
Méndez, Juan I-1288
Mesa, Héctor I-1045
Meziat, D. I-813
Miche, Yoan I-1224
Migallón, H. II-99
Miguélez, Mónica I-1192
Miguel-Quintales, Luis A. I-367
Miranda, Gara II-41
Miró-Julià, Margaret II-902
Mitchell, Philip II-374
Mitsui, Ikugo II-590
Mohamad, Mohd Saberi II-187, II-495
Molina, José Manuel II-13, II-706, I-714
Molinero, Carlos I-343
Monroy, Carmen I-530
Montaña, José L. I-1
Monteiro, Mariana B. II-1118
Montes, Esther II-1130
Montes, Rosana II-1097
Montisci, Augusto I-744
Montoro, Germán II-772
Mora-Gimeno, Francisco José I-1296
Mora, A. I-261
Mora, Antonio M. II-163
Mora, Higinio II-612

Mora, Marco II-1248
Moral, Sergi del I-773
Moreira, Patrícia R. II-1118
Moreno, Antonio II-232
Moreno, G. I-253
Moreno, Lorenzo II-582
Moreno, Mailyn II-138
Moreno-Clari, Paloma II-562
Morente, Laura I-1045
Moret-Bonillo, V. I-997
Morganti, G. I-585
Morihisa, Hiroshi II-412
Moura, Gabriela R. II-1047
Mourelle, Luiza de Macedo I-736
Moya, Eduardo J. II-240
Moya, Francisco II-130
Moya, José M. II-695, II-752, II-800, II-914
Müller, Klaus-Robert I-651, I-674
Munoz-Hernandez, Susana I-270
Muñoz, Jorge I-720
Muñoz, Jose II-1274
Muñoz-Perez, J. I-188, I-1208
Muñoz-Romero, Sergio I-440
Munteanu, Cristian R. I-917
Murakami, Koso II-523

Nagy, Benedek I-504
Naimi, Mohamed II-58
Nakada, Hidemoto II-651
Naoki, Mori II-171
Nasrudin, Mohammad Faidzul II-256
Navarro, Alfredo T. I-1005
Navarro, Martí I-553
Nedjah, Nadia I-736
Neves, José II-788
Nieto-Taladriz, Octavio II-695
Nieves, Javier II-554
Nishio, Shojiro II-646
Nistal, Ángel Martínez II-718
Noguera, Manuel II-211
Nose, Kazuo II-416
Novais, Paulo II-566, II-788
Novelle, Ignacio Rodríguez II-330
Nowé, Ann II-458
Núñez, Manuel I-327, I-343
Nuta, Costin I-1083, I-1161

Ogawa, Hirotaka II-651
Oiso, Hiroaki II-397
Ojeda-Aciego, Manuel I-236
Ojeda-Magaña, B. I-1029
Okayama, Yasumi II-397
Olivera, Ana C. I-1272
Oliveira, Arlindo L. II-1013
Oliveira, José Luís II-1047, II-1101
Oliveira, P.B. de Moura II-1222, II-1231
Olivieri, David II-1187
Olteanu, Madalina I-886
Omar, Khairuddin II-256
Omatu, Sigeru II-1, II-187, II-307, II-495, II-616
Ördög, Rafael II-1162
Ortea, Ignacio II-1147
Ortega, Alfonso I-472
Ortega, D. González I-149
Ortega, Juan Antonio II-843
Ortega, Julio II-462, I-760, I-853, I-861
Ortiz, Andres II-462
Ortiz-de-Lazcano-Lobato, Juan Miguel I-106, I-180, I-432
Ossowski, Sascha I-561, I-594
Otero, Abraham II-91, II-1020
Otero, M^{a}. Jesús L. II-531
Ou, Yu-Yen I-869
Oudheusden, Dirk Van I-1248

Pablos, A. Tejero de I-149
Pachón, Victoria II-1205
Padilla, N. II-515
Palacio, Daniel Zapico II-292, II-330
Palacios-González, Elías II-299
Palafox, Luis E. II-827
Paletta, Mauricio I-114
Palma, Jose T. II-819
Palomo, Esteban J. I-432, II-1274
Pannicelli, Alessandro II-1195
Pardo, Diego I-773
Parera, Jordi I-1137
Park, Taesung I-837
Parra, Xavier I-1129
Parras-Gutierrez, Elisabet II-37, I-383
Pascual, Pedro I-9
Pasek, Zbigniew J. II-350
Patricio, Mateus II-1125
Pau, Iván II-679
Pavón, Juan I-545, II-138
Pavón, Reyes II-926, II-950
Pawlewski, Paweł II-350, II-358, II-390
Paz-Vicente, R. I-797

Paz, Juan F. De II-663, II-116, II-964, I-1106, I-1256,
Paz, R. I-97
Pazos, Alejandro I-391
Pech-May, Fernando II-539
Pedrero, Alberto I-602
Peet, Andrew C. I-1005
Peláez, José Ignacio I-41
Pelayo, Francisco I-666, I-683
Pelckmans, K. I-65
Pelechano, Vicente II-740
Penabad, J. I-253
Penya, Yoseba K. II-554
Perakis, Konstantinos II-878
Perallos, Asier II-626
Peń, Jose Maria II-979
Peña, Andrés de la II-586
Peñalver, Ana II-679
Pérez-Caparrós, D. II-99
Pérez-Castrejón, Esteban I-1070
Pérez-Delgado, María Luisa II-424, II-507, II-596
Pérez-Godoy, M. Dolores I-57, I-752
Pérez-Lorenzo, J.M. I-781
Pérez-Sánchez, Beatriz I-157
Pérez, Andrés Sánchez I-893
Pérez, Belén I-1256
Pérez, David II-819
Pérez, Diana I-472
Pérez, Javier Andréu II-843
Pérez, Juan L. I-1184
Perdon, A.M. I-585
Pereira, Max II-1043
Perekrestenko, Alexander I-472
Peris, Guillermo II-1154
Pernas, F.J. Díaz I-149
Pesquita, Catia II-934
Pet, José Varela II-487
Petsatodis, Theodore II-756
Peula, Jose M. I-1145, I-1153
Pęzik, Piotr II-942
Pianesi, Fabio II-687
Piétrus, Alain I-909
Picado, Hugo I-805
Piera, Javier Roca II-50
Pieri, Edson R. De II-1256
Pineda, Luis A. I-789
Piñeiro, J.D. II-582
Pino-Mejías, Rafael I-1304
Pino, Miguel Angel Pérez del I-981
Pintado, Manuela E. II-1118
Pires, E.J. Solteiro II-1222, II-1231
Pires, Paulo F. II-851
Pisano, David G. II-1092
Pita, Víctor II-531
Pizzuti, Stefano II-1195
Pomares, Héctor I-49, I-335, I-683, I-820, I-845, I-853
Ponzoni, Ignacio II-1005, I-1272
Popescu, Florin I-651
Posada, David II-1125
Posadas, Juan L. II-83, II-432
Potamianos, Gerasimos II-687
Poza, José L. II-83, II-432
Prado, R.P. I-286
Prieto, Alberto II-462, I-683, I-925
Prieto, B. I-973
Prueckl, Robert I-690
Puertas, Jerónimo I-1192
Pujol, Francisco A. II-547, II-612
Pujol, Mar II-547
Punam, Bedi II-25
Puntonet, C.G. I-949, I-965, I-973

Quesada, C. II-99
Quesada, Humberto II-1143
Quintanilla-Domínguez, J. I-1029
Quintanilla, Andrés Alonso II-483

Rabanal, Pablo I-310, I-318, I-351
Raboso, Mariano II-675
Rabuñal, Juan Ramón I-1192
Ramabhadran, Bhuvana II-687
Ramírez, J. I-941, I-949, I-965, I-973
Rancoita, Paola M.V. II-1109
Răsvan, Vladimir I-24
Real, Pedro I-415
Rebholz-Schuhmann, Dietrich II-942
Rebollo, Miguel II-146, I-553, I-642
Recio, Ignacio II-914
Redel-Macías, M. Dolores II-1240
Redondo, Carmen M. II-950
Redondo, Juana L. II-219
Redondo, Raquel II-1282
Rees, Robert C. I-877
Reina-Tosina, Javier II-722
Reis, Luis P. I-805
Renaud, Sylvie I-140
Rey, Ángel Martín del II-503, I-1200

Reyneri, Leonardo M. I-89
Reynoso-Meza, Gilberto I-375
Rivas, M. I-97
Rivas, Victor M. II-37, I-383
Rivera, A.J. I-57
Rivera, Antonio J. I-752
Rivero, Daniel I-391
Rizo, Ramón II-547
Roa-Valverde, Antonio J. II-67
Roa, Laura M. II-722
Robles, Montserrat I-1005
Robles, Victor II-979
Roca-Piera, Javier II-227
Rocha, Isabel II-954, II-1063, II-1071, II-1079
Rocha, Luis M. II-954
Rocha, Miguel II-954, II-1063, II-1071
Rodríguez-Molinero, A. I-1137
Rodríguez-Patón, Alfonso II-1170
Rodríguez, Ángel II-232
Rodriguez, David II-675
Rodríguez, Gerardo II-1055
Rodríguez, Héctor de Arazoza I-893
Rodríguez, Ismael I-310, I-351
Rodríguez, Juan Enrique Arias II-487
Rodríguez, M. Antón I-149
Rodriguez, Marcela D. II-702
Rodríguez, Sara I-220, I-594, I-1106, I-1256, II-964
Rojas, I. I-49, I-335, I-820, I-845, I-853, I-1224
Roli, Andrea II-1035
Román-López, Raúl II-808
Romero, Elena II-695, II-800
Romero, Enrique I-1013
Romero, Luís II-531
Romero, Luis Alonso II-621
Ron-Angevin, Ricardo I-666, I-698
Ros, María II-764
Rosa, Ramón de la II-675, II-839, II-863
Rosa, Sonia de la II-863
Rosal, Emilio del I-472
Rosário, Maria J. II-1231
Rosete, Alejandro II-138
Roth, Ido I-131
Roy-Yarza, Armando II-906
Rubio-Manzano, Clemente I-245
Rubio-Montero, Antonio J. II-1130
Rubio, Fernando I-310, I-318
Rubio, Ginés I-49, I-335, I-853
Rudolph, Michelle I-140
Ruiz, Ibon II-671, II-812, II-859

Saavedra, Alberto II-663
Sabourin, Christophe I-1328
Sacristán, Juan Carlos II-466
Saïghi, Sylvain I-140
Salaberria, Itziar II-626
Salas-Gonzalez, D. I-941, I-949, I-965, I-973
Salas-Morera, Lorenzo II-1240
Sánchez-Aguilera, José M. II-475
Sánchez-Maroño, Noelia I-456
Sánchez-Pérez, Juan Manuel II-179
Sánchez-Pi, Nayat II-706
Sánchez, David II-232
Sánchez, Gerardo Rodríguez I-1200
Sánchez, José Luis II-612
Sánchez, Miguel II-475
Sánchez, Miguel A. II-475
Sánchez, Montserrat Mateos I-1232, I-1240
Sánchez, S. I-813
Sánchez, Teresita Noriega I-893
Sanchis, Araceli I-720
Sanchis, Javier I-375
Sandoval, Francisco I-424, I-901, I-1145, I-1153
Sandri, Sandra I-1053
Sanjuan-Martínez, Óscar II-299, II-315, II-323
Santiago, A.J. Sánchez I-286
Santofimia, Maria J. II-130
Santos-Cadenas, José Antonio II-808
Santos, Igor II-554
Santos, Manuel A.S. II-1047
Santos, Pedro II-1282
Sanz, Andres Castillo II-292
Sarasa, Antonio II-466
Sas, Paul II-1240
Satoh, Hironobu II-272
Savio, Alexandre I-957
Scaradozzi, D. I-585
Schleif, F.-M. I-933
Schmittdiel, Michael II-338
Sedano, Javier II-1282
Segovia, F. I-941, I-949, I-965, I-973
Segura, Carlos II-41
Seijas, J. I-1029
Seoane, Jose A. I-917

Serrano, Emilio I-1090
Severo, Aymee Marrero I-893
Sevilla, Juan P. II-163
Sgarbi, Mirko I-89, I-1320
Shikha, Mehta II-25
Shimada, Hideki II-659
Shiraishi, Yuhki II-203
Shoji, Nobuo II-590
Sierra, Basilio II-1264
Sigut, Marta II-21
Silva-Ramírez, Esther-Lydia I-1304
Silva, M. Laura da II-227
Silva, Marcus Vinícius Carvalho da I-736
Simó, José E. II-83, II-432
Simón-Hurtado, Arancha I-1037
Skarmeta, Antonio F.G. II-882
Soldatos, John II-756
Sopena, Josep María I-1013
Sorjamaa, Antti I-1224
Sosa-Sosa, Victor II-539, II-574
Soto, Axel J. II-1005
Sotos, León González II-992
Souffiau, Wouter I-1248
Sovilj, Dušan I-359
Spiru, Luiza I-1083, I-1161
Stachowiak, Agnieszka II-390
Stefan, Lucian I-1083, I-1161
Strass, Hannes I-270
Strickert, M. I-933
Such, Jose M. I-522
Sueishi, Junichiro II-412
Sunahara, Hideki II-655, II-659
Suykens, J.A.K. I-65
Svagård, Ingrid II-922

Takahashi, Kentaro II-646
Takeda, Fumiaki II-203, II-272
Takeuchi, Saori II-171
Tamargo, Marco Antonio García II-483, II-718
Taminau, J. II-458
Tapia, Dante I. I-1106, I-1256, II-116, II-663
Tello-Leal, Edgar II-574
Terol, Amelia Bilbao II-483
Thiare, Ousmane II-58
Tidor, Bruce II-1063
Tode, Hideki II-523
Toledo, Jonay II-21
Tolosa, José Barranquero II-299, II-315, II-323
Tomé, Ana M. I-805
Toro, Francisco de I-760
Toro, Raúl M. del II-338
Torrent, Marc I-1129
Torres, Joaquín J. I-17, I-228
Tortajada, Salvador I-1005
Toscanelli, Orlando I-1320
Trigo, Paulo I-569
Trinanes, Joaquín A. II-491
Trundle, Paul II-735
Trundle, Paul R. II-780
Tscheligi, Manfred II-756
Turchenko, Volodymyr II-223
Turcu, Ileana I-1083, I-1161

Uchida, Kanako II-523
Uranishi, Yuki II-659
Urdiales, Cristina I-1145, I-1153
Urquiza, J. I-49
Urquiza, J.M. I-845, I-853, I-861
Urra, Oscar I-627

Vadillo, Laura II-679
Val, Elena del I-553, II-146
Val, Lara del II-675, II-839, II-863
Valenzuela, O. I-820
Valero, Miguel A. II-679
Vallejo, Juan Carlos II-695, II-752, II-800, II-914
Vannucci, Marco I-89, I-1320
Vansteenwegen, Pieter I-1248
Varela, José II-124
Vargas-González, María Carmen I-106, I-180
Vaz, João C. II-1231
Vázquez-Martín, R. I-781
Vazquez, Gustavo E. II-1005
Vega-Corona, A. I-1029, I-1336
Vega-Rodríguez, Miguel Angel II-179
Velasco-Álvarez, Francisco I-698
Vellido, Alfredo I-1013
Veredas, Francisco J. I-1045
Viadero, Carlos Fernández I-981
Vicente, Javier I-1005, II-671, II-812
Vidales, Miguel Ángel Sánchez I-1232, I-1240
Vidaurre, Carmen I-674
Vieira, Cristina II-1001
Vieira, Jorge II-1001

Vila, Amparo II-764
Villa, Ana II-819
Villa, David II-130
Villacorta, Juan J. II-675
Villanúa, Jorge I-957
Villanueva, Daniel II-695, II-800
Villanueva, Felix J. II-130
Villanueva, Ismael II-827
Villar, A. I-635
Villar, José Ramón II-1282
Villarreal, Vladimir II-870
Villarroya, Sebastián II-531
Villmann, T. I-933
Viqueira, José R.R. II-124
Vivaracho-Pascual, Carlos I-1037
Volosyak, Ivan I-706
Vrančić, Damir II-1222

Wada, Kennosuke I-829
Watanabe, Keisuke II-404
Webb, Russell Y. I-1176
Wesseling, Pieter I-1005
Wong, Man Hon II-107
Wu, Dan II-107

Xuan, Nguyen II-171

Yamasaki, Takahiro II-416
Yokoya, Naokazu II-659
Yoshioka, Michifumi II-187, II-307, II-495, II-616
Yuste, A.J. I-286

Zainal, Anazida II-495
Zakaria, Mohamad Shanudin II-256
Zamora-Martínez, F. I-33
Zamora, Miguel A. II-882
Zaragozá, Irene II-894
Zardoya, Rafael II-1125
Zufiria, Pedro J. I-1168