

Emilio Corchado
Ajith Abraham
Witold Pedrycz (Eds.)

LNAI 5271

Hybrid Artificial Intelligence Systems

Third International Workshop, HAIS 2008
Burgos, Spain, September 2008
Proceedings

 Springer

Lecture Notes in Artificial Intelligence 5271

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Emilio Corchado Ajith Abraham
Witold Pedrycz (Eds.)

Hybrid Artificial Intelligence Systems

Third International Workshop, HAIS 2008
Burgos, Spain, September 24-26, 2008
Proceedings

Series Editors

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

Volume Editors

Emilio Corchado
Universidad de Burgos
Escuela Politécnica Superior
GICAP Research Group
E-mail: escorchado@ubu.es

Ajith Abraham
Norwegian University of Science and Technology
Center of Excellence for Quantifiable Quality of Service
7491 Trondheim, Norway
E-mail: ajith.abraham@ieee.org

Witold Pedrycz
University of Alberta
Department of Electrical and Computer Engineering
Edmonton, Alberta T6G 2V4, Canada
E-mail: pedrycz@ee.ualberta.ca

Library of Congress Control Number: 2008935394

CR Subject Classification (1998): I.2.6, I.2, H.3, H.4, H.2.8, F.2.2, I.4-6

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-87655-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-87655-7 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 is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2008
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12522033 06/3180 5 4 3 2 1 0

Preface

The Third International Workshop on Hybrid Artificial Intelligence Systems (HAIS 2008) presented the most recent developments in the dynamically expanding realm of symbolic and sub-symbolic techniques aimed at the construction of highly robust and reliable problem-solving techniques. Hybrid intelligent systems have become increasingly popular given their capabilities to handle a broad spectrum of real-world complex problems which come with inherent imprecision, uncertainty and vagueness, high-dimensionality, and non stationarity. These systems provide us with the opportunity to exploit existing domain knowledge as well as raw data to come up with promising solutions in an effective manner. Being truly multidisciplinary, the series of HAIS workshops offers a unique research forum to present and discuss the latest theoretical advances and real-world applications in this exciting research field.

This volume of *Lecture Notes on Artificial Intelligence* (LNAI) includes accepted papers presented at HAIS 2008 held in University of Burgos, Burgos, Spain, September 2008

The global purpose of HAIS conferences has been to form a broad and interdisciplinary forum for hybrid artificial intelligence systems and associated learning paradigms, which are playing increasingly important roles in a large number of application areas.

Since its first edition in Brazil in 2006, HAIS has become an important forum for researchers working on fundamental and theoretical aspects of hybrid artificial intelligence systems based on the use of agents and multiagent systems, bioinformatics and bio-inspired models, fuzzy systems, artificial vision, artificial neural networks, optimization models and alike.

This conference featured a number of special sessions: Hybrid Systems Based on Negotiation and Social Network Modelling, Real-World Applications of HAIS Under Uncertainty, Hybrid Intelligent Systems for Multi-robot and Multi-agent Systems, Genetic Fuzzy Systems: Novel Approaches and Applications of Hybrid Artificial Intelligence in Bioinformatics.

HAIS 2008 received over 280 technical submissions. After a thorough peer-review process, the International Program Committee selected 93 papers which are published in this conference proceedings. The large number of submissions is certainly not only a testimony of the vitality and attractiveness of the field but an indicator of the interest in the HAIS conferences themselves.

As a follow-up of the conference, we anticipate further publication of selected papers in special issues scheduled for the journal of *Information Sciences*, Elsevier Sciences, The Netherlands and the *International Journal On Computational Intelligence Research* (IJCIR). We would like to express our thanks to the Program Committee whose members did an outstanding job under a very tight schedule. Our thanks go to the keynote speakers: Bogdan Gabrys from Bournemouth University (UK), Francisco Herrera from the University of Granada (Spain), Xindong Wu from the University of Vermont (USA), and Hujun Yin from the University of Manchester (UK).

We wish to thank the staff of Springer for their help and collaboration during this demanding publication project. We would like to fully acknowledge the support we received from Junta de Castilla y León, Genoma España, University of Burgos, Fundación General de la Universidad de Burgos, and Ayuntamiento de Burgos y Diputación de Burgos.

September 2008

Emilio Corchado
Ajith Abraham
Witold Pedrycz

Organization

Honorary Chair

Alfonso Murillo Rector of the University of Burgos (Spain)

General Chair

Emilio Corchado University of Burgos (Spain)

International Advisory Committee

Ajith Abraham Norwegian University of Science and Technology (Norway)

Carolina Blasco Director of Telecommunication, Regional Government of Castilla y León (Spain)

Juan M. Corchado University of Salamanca (Spain)

José R. Dorronsoro Autonomous University of Madrid (Spain)

Samuel Kaski Helsinki University of Technology (Finland)

Isidro Laso D.G. Information Society and Media (European Commission)

Xin Yao University of Birmingham (UK)

Hujun Yin University of Manchester (UK)

Publicity Chairs

Dacheng Tao Hong Kong Polytechnic University (Hong Kong)

Emilio Corchado University of Burgos (Spain)

Program Committee

Abraham, Ajith Norwegian University of Science and Technology (Norway) (PC Co-chair)

Pedrycz, Witold University of Alberta (Canada) (PC Co-chair)

Alcalá, Rafael University of Granada (Spain)

Alonso, Luis University of Salamanca (Spain)

Anguita, Davide University of Genova (Italy)

Apolloni, Bruno Università degli Studi di Milano (Italy)

Aragón, Alberto University of Burgos (Spain)

Baets, Bernard de Ghent University (Belgium)

Bajo, Javier	University Pontificia of Salamanca (Spain)
Baruque, Bruno	University of Burgos (Spain)
Botía, Juan	University of Murcia (Spain)
Botti, Vicente	Polytechnic University of Valencia (Spain)
Bustillo, Andrés	University of Burgos (Spain)
Carvalho, André CPLF de	University of São Paulo (Brazil)
Castillo, Oscar	Tijuana Institute of Technology (Mexico)
Chbeir, Richard	Bourgogne University (France)
Cichocki, Andrzej	Brain Science Institute (Japan)
Corchado Emilio	University of Burgos (Spain)
Corchado, Juan M.	University of Salamanca (Spain)
Corchuelo, Rafael	University of Sevilla (Spain)
Curiel, Leticia	University of Burgos (Spain)
Damiani, Ernesto	University of Milan (Italy)
Dahal, Keshav	University of Bradford (UK)
Del Olmo, Ricardo	University of Burgos (Spain)
Dorrnsoro, José	Autonomous University of Madrid (Spain)
Dreyfus, Gérard	École Supérieure de Physique et de Chimie Industrielles de Paris (France)
Dumitrescu, Dan	University Babes-Bolyai (Romania)
Flores, Juan J.	Universidad Michoacana (Mexico)
Fukushima, Kunihiko	Kansai University (Japan)
Gabrys, Bogdan	Bournemouth University (UK)
Gams, Matjaz	Jozef Stefan Institute Ljubljana (Slovenia)
Girolami, Mark	University of Glasgow (UK)
Gopych, Petro	V.N. Karazin Kharkiv National University (Ukraine)
Graña, Manuel	University of Pais Vasco (Spain)
Grzymala-Busse, Jerzy	University of Kansas (USA)
Håkansson, Anne	Uppsala University (Sweden)
Halgamuge, Saman	The University of Melbourne (Australia)
Hassanién, Aboul Ella	Cairo University (Egypt)
Hatzilygeroudis, Ioannis	University of Patras (Greece)
Herrera, Francisco	University of Granada (Spain)
Herrero, Alvaro	University of Burgos (Spain)
Honavar, Vasant	Iowa State University (USA)
Jain, Lakhmi	University of South Australia (Australia)
Julián, Vicent	Universidad Politécnica de Valencia (Spain)
Karhunen, Juha	Helsinki University of Technology (Finland)
Karny, Miroslav	Academy of Sciences of Czech Republic (Czech Republic)
Keim, Daniel A.	Universität Konstanz (Germany)
Klawonn, Frank	University of Applied Sciences Braunschweig/Wolfenbuettel (Germany)
Köppen, Mario	Kyushu Institute of Technology (Japan)
König, Andreas	University of Kaiserslautern (Germany)
Kruse, Rudolf	Otto-von-Guericke-Universität Magdeburg (Germany)

Lee, Soo-Young	Brain Science Research Center (Korea)
Lhotská, Lenka	Czech Technical University (Czech Republic)
Liu, Honghai	University of Portsmouth (UK)
Luo, Wenjian	University of Science and Technology of China (China)
Markowska-Kaczmar, Urszul	Wroclaw University of Technology (Poland)
Martínez, José F.	Instituto Nacional de Astrofísica, Óptica y Electrónica (Mexico)
Mauri, Giancarlo	University of Milano-Bicocca (Italy)
Mira, José	Universidad Nacional de Educación a Distancia (Spain)
Nerode, Anil	Cornell University (USA)
Nicoletti, Maria do Carmo	Universidade Federal de São Carlos (Brazil)
Nojima, Yusuke	Osaka Prefecture University (Japan)
Pacheco, Joaquín	University of Burgos (Spain)
Palade, Vasile	Oxford University (UK)
Pavón, Juan	University Complutense of Madrid (Spain)
Pereira, Carlos	Universidade de Coimbra (Portugal)
Phillips-Wren, Gloria	Loyola College (USA)
Posada, Jorge	VICOMTech (Spain)
Reguera, Perfecto	University of León (Spain)
Ribeiro, Bernardete	University of Coimbra (Portugal)
Rizo, Ramón	University of Alicante (Spain)
Rossi, Fabrice	Institut National de Recherche en Informatique et en Automatique (France)
Samuelson Hong, Wei-Chiang	Oriental Institute of Technology (Taiwan)
Sedano, Javier	University of Burgos (Spain)
Tan, Ying	Peking University (China)
Tang, Ke	University of Science and Technology of China (China)
Uchino, Eiji	Yamaguchi University (Japan)
Villar, José R.	University of Oviedo (Spain)
Wang, Lipo	Nanyang Technological University (Singapore)
Wang, Tzai-Der	Cheng Shiu University (Taiwan)
Wermter, Stefan	University of Sunderland (UK)
Xu, Lei	Chinese University of Hong Kong (Hong Kong)
Yager, Ronald R.	Iona College (USA)
Yang, Ron	University of Exeter (UK)
Yao, Xin	University of Birmingham (UK)
Yin, Hujun	University of Manchester (UK)
Zunino, Rodolfo	University of Genoa (Italy)

Organizing Committee

Corchado, Emilio	University of Burgos (Chair)
Baruque, Bruno	University of Burgos (Co-chair)
Herrero, Álvaro	University of Burgos (Co-chair)
Arroyo, Angel	University of Burgos
Burgos, Pedro	University of Burgos
Bustillo, Andrés	University of Burgos
Canales, Jacinto	CPIICyL
Corchado, Juan Manuel	University of Salamanca
Curiel, Leticia	University of Burgos
Lara, Ana M	University of Burgos
López, Carlos	University of Burgos
Manzanedo, Miguel Ángel	University of Burgos
Marticorena, Raúl	University of Burgos
Martín, David	University of Burgos
Martín, Juan Vicente	University of Burgos
Pérez, Juan Carlos	University of Burgos
Sáiz, Jose Manuel	University of Burgos
Sáiz, Lourdes	University of Burgos
Sedano, Javier	University of Burgos
Vaquerizo, Belén	University of Burgos

Table of Contents

Invited Talks

Data Mining: Algorithms and Problems (Abstract)	1
<i>Xindong Wu</i>	
Do Smart Adaptive Systems Exist? Hybrid Intelligent Systems Perspective (Abstract)	2
<i>Bogdan Gabrys</i>	
Design of Experiments in Computational Intelligence: On the Use of Statistical Inference	4
<i>Salvador García and Francisco Herrera</i>	
Nonlinear Principal Manifolds – Adaptive Hybrid Learning Approaches	15
<i>Hujun Yin</i>	

Agents and Multi-agent Systems

Multi-agent ERA Model Based on Belief Interaction Solves Wireless Sensor Networks Routing Problem	30
<i>Yanbin Liu, Chunguang Zhou, Kangping Wang, Dan Li, and Dongwei Guo</i>	
Multi-agent System for Management and Monitoring of Routes Surveillance	38
<i>Sara Rodríguez and Javier Bajo</i>	
Classification Agent-Based Techniques for Detecting Intrusions in Databases	46
<i>Cristian Pinzón, Yanira De Paz, and Rosa Cano</i>	
Hybrid Multi-Agent Architecture (HoCa) Applied to the Control and Supervision of Patients in Their Homes	54
<i>Juan A. Fraile, Dante I. Tapia, and Miguel A. Sánchez</i>	
JADE/LEAP Agents in an Aml Domain	62
<i>Nayat Sánchez-Pi, Javier Carbó, and José Manuel Molina</i>	
Design Patterns for Combining Social and Individual Intelligences on Modular-Based Agents	70
<i>Bianca Innocenti, Beatriz López, and Joaquim Salvi</i>	

Experiments in Multi Agent Learning 78
Maria Cruz Gaya and J. Ignacio Giraldez

Agent-Based Simulation of Business Processes in a Virtual World 86
Branislav Bošanský and Cyril Brom

Temporal-Bounded CBR for the Management of Commitments in
 RT-Agents 95
Marti Navarro, Stella Heras, Vicente Botti, and Vicente Julián

Evolutionary Computation

A Constrained Dynamic Evolutionary Algorithm with Adaptive
 Penalty Coefficient 103
*Bo Xiao, Danpin Yu, Lei Zhang, Xin Tian, Song Gao, and
 Sanyou Zeng*

Enhanced Cooperative Co-evolution Genetic Algorithm for Rule-Based
 Pattern Classification 113
Fangming Zhu and Sheng-Wei Guan

Learning User Profile with Genetic Algorithm in AmI Applications 124
Verónica Venturini, Javier Carbó, and José M. Molina

Unsupervised Genetic Algorithm Deployed for Intrusion Detection 132
Zorana Banković, Slobodan Bojanić, Octavio Nieto, and Atta Badii

Automatic Neural Net Design by Means of a Symbiotic Co-evolutionary
 Algorithm 140
Elisabet Parras-Gutierrez, Víctor M. Rivas, and Maria Jose del Jesus

Hybrid Multi-population Collaborative Asynchronous Search 148
Anca Gog, Camelia Chira, and D. Dumitrescu

An Evolutionary Approach for Tuning Artificial Neural Network
 Parameters 156
Leandro M. Almeida and Teresa B. Ludermir

A Hybrid Evolutionary Multiobjective Approach for the Component
 Selection Problem 164
Andreea Vescan and Crina Grosan

A New Quantum Evolutionary Local Search Algorithm for MAX 3-SAT
 Problem 172
Abdesslem Layeb and Djamel-Eddine Saidouni

Neuro-evolutionary Decision Support System for Financial Time Series
 Analysis 180
Piotr Lipinski

Connectionist Models

Optimization of Knowledge in Companies Simulating M6PROK [©] Model Using as Hybrid Methodology a Neuronal Network and a Memetic Algorithm	188
<i>Ana Maria Lara, Lourdes Sáiz, Joaquín Pacheco, and Rafael Brotóns</i>	
STARMIND: Automated Classification of Astronomical Data Based on an Hybrid Strategy	196
<i>Alejandra Rodríguez, Iciar Carricajo, Minia Manteiga, Carlos Dafonte, and Bernardino Arcay</i>	
Spatio-temporal Road Condition Forecasting with Markov Chains and Artificial Neural Networks	204
<i>Konsta Sirvio and Jaakko Hollmén</i>	
Parameter Extraction from RVS Stellar Spectra by Means of Artificial Neural Networks and Spectral Density Analysis	212
<i>Diego Ordóñez, Carlos Dafonte, Minia Manteiga, and Bernardino Arcay</i>	
Supervised Classification Fuzzy Growing Hierarchical SOM	220
<i>Rafael del-Hoyo, Nicolás Medrano, Bonifacio Martín-del-Brio, and Francisco José Lacueva-Pérez</i>	
Self Optimizing Neural Networks SONN-3 for Classification Tasks	229
<i>Adrian Horzyk</i>	
Efficient MRI Reconstruction Using a Hybrid Framework for Integrating Stepwise Bayesian Restoration and Neural Network Models in a Memory Based Priors System	237
<i>D.A. Karras</i>	
Traffic Data Preparation for a Hybrid Network IDS	247
<i>Álvaro Herrero and Emilio Corchado</i>	

Optimization Systems

Comparing Hybrid Versions of SS and DE to Solve a Realistic FAP Problem	257
<i>José M. Chaves-González, Marisa da Silva Maximiano, Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido, and Juan M. Sánchez-Pérez</i>	
PSO for Selecting Cutting Tools Geometry	265
<i>Orlando Duran, Nibaldo Rodriguez, and Luiz Airton Consalter</i>	

A Hybrid Ant-Based System for Gate Assignment Problem	273
<i>Camelia-M. Pinteá, Petrica C. Pop, Camelia Chira, and D. Dumitrescu</i>	
Extracting Multi-knowledge from fMRI Data through Swarm-Based Rough Set Reduction	281
<i>Hongbo Liu, Ajith Abraham, and Hong Ye</i>	
Estimation Using Differential Evolution for Optimal Crop Plan	289
<i>Millie Pant, Radha Thangaraj, Deepti Rani, Ajith Abraham, and Dinesh Kumar Srivastava</i>	
Hybrid Metaheuristics for Global Optimization: A Comparative Study	298
<i>Antoniya Georgieva and Ivan Jordanov</i>	
Generating Routes with Bio-inspired Algorithms under Uncertainty	306
<i>Maria Belén Vaquerizo García</i>	

Fuzzy Logic Systems

Computer-Assisted Diagnosis of Primary Headaches	314
<i>Svetlana Simić, Dragan Simić, Petar Slankamenac, and Milana Simić-Ivkov</i>	
Ambient Temperature Modelling through Traditional and Soft Computing Methods	322
<i>Francesco Ceravolo, Matteo De Felice, and Stefano Pizzuti</i>	
Providing Dynamic Instructional Adaptation in Programming Learning	329
<i>Francisco Jurado, Olga C. Santos, Miguel A. Redondo, Jesús G. Boticario, and Manuel Ortega</i>	
Modelling Radial Basis Functions with Rational Logic Rules	337
<i>Davide Sottara and Paola Mello</i>	

Classification and Classifiers

On Combining Classifiers by Relaxation for Natural Textures in Images	345
<i>María Guijarro, Gonzalo Pajares, and P. Javier Herrera</i>	
An Ensemble Approach for the Diagnosis of Cognitive Decline with Missing Data	353
<i>Patricio García Báez, Carlos Fernández Viadero, José Regidor García, and Carmen Paz Suárez Araujo</i>	

Fusers Based on Classifier Response and Discriminant Function – Comparative Study	361
<i>Michal Wozniak and Konrad Jackowski</i>	
Simple Clipping Algorithms for Reduced Convex Hull SVM Training . . .	369
<i>Jorge López, Álvaro Barbero, and José R. Dorronsoro</i>	
A WeVoS-CBR Approach to Oil Spill Problem	378
<i>Emilio Corchado, Bruno Baruaque, Aitor Mata, and Juan M. Corchado</i>	

Cluster Analysis

Clustering Likelihood Curves: Finding Deviations from Single Clusters	385
<i>Claudia Hundertmark and Frank Klawonn</i>	
Unfolding the Manifold in Generative Topographic Mapping	392
<i>Raúl Cruz-Barbosa and Alfredo Vellido</i>	
Evaluation of Subspace Clustering Quality	400
<i>Urszula Markowska-Kaczmarska and Arletta Hurej</i>	
Clustering by Chaotic Neural Networks with Mean Field Calculated Via Delaunay Triangulation	408
<i>Elena N. Benderskaya and Sofya V. Zhukova</i>	

Video and Image Analysis

Image Fusion Algorithm Using RBF Neural Networks	417
<i>Hong Zhang, Xiao-nan Sun, Lei Zhao, and Lei Liu</i>	
Behaviour of Texture Features in a CBIR System	425
<i>César Reyes, María Luisa Durán, Teresa Alonso, Pablo G. Rodríguez, and Andrés Caro</i>	
Object Tracking Using Grayscale Appearance Models and Swarm Based Particle Filter	433
<i>Bogdan Kwolek</i>	
Extraction of Geometrical Features in 3D Environments for Service Robotic Applications	441
<i>Paloma de la Puente, Diego Rodríguez-Losada, Raul López, and Fernando Matía</i>	
Hybrid GNG Architecture Learns Features in Images	451
<i>José García-Rodríguez, Francisco Flórez-Revuelta, and Juan Manuel García-Chamizo</i>	

Learning Systems, Algorithms and Applications

Information-Theoretic Measures for Meta-learning	458
<i>Saddys Segrera, Joel Pinho, and María N. Moreno</i>	
An EM-Based Piecewise Linear Regression Algorithm	466
<i>Sebastian Nusser, Clemens Otte, and Werner Hauptmann</i>	
On the Use of Linear Cellular Automata for the Synthesis of Cryptographic Sequences	475
<i>A. Fúster-Sabater, P. Caballero-Gil, and O. Delgado</i>	
Ontology-Based Deep Web Data Sources Selection	483
<i>Wei Fang, Pengyu Hu, Pengpeng Zhao, and Zhiming Cui</i>	
A Type-2 Fuzzy Set Recognition Algorithm for Artificial Immune Systems	491
<i>Andrea Visconti and Hooman Tahayori</i>	
Symbolic Hybrid Programming Tool for Software Understanding	499
<i>Erkki Laitila</i>	

Hybrid Systems Based on Negotiation and Social Network Modelling

Characterizing Massively Multiplayer Online Games as Multi-Agent Systems	507
<i>G. Aranda, C. Carrascosa, and V. Botti</i>	
A Dialogue Game Protocol for Recommendation in Social Networks	515
<i>Stella Heras, Miguel Rebollo, and Vicente Julián</i>	
Friends Forever: Social Relationships with a Fuzzy Agent-Based Model	523
<i>Samer Hassan, Mauricio Salgado, and Juan Pavon</i>	
R ² -IBN: Argumentation Based Negotiation Framework for the Extended Enterprise	533
<i>Lobna Hsairi, Khaled Ghédira, Adel M. Alimi, and Abdellatif BenAbdelhafid</i>	
Extending Pattern Specification for Design the Collaborative Learning at Analysis Level	543
<i>Jaime Muñoz Arteaga, Ma. De Lourdes Margain Fuentes, Fco. Álvarez Rodríguez, and Carlos Alberto Ochoa Ortíz Zezzatti</i>	
Towards the Simulation of Social Interactions through Embodied Conversational Agents	551
<i>María Lucila Morales-Rodríguez, Bernard Pavard, Juan J. González B., and José A. Martínez F.</i>	

Ontology-Based Approach for Semi-automatic Generation of Subcategorization Frames for Spanish Verbs	558
<i>Rodolfo A. Pazos R., José A. Martínez F., Javier González B., María Lucila Morales-Rodríguez, Gladis M. Galiana B., and Alberto Castro H.</i>	
Diffusion of Domestic Water Conservation Technologies in an ABM-GIS Integrated Model	567
<i>José M. Galán, Ricardo del Olmo, and Adolfo López-Paredes</i>	
Real World Applications of HAIS Under Uncertainty	
Hybrid IT2 NSFLS-1 Used to Predict the Uncertain MXNUSD Exchange Rate	575
<i>Gerardo M. Mendez and Angeles Hernandez</i>	
Minimizing Energy Consumption in Heating Systems under Uncertainty	583
<i>José Ramón Villar, Enrique de la Cal, and Javier Sedano</i>	
Learning to Trade with Incremental Support Vector Regression Experts	591
<i>Giovanni Montana and Francesco Parrella</i>	
Craniofacial Superimposition Based on Genetic Algorithms and Fuzzy Location of Cephalometric Landmarks	599
<i>Oscar Ibáñez, Oscar Cordón, Sergio Damas, and Jose Santamaría</i>	
A Minimum Risk Wrapper Algorithm for Genetically Selecting Imprecisely Observed Features, Applied to the Early Diagnosis of Dyslexia	608
<i>Luciano Sánchez, Ana Palacios, and Inés Couso</i>	
Hybrid Intelligent Systems for Multi-robot and Multi-agent Systems	
An Approach to Flocking of Robots Using Minimal Local Sensing and Common Orientation	616
<i>Iñaki Navarro, Álvaro Gutiérrez, Fernando Matía, and Félix Monasterio-Huelín</i>	
Applying Reinforcement Learning to Multi-robot Team Coordination . . .	625
<i>Yolanda Sanz, Javier de Lope, and José Antonio Martín H.</i>	
A Complex Systems Based Tool for Collective Robot Behavior Emergence and Analysis	633
<i>Abraham Prieto, Francisco Bellas, Pilar Caamaño, and Richard J. Duro</i>	

On the Need of Hybrid Intelligent Systems in Modular and Multi Robotics	641
<i>Richard J. Duro, Manuel Graña, and Javier de Lope</i>	
Modelling of Modular Robot Configurations Using Graph Theory	649
<i>José Baca, Ariadna Yerpès, Manuel Ferre, Juan A. Escalera, and Rafael Aracil</i>	
A Hybrid Intelligent System for Robot Ego Motion Estimation with a 3D Camera	657
<i>Ivan Villaverde and Manuel Graña</i>	
Evolutionary Parametric Approach for Specular Correction in the Dichromatic Reflection Model	665
<i>Ramón Moreno, Alicia d'Anjou, and Manuel Graña</i>	
On Distributed Cooperative Control for the Manipulation of a Hose by a Multirobot System	673
<i>José Manuel López-Guede, Manuel Graña, and Ekaitz Zulueta</i>	
Multi-robot Route Following Using Omnidirectional Vision and Appearance-Based Representation of the Environment	680
<i>Luis Payá, Oscar Reinoso, Arturo Gil, and Javier Sogorb</i>	

Applications of Hybrid Artificial Intelligence in Bioinformatics

Using CBR Systems for Leukemia Classification	688
<i>Juan M. Corchado and Juan F. De Paz</i>	
Crosstalk and Signalling Pathway Complexity – A Case Study on Synthetic Models	696
<i>Zheng Rong Yang</i>	
Explore Residue Significance in Peptide Classification	706
<i>Zheng Rong Yang</i>	
Analysis of Non-stationary Neurobiological Signals Using Empirical Mode Decomposition	714
<i>Zareen Mehboob and Hujun Yin</i>	

Genetic Fuzzy Systems: Novel Approaches

Approximate Versus Linguistic Representation in Fuzzy-UCS	722
<i>Albert Orriols-Puig, Jorge Casillas, and Ester Bernadó-Mansilla</i>	
Fuzzy Classification with Multi-objective Evolutionary Algorithms	730
<i>Fernando Jiménez, Gracia Sánchez, José F. Sánchez, and José M. Alcaraz</i>	

Cooperation between the Inference System and the Rule Base by Using Multiobjective Genetic Algorithms	739
<i>Antonio Márquez, Francisco Alfredo Márquez, and Antonio Peregrín</i>	
Knowledge Base Learning of Linguistic Fuzzy Rule-Based Systems in a Multi-objective Evolutionary Framework	747
<i>P. Ducange, R. Alcalá, F. Herrera, B. Lazzerini, and F. Marcelloni</i>	
Effects of Diversity Measures on the Design of Ensemble Classifiers by Multiobjective Genetic Fuzzy Rule Selection with a Multi-classifier Coding Scheme	755
<i>Yusuke Nojima and Hisao Ishibuchi</i>	
Author Index	765

Data Mining: Algorithms and Problems

Xindong Wu

Department of Computer Science, University of Vermont
33 Colchester Avenue,
Burlington, Vermont 05405, USA
xwu@cs.uvm.edu

Abstract. Data mining, or knowledge discovery in databases (KDD), is an interdisciplinary field that integrates techniques from several research areas including machine learning, statistics, database systems, and pattern recognition, for the analysis of large volumes of possibly complex, highly-distributed and poorly-organized data. The prosperity of the data mining field may attribute to two essential reasons. Firstly, a huge amount of data is collected and stored everyday. On the one hand, along with the continuing development of advanced technologies in many domains, data is generated at enormous speeds. For examples, purchases data at department/grocery stores, bank/credit card transaction data, e-commerce data, Internet traffic data that describes the browsing history of Web users, remote sensor data from agricultural satellites, and gene expression data from microarray technology. On the other hand, the progress made in hardware technology allows today's computer systems to store very large amounts of data. Secondly, with these large volumes of data at hand, the data owners have an imminent intent to turn them into useful knowledge. From a commercial viewpoint, the ultimate goal of the data owners is to gain more and pay less for their business activities. Under the competition pressure, they want to enhance their services, develop cost-effective strategies, and target the right group of potential customers. From a scientific viewpoint, when traditional techniques are infeasible in dealing with the raw data, data mining may help scientists in many ways, such as classifying and segmenting data. By applying the knowledge extracted from data mining, the business analyst may rate customers by their propensity to respond to an offer, the doctor may estimate the probability of an illness re-occurrence, the website publisher may display customized Web pages to individual Web users according to their browsing habit, and the geneticist may discover novel gene-gene interaction patterns. In this talk, we aim to provide a general picture for important data mining steps, topics, algorithms and challenges.

Do Smart Adaptive Systems Exist? Hybrid Intelligent Systems Perspective

Bogdan Gabrys

Smart Technology Research Centre
Computational Intelligence Research Group
Bournemouth University
Talbot Campus, Fern Barrow, Poole, Dorset, BH12 5BB, UK
bgabrys@bournemouth.ac.uk

Abstract. Rapid development in computer and sensor technology not only used for highly specialised applications but widespread and pervasive across a wide range of business and industry has facilitated easy capture and storage of immense amounts of data. Examples of such data collection include medical history data in health care, financial data in banking, point of sale data in retail, plant monitoring data based on instant availability of various sensor readings in various industries, or airborne hyperspectral imaging data in natural resources identification to mention only a few. However, with an increasing computer power available at affordable prices and the availability of vast amount of data there is an increasing need for robust methods and systems, which can take advantage of all available information. In essence there is a need for intelligent and smart adaptive methods but do they really exist? Are there any existing intelligent techniques which are more suitable for certain type of problems than others? How do we select those methods and can we be sure that the method of choice is the best for solving our problem? Do we need a combination of methods and if so then how to best combine them for different purposes? Are there any generic frameworks and requirements which would be highly desirable for solving data intensive and unstationary problems? All these questions and many others have been the focus of research vigorously pursued in many disciplines and some of them will be discussed in the talk and have been addressed in greater detail in our recently compiled book with the same title: "Do Smart Adaptive Systems Exist?". One of the more promising approaches to constructing smart adaptive systems is based on intelligent technologies including artificial neural networks, fuzzy systems, methods from machine learning, parts of learning theory and evolutionary computing which have been especially successful in applications where input-output data can be collected but the underlying physical model is unknown. The incorporation of intelligent technologies has been used in the conception and design of complex systems in which analytical and expert systems techniques are used in combination. Viewed from a much broader perspective, the above mentioned intelligent technologies are constituents of a very active research area known under the names of soft computing, computational intelligence or hybrid intelligent systems. However, hybrid soft computing frameworks are relatively young, even comparing to the individual constituent technologies, and a lot of research is required to

understand their strengths and weaknesses. Nevertheless hybridization and combination of intelligent technologies within a flexible open framework seem to be the most promising direction in achieving the truly smart and adaptive systems today. Despite all the challenges it is unquestionable that smart adaptive intelligent systems and intelligent technology have started to have a huge impact on our everyday life and many applications can already be found in various commercially available products as illustrated in the recent report compiled by one of the world's leading think tank advanced technology organisations and very suggestively titled: "Get smart: How intelligent technology will enhance our world".

Design of Experiments in Computational Intelligence: On the Use of Statistical Inference

Salvador García and Francisco Herrera

University of Granada, Department of Computer Science and Artificial Intelligence,
E.T.S.I. Informática, 18071 Granada, Spain
{salvag1,herrera}@decsai.ugr.es

Abstract. The experimental analysis on the performance of a proposed method is a crucial and necessary task to carry out in a research. In this contribution we focus on the use of statistical inference for analyzing the results obtained in a design of experiment within the field of computational intelligence. We present some studies involving a set of techniques in different topics which can be used for doing a rigorous comparison among the algorithms studied in an experimental comparison.

Particularly, we study whether the sample of results from multiple trials obtained by the run of several algorithms checks the required conditions for being analyzed through parametric tests. In most of the cases, the results indicate that the fulfillment of these conditions are problem dependent and indefinite, which justifies the need of using nonparametric statistics in the experimental analysis. We show a case study which illustrates the use of nonparametric tests and finally we give some guidelines on the use of nonparametric statistical tests.

1 Introduction

It is not possible to find one algorithm being better in behaviour for any problem, as the "no free lunch" theorem suggests [17,18]. On the other hand, we know that we dispose of several degrees of knowledge associated with the problem which we expect to solve, and this is not the same to work without knowledge on the problem than to work with partial knowledge on the problem. This knowledge allows us to design algorithms with specific properties that can make them more suitable to the solution of the problem.

Having the previous premise in mind, the question about deciding when an algorithm is better than other one is suggested. This fact has recently conditioned the growing interest in the analysis of experiments in the field of computational intelligence. This interest has brought the use of statistical inference in the analysis of empirical results obtained by the algorithms. Inferential statistics provide of how well a sample of results supports a certain hypothesis and whether the conclusions achieved could be generalized beyond what was tested.

Statistical analysis is highly demanded in any research work and thus we can find recent studies that propose some methods for conducting comparisons among various approaches [8,7,10]. Mainly, two types of statistical tests exist

in the literature: parametric tests and nonparametric tests. The decision on the use of the former or the latter may depend on the properties of the sample of results to be analyzed. The use of parametric tests is only adequate when the sample of results fulfills three required conditions: independency, normality and homoscedasticity [15]. In this contribution we will show that these conditions are usually not checked by the parametric statistical procedures.

The analysis of results can be done following two alternatives: single-problem analysis and multiple-problem analysis. The first one corresponds to the study of the performance of several algorithms over a unique problem. The second one will suppose the study of several algorithms over more than one problem simultaneously, assimilating the fact that each problem has a degree of difficulty and that the results obtained among different problems are not comparable. The single-problem analysis is well-known and is usually found in specialized literature [8]. Although the required conditions for using parametric statistics are usually not fulfilled, a parametric statistical study could obtain similar conclusions to a non-parametric one. However, in a multiple-problem analysis, a parametric test may reach erroneous conclusions.

This paper is interested in the study of the most appropriate statistical techniques for analyzing a comparison among several computational intelligence algorithms. We will center on the study of the fulfillment of the required conditions for a safe usage of parametric tests and then we will present some non-parametric statistical procedures for making pairwise comparisons and multiple comparisons of algorithms. This study will be illustrated by three case studies: analysis of Genetics-Based Machine Learning (GBML) algorithms' performance in classification, analysis of the behaviour of some Neural Networks (NNs) [13] in classification and analysis of real parameters Evolutionary Algorithms (EAs) [5,6] for continuous optimization.

In order to achieve this objective, the contribution is organized as follows. Section 2 presents the required conditions for a safe use of parametric tests and conducts an empirical study involving the three case studies mentioned above. Some case studies for explaining the use of nonparametric are depicted in Section 3. Section 4 enumerates several considerations to take into account on the use of nonparametric test. Finally, the conclusions are remarked in Section 5.

2 Previous Conditions for a Safe Usage of Parametric Tests

In this section, we will describe and analyze the conditions that must be satisfied for the safe usage of parametric tests (Subsection 2.1). In the analysis, we will show some case studies collecting the overall set of results obtained by an EA for continuous optimization (*BLX-MA* [11]), a NN for classification (Radial Basis Function Network, RBFN [13]) and a GBML method for classification (XCS [16]). With them, we will firstly analyze the indicated conditions over the complete sample of results for each function / data set, in a single-problem analysis (see Subsection 2.2). Finally, we will consider the average results for each

function / data set to composite a sample of results for each one of the three algorithms. Finally, with the EA for continuous optimization, we will check again the required conditions for the safe use of parametric test in a multiple-problem scheme (see Subsection 2.3).

2.1 Conditions for the Safe Use of Parametric Tests

In [15], the distinction between parametric and nonparametric tests is based on the level of measure represented by the data which will be analyzed. In this way, a parametric test uses data composed by real values.

The latter does not imply that when we always dispose of this type of data, we should use a parametric test. There are other initial assumptions for a safe usage of parametric tests. The non fulfillment of these conditions might cause a statistical analysis to lose credibility.

In order to use the parametric tests, it is necessary to check the following conditions [15][20]:

- Independence: In statistics, two events are independent when the fact that one occurs does not modify the probability of the other one occurring.
- Normality: An observation is normal when its behaviour follows a normal or Gauss distribution with a certain value of average μ and variance σ . A normality test applied over a sample can indicate the presence or absence of this condition in observed data. We will use D’Agostino-Pearson’s normality test: It first computes the skewness and kurtosis to quantify how far from Gaussian the distribution is in terms of asymmetry and shape. It then calculates how far each of these values differs from the value expected with a Gaussian distribution, and computes a single p -value from the sum of these discrepancies.
- Heteroscedasticity: This property indicates the existence of a violation of the hypothesis of equality of variances. Levene’s test is used for checking whether or not k samples present this homogeneity of variances (homoscedasticity).

In our case, it is obvious the independence of the events given that they are independent runs of the algorithm with randomly generated initial seeds. In the following, we will show some case studies of the normality analysis on single-problem and multiple-problem analysis, and heteroscedasticity analysis by means of Levene’s test.

2.2 On the Study of the Required Conditions over Single-Problem Analysis

With the samples of results obtained from running 25 times the algorithm *BLX-MA* for each function, we can apply statistical tests for determining whether it checks or not the normality and homoscedasticity properties. We have seen before that the independence condition is easily satisfied in this type of experiments [10].

All the tests used in this section will obtain the p -value associated, which represents the dissimilarity of the sample of results with respect to the normal

Table 1. Test of Normality of D’Agostino-Pearson

	f1	f2	f3	f4	f5	f6	f7	f8	f9
BLX-MA	* (.00)	* (.00)	(.22)	* (.00)	* (.00)	* (.00)	(.19)	(.12)	* (.00)
	f10	f11	f12	f13	f14	f15	f16	f17	f18
BLX-MA	(.89)	* (.00)	* (.03)	(.38)	(.16)	* (.00)	(.21)	(.54)	* (.04)
	f19	f20	f21	f22	f23	f24	f25		
BLX-MA	* (.00)	* (.00)	(.25)	* (.00)	* (.00)	* (.00)	(.20)		

shape. Hence, a low p -value points out a non-normal distribution. In this study, we will consider a level of significance $\alpha = 0.05$, so a p -value greater than α indicates that the condition of normality is fulfilled. All the computations have been performed by the statistical software package SPSS.

Table 1 shows the results where the symbol “*” indicates that the normality is not satisfied and the p -value in brackets.

In addition to this particular study, we show the sample distribution in three cases, with the objective of illustrating representative cases in which the normality tests obtain different results.

From Figure 1 to Figure 3, different examples of graphical representations of histograms and Q-Q graphics are shown. A histogram represents a statistical variable by using bars, so that the area of each bar is proportional to the frequency of the represented values. A Q-Q graphic represents a confrontation between the quartiles from data observed and those from the normal distributions.

In Figure 1 we can observe a general case in which the property of abnormality is clearly presented in the case study of EAs for continuous optimization. Figures 2 and 3 illustrate similar cases of abnormality presented in the classification field, employing different evaluation measures (accuracy in the first case and Cohen’s Kappa 3 in the second case).

With respect to the study of homoscedasticity property, Table 2 shows the results by applying Levene’s test, where the symbol “*” indicates that the variances of the distributions of the different algorithms for a certain function are not

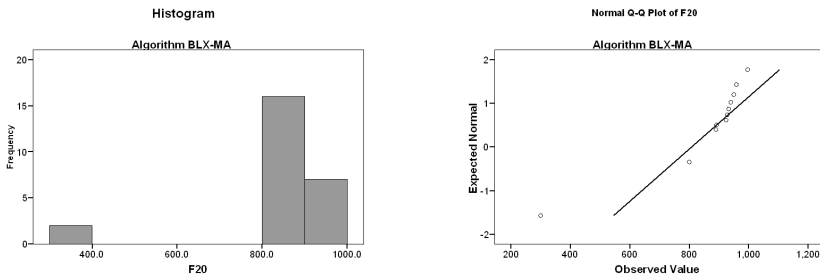


Fig. 1. Example of non-normal distribution: Function f20 and BLX-MA algorithm: Histogram and Q-Q Graphic

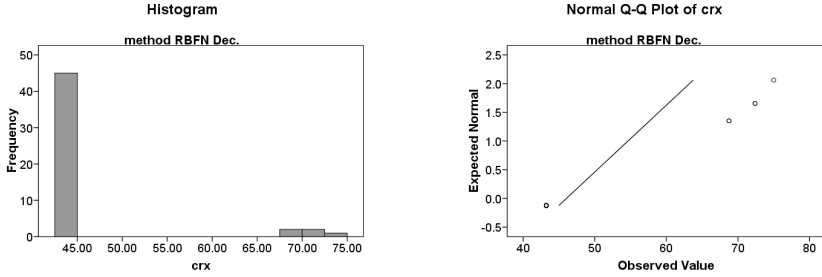


Fig. 2. Results of RBFN over crx data set using Hold-out Validation: Histogram and Q-Q Graphic

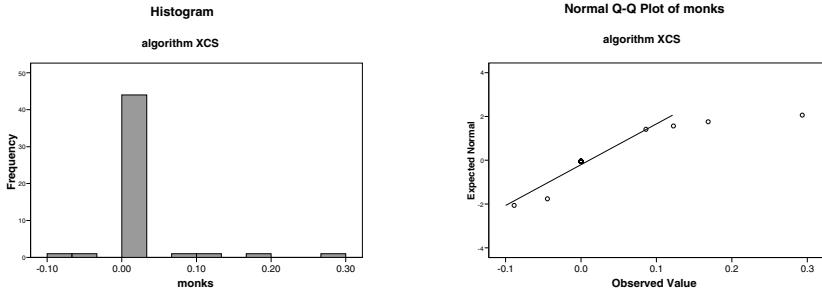


Fig. 3. Results of XCS over monks data set using 10-fold cross validation: Histogram and Q-Q Graphic

Table 2. Test of Heteroscedasticity of Levene (based on means)

	f1	f2	f3	f4	f5	f6	f7	f8	f9
LEVENE	(.07)	(.07)	* (.00)	* (.04)	* (.00)	* (.00)	* (.00)	(.41)	* (.00)
	f10	f11	f12	f13	f14	f15	f16	f17	f18
LEVENE	(.99)	* (.00)	(.98)	(.18)	(.87)	* (.00)	* (.00)	(.24)	(.21)
	f19	f20	f21	f22	f23	f24	f25		
LEVENE	* (.01)	* (.00)	* (.01)	(.47)	(.28)	* (.00)	* (.00)		

homogeneities (we reject the null hypothesis at a level of significance $\alpha = 0.05$). The use of Levene’s test requires at least two algorithms, so we show a case studied in [10] where two algorithms are studied.

Clearly, in both cases, the non fulfillment of the normality and homoscedasticity conditions is perfectible. In [10], we can find a complete experimental study on the CEC’2005 Special Session of Evolutionary Algorithms for Real Parameters Optimization, in which the statistical properties and some guidelines to use nonparametric test are given.

2.3 On the Study of the Required Conditions over Multiple-Problem Analysis

When tackling a multiple-problem analysis, the data to be used is an aggregation of results obtained from individual algorithms' runs. In this aggregation, there must be only a result representing a problem. This result could be obtained through averaging results for all runs or something similar, but the procedure followed must be the same for each problem; i.e., in this paper we have used the average of the 25 runs of an EA in each function. The size of the sample of results to be analyzed, for each algorithm, is equal to the number of problems. In this way, a multiple-problem analysis allows us to compare two or more algorithms over a set of problems simultaneously.

In the case study of EAs for continuous optimization, we can use the results published in the CEC'2005 Special Session to perform a multiple-problem analysis. Indeed, we will follow the same procedure as the previous subsection. We will analyze the required conditions for the safe usage of parametric tests over the sample of results obtained by averaging the error rate on each function.

D'Agostino Pearson's test indicates us the absence of normality over the sample of results obtained by by *BLX-MA* with a $p = 0.00$. Figure 4 represents the histograms and Q-Q plots for such sample.

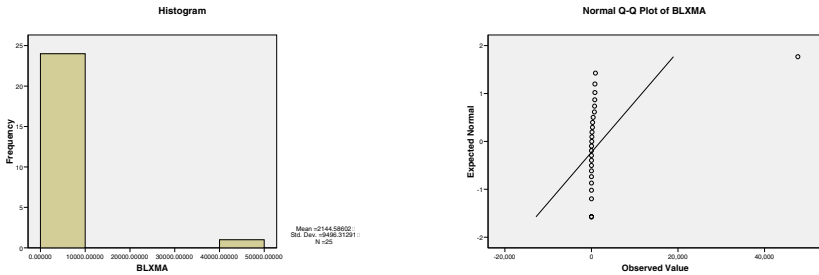


Fig. 4. BLX-MA algorithm: Histogram and Q-Q Graphic

As we can see, the normality condition is not satisfied because the sample of results is composed by 25 average error rates computed in 25 different problems. This fact may induce us to using nonparametric statistics for analyzing the results in multiple-problems. Nonparametric statistics do not need prior assumptions related to the sample of data for being analyzed and, in the example shown in this section, we have seen that they could obtain reliable results.

3 Using Nonparametric Tests: Two Case Studies

This section presents two case studies on the use of Wilcoxon's test as a pairwise comparison nonparametric test and Friedman's test and post-hoc procedures as multiple comparison nonparametric tests. The description of these tests is available in [7,10].

Firstly, we will perform the statistical analysis by means of pairwise comparisons by using the results of performance based on accuracy obtained by several GBML algorithms (Pitts-GIRLA [4], XCS [16], GASSIST-ADI [2], HIDER [1]) over 14 classification data sets.

In order to compare the results between two algorithms and to stipulate which one is the best, we can perform Wilcoxon signed-rank test for detecting differences in both means. This statement must be enclosed by a probability of error, that is the complement of the probability of reporting that two systems are the same, called the p -value [20]. The computation of the p -value in Wilcoxon’s distribution could be carried out by computing a normal approximation [15]. This test is well known and it is usually included in standard statistics packages (such as SPSS, R, SAS, etc.).

Table 3 shows the results obtained in all possible comparisons among the GBML algorithms in accuracy. We stress in bold the winner algorithm in each row when the p -value associated is below 0.05.

Table 3. Wilcoxon’s test applied over the all possible comparisons between the 4 algorithms in accuracy

Comparison	Classification rate		
	R^+	R^-	p -value
Pitts-GIRLA - XCS	0.5	104.5	0.001
Pitts-GIRLA - GASSIST-ADI	2	103	0.002
Pitts-GIRLA - HIDER	1	104	0.001
XCS - GASSIST-ADI	81	24	0.074
XCS - HIDER	53	52	0.975
GASSIST-ADI - HIDER	13	92	0.013

In order to illustrate the use of the Friedman test, we resort to the results offered by NN methods in classification. In this case study, 16 classification data sets have been used for comparing 7 NN algorithms (C-SVM, NU-SVM [9], RBFN, RBFN Decremental, RBFN Incremental, MLP and LVQ [13]). In Table 4 we show the results of applying the tests of Friedman and Iman-Davenport in order to detect whether differences in the results exist. In bold appears the highest value of the compared ones, and if it is the corresponding statistical (column named "Friedman Value" or "Iman-Davenport Value"), then the null hypothesis is rejected. In case contrary, the null hypothesis is not rejected, so the results are not significantly different. Both test are applied with a level of confidence $\alpha = 0.05$.

In our case, both Friedman’s and Iman-Davenport’s tests indicate that significant differences in the results are found. Due to these results, a post-hoc

Table 4. Results for Friedman and Iman-Davenport tests

Partitioning procedure	Friedman Value	Value of χ^2	Iman-Davenport Value	F_F Value
Hold-Out Validation	21.609	12.592	4.357	2.201

Table 5. Adjusted p -values in Hold-Out Validation in NN algorithms (C-SVM is the control)

i	algorithm	unadjusted p	p_{Bonf}	p_{Holm}	p_{Hoch}
1	RBFN Dec.	$2.505 \cdot 10^{-5}$	$1.503 \cdot 10^{-4}$	$1.503 \cdot 10^{-4}$	$1.503 \cdot 10^{-4}$
2	LVQ	$9.191 \cdot 10^{-4}$	0.00551	0.0046	0.0046
3	RBFN	0.00475	0.02853	0.01902	0.01902
4	MLP	0.01578	0.09466	0.04733	0.04733
5	NU-SVM	0.07181	0.43088	0.14363	0.07851
6	RBFN Inc.	0.07851	0.47108	0.14363	0.07851

statistical analysis is required. In this analysis, we choose the best performing method, *C-SVM*, as the control method for being compared with the rest of algorithms.

We will apply basic, such as Bonferroni-Dunn, and advanced procedures, such as Holm's and Hochberg's, for comparing the control algorithm with the rest of algorithms. Table 5 shows all the adjusted p -values [19] for each comparison which involves the control algorithm. The p -value is indicated in each comparison and we stress in bold the algorithms which are worse than the control, considering a level of significance $\alpha = 0.05$.

4 Considerations on the Use of Nonparametric Tests in Computational Intelligence

In this section, we suggest some aspects and details about the use of nonparametric statistical techniques:

- A multiple comparison of various algorithms must be carried out first by using a statistical method for testing the differences among the related samples means, that is, the results obtained by each algorithm. Once this test rejects the hypothesis of equivalence of means, the detection of the concrete differences among the algorithms can be done with the application of post-hoc statistical procedures, which are methods used for comparing a control algorithm with two or more algorithms.
- Holm's procedure can always be considered better than Bonferroni-Dunn's one, because it appropriately controls the Family Wise Error Rate (FWER) [14] and it is more powerful than the Bonferroni-Dunn's. We strongly recommend the use of Holm's method in a rigorous comparison. Nevertheless, the results offered by the Bonferroni-Dunn's test are suitable to be visualized in graphical representations.
- Hochberg's procedure is more powerful than Holm's. The differences reported between it and Holm's procedure are in practice rather small, but in this paper, we have shown a case in which Hochberg's method obtains lower p -values than Holm's (see Table 4). We recommend the use of this test together with Holm's method.

- Although Wilcoxon’s test and the remaining post-hoc tests for multiple comparisons belong to the nonparametric statistical tests, they operate in a different way. The main difference lies in the computation of the ranking. Wilcoxon’s test computes a ranking based on differences between functions independently, whereas Friedman and derivative procedures compute the ranking between algorithms.
- In relation to the sample size (number of problems when performing Wilcoxon’s or Friedman’s tests in multiple-problem analysis), there are two main aspects to be determined. Firstly, the minimum sample considered acceptable for each test needs to be stipulated. There is no established agreement about this specification. Statisticians have studied the minimum sample size when a certain power of the statistical test is expected [12]. In our case, the employment of a, as large as possible, sample size is preferable, because the power of the statistical tests (defined as the probability that the test will reject a false null hypothesis) will increase. Moreover, in a multiple-problem analysis, the increasing of the sample size depends on the availability of new instances of the problem (which should be well-known in the problem in question). Secondly, we have to study how the results are expected to vary if there was a larger sample size available. In all statistical tests used for comparing two or more samples, the increasing of the sample size benefits the power of the test. In the following items, we will state that Wilcoxon’s test is less influenced by this factor than Friedman’s test. Finally, as a rule of thumb, the number of problems (N) in a study should be $N = a \cdot k$, where k is the number of algorithms to be compared and $a \geq 2$.
- Taking into account the previous observation and knowing the operations performed by the nonparametric tests, we can deduce that Wilcoxon’s test is influenced by the number of problems used. On the other hand, both the number of algorithms and problems are crucial when we refer to the multiple comparisons tests (such as Friedman’s test), given that all the critical values depend on the value of N . However, the increasing / decreasing of the number of problems rarely affects in the computation of the ranking. In these procedures, the number of problems used is an important factor to be considered when we want to control the FWER.
- An appropriate number of algorithms in contrast with an appropriate number of problems are needed to be used in order to employ each type of test. The number of algorithms used in multiple comparisons procedures must be lower than the number of problems. In the study of the CEC’2005 Special Session, we can appreciate the effect of the number of functions used whereas the number of algorithms stays constant [10]. See, for instance, the p -value obtained when considering the f15-f25 group and all functions. In the latter case, p -values obtained are always lower than in the first one, for each testing procedure. In general, p -values are lower agreeing with the increasing of the number of functions used in multiple comparison procedures; therefore, the differences among the algorithms are more detectable.
- The previous statement may not be true in Wilcoxon’s test. The influence of the number of problems used is more noticeable in multiple comparisons procedures than in Wilcoxon’s test.

5 Concluding Remarks

The present work studies the use of statistical techniques for analyzing some cases of computational intelligence algorithms in continuous optimization and classifications problems.

The need for using nonparametric tests for results's analysis is very clear, since the initial conditions required for obtaining safe conclusions from parametric tests are not met.

On the use of nonparametric tests, we have shown an example of performing pairwise comparisons with the Wilcoxon test and multiple comparisons with the Friedman test and post-hoc procedures. Some recommendations on the use of these statistical procedures in computational intelligence studies are given.

Acknowledgement. This work was supported by TIN2005-08386-C05-01.

References

1. Aguilar-Ruiz, J.S., Giráldez, R., Riquelme, J.C.: Natural encoding for evolutionary supervised learning. *IEEE Transactions on Evolutionary Computation* 11(4), 466–479 (2007)
2. Bacardit, J.: Pittsburgh genetic-based machine learning in the data mining era: representations, generalization and run-time. Phd Thesis, Dept. Computer Science University Ramon Llull, Barcelona, Spain (2004)
3. Ben-David, A.: A lot of randomness is hiding in accuracy. *Engineering Applications of Artificial Intelligence* 20, 875–885 (2007)
4. Corcoran, A.L., Sen, S.: Using real-valued genetic algorithm to evolve rule sets for classification. In: *Proceedings of the IEEE Conference on Evolutionary Computation*, pp. 120–124 (1994)
5. Eiben, A.E., Smith, J.E.: *Introduction to Evolutionary Computing*. Springer, Heidelberg (2003)
6. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
7. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
8. Dietterich, T.G.: Approximate Statistical Test For Comparing Supervised Classification Learning Algorithms. *Neural Computation* 10(7), 1895–1923 (1998)
9. Fan, R.-E., Chen, P.-H., Lin, C.-J.: Working set selection using second order information for training SVM. *Journal of Machine Learning Research* 6, 1889–1918 (2005)
10. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of nonparametric tests for analyzing the evolutionary algorithm's behaviour: a case study on the CEC 2005 Special Session on Real Parameter Optimization. *Journal of Heuristics* (in press, 2008) DOI: 10.1007/s10732-008-9080-4
11. Molina, D., Herrera, F., Lozano, M.: Adaptive local search parameters for real-coded memetic algorithms. In: *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, pp. 888–895 (2005)
12. Noether, G.E.: Sample size determination for some common nonparametric tests. *Journal of American Statistical Association* 82(398), 645–647 (1987)

13. Rojas, R., Feldman, J.: *Neural Networks: A systematic introduction*. Springer, Heidelberg (1996)
14. Shaffer, J.P.: Multiple hypothesis testing. *Annual Review of Psychology* 46, 561–584 (1995)
15. Sheskin, D.J.: *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press, Boca Raton (2006)
16. Wilson, S.W.: Classifier fitness based on accuracy. *Evolutionary Computation* 3(2), 149–175 (1995)
17. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1(1), 67–82 (1997)
18. Wolpert, D.H.: The supervised learning no-free-lunch theorems. In: *Proc. 6th On-line World Conference on Soft Computing in Industrial Applications* (2001)
19. Wright, S.P.: Adjusted p-values for simultaneous inference. *Biometrics* 48, 1005–1013 (1992)
20. Zar, J.H.: *Biostatistical Analysis*. Prentice-Hall, Englewood Cliffs (1999)

Nonlinear Principal Manifolds – Adaptive Hybrid Learning Approaches

Hujun Yin

The University of Manchester, Manchester, M60 1QD, UK
h.yin@manchester.ac.uk

Abstract. Dimension reduction has long been associated with retinotopic mapping for understanding cortical maps. Multisensory information is processed, fused, fed and mapped to a 2-D cortex in a near-optimal information preserving manner. Data projection and visualization, inspired by this mechanism, are playing an increasingly important role in many computational applications such as cluster analysis, classification, data mining, knowledge management and retrieval, decision support, marketing, image processing and analysis. Such tasks involving either visual and spatial analysis or reduction of features or volume of the data are essential in many fields from biology, neuroscience, decision support, to management science. The topic has recently attracted a great deal of attention. There have been considerable advances in methodology and techniques for extracting nonlinear manifold as to reduce data dimensionality and a number of novel methods have been proposed from statistics, geometry theory and adaptive neural networks. Typical approaches include multidimensional scaling, nonlinear PCA and principal curve/surface. This paper provides an overview on this challenging and emerging topic. It discusses various recent methods such as self-organizing maps, kernel PCA, principal manifold, isomap, local linear embedding, Laplacian eigenmap and spectral clustering, and many of them can be seen as a combined, adaptive learning framework. Their usefulness and potentials will be presented and illustrated in various applications.

1 Introduction

Many computational intelligence tasks involve analysis and exploration of a vast amount of data in order to extract useful information and discover meaningful patterns and rules. Clustering the data sets and projecting them onto a lower dimensional space are common practices. Dimensionality reduction has long been associated with retinotopic mapping for understanding cortical maps [10]. Multisensory information is processed, fused, fed and mapped to a 2-D cortex in a near-optimal information preserving manner. There exists an increasing demand in an increasing number of fields for dimensionality reduction, range from high-throughput bioinformatics and neuroinformatics, web information science, data mining, knowledge management and retrieval, decision support, marketing, to computer vision and image processing. Seeking a suitable, smaller and featured representation of a raw data set can make data analysis and pattern recognition easier and efficient. Projecting and abstracting the data onto their underlying or principal subspaces can help reduce the number of

features, identify latent variables, detect intrinsic structures, and facilitate visualization of variable interactions. With the ever fast increasing data quantity, complexity and dimensionality in many computational tasks, more sophisticated methods are required and being developed. A great deal of research has been devoted to this emerging topic on improving and extending the classic methods such as principal component analysis (PCA) and multidimensional scaling (MDS) [58, 16].

PCA projects data set onto its principal directions represented by the orthogonal eigenvectors of the covariance matrix of the data. It has long been used in reducing the number of variables and visualizing data in scatter plots or linear subspaces. Singular value decomposition and factor analysis are adopted to perform the task due to various advantages such as direct operation on a data matrix, stable results even when the data matrix is ill-conditioned, and decomposition at both feature and data levels. The linearity of PCA, however, limits its power for practical, increasingly large and complex data sets, because it cannot capture nonlinear relationships defined by beyond second order statistics. Extension to nonlinear projection, in principle, can tackle practical problems better; yet a unique solution is still to be defined [35]. Various nonlinear methods along this line have been proposed, such as the auto-associative networks [28], generalized PCA [22], kernel PCA [47], principal curve and surface [19] and local linear embedding (LLE) [44].

MDS is another popular methodology that projects high dimensional data points onto a low (often two) dimensional plane by preserving as close as possible the inter-point distances (or pair-wise dissimilarities) [8]. Metric MDS generalises classical MDS by minimizing a stress function. The mapping is generally nonlinear and can reveal the overall structure of the data. Sammon mapping [45] is a widely known example. In contrast to metric MDS, nonmetric MDS finds a monotonic relationship (instead of metric ones) between the dissimilarities of the data points in the data space and those of their corresponding coordinates in the projected space. More general weighting schemes have been proposed recently and the resulting MDS is called generalized MDS [6]. Isomap [49] applies scaling on geodesic instead of Euclidean distances. MDS methods are generally point-to-point mappings and do not provide a generalizing mapping function or a generative manifold.

Adaptive neural networks present alternative approaches to nonlinear data projection and dimension reduction. They can provide (implicit) generalizing mapping functions. Early examples include feed-forward neural network based mapping [36] and radial-basis-function based MDS [32]. The self-organizing map (SOM) [24, 27] has become a widely used method for data visualization. Self-organization is a fundamental pattern recognition process, in which intrinsic inter- and intra-pattern relationships within the sensory data are learnt. Kohonen's SOM is a simplified, abstracted version of Willshaw and von der Malsburg's retinotopic mapping model [52]. Modelling and analyzing such mappings are important to understanding how the brain perceives, encodes, recognizes, and processes the patterns it receives and thus, if somewhat indirectly, are beneficial to machine-based pattern recognition. The SOM is topology-preserving vector quantization. The topology-preserving property is utilized to extract and visualize relative mutual relationships among the data. Many variants and extensions have since been proposed, including the visualization induced SOM (ViSOM) [54]. The ViSOM regularizes the inter-neuron distances within a neighbourhood so to preserve (local) distances on the map. The SOM and some variants have been linked

with principal curve and surface (e.g. [43]; [55]). It has also been widely observed that SOMs produce a similar effect to MDS. In fact it has been argued that the SOM is closer to MDS than to principal curve/surface [42]. However, the exact connection between SOMs and MDS has not been established. Further analysis has shown the exact connections [58, 59]. A growing variant of metric preserving ViSOM has been proposed for embedding nonlinear manifolds.

The paper is organized as follows. Section 2 provides a review on existing approaches on nonlinearizing PCA. Section 3 describes MDS and recent related approaches. Principal manifold (curve and surface) is explained in the next section. Section 5 focuses on SOM, its variants and related adaptive approaches in providing nonlinear, generative manifolds of data sets, as well as their relationship with the previous approaches. Section 6 wraps up with a set of various computational applications and potentials of these methods, followed by a brief conclusion.

2 Nonlinear PCA Approaches

PCA is a classic linear projection aiming at finding orthogonal, principal directions of a data set, along which the data exhibits the largest variances. PCA is obtained by solving an eigenvalue problem of the covariance matrix \mathbf{C} of the data set, $\lambda\mathbf{V}=\mathbf{C}\mathbf{V}$, where columns of \mathbf{V} are eigen vectors. By discarding the minor components, PCA can effectively reduce the number of variables and display the dominant ones in a linear, low dimensional subspace. It is the optimal linear projection in the sense of the mean-square error between original points and projected ones, i.e.

$$\min_{\mathbf{x}} \sum \left(\mathbf{x} - \sum_{j=1}^m (\mathbf{q}_j^T \mathbf{x}) \mathbf{q}_j \right)^2 \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the n -dimensional data and $\{\mathbf{q}_j, j=1,2, \dots, m, m \leq n\}$ are orthogonal eigenvectors representing principal directions. They are the first m principal eigenvectors of the covariance matrix of the data set. The second term in the above bracket is the reconstruction or projection of \mathbf{x} on these eigenvectors. The term $\mathbf{q}_j^T \mathbf{x}$ represents the projection of \mathbf{x} onto the j -th principal dimension. Efficient and robust statistical methods exist for solving eigenvector problems. Several adaptive learning algorithms have also been proposed for performing PCA such as, the subspace network [41] and the generalized Hebbian algorithm [46]. The limitation of linear PCA is obvious, as it cannot capture nonlinear relationships defined by higher than the second order statistics. If the input dimension is much higher than two, the projection onto linear principal plane will provide limited visualization power.

Nonlinear extension of PCA can be intuitively approached using combined or piecewise local PCA models. That is, the entire input space is segmented (e.g. using a clustering algorithm) into non-overlapping regions and a local PCA can be formed in each region. However, the extension to nonlinear PCA is not unique due to the lack of a unified mathematical structure and an efficient and reliable algorithm, and in some cases due to excessive freedom in selection of representative basis functions [35][22]. Existing methods include the five-layer feedforward associative network [28] and kernel PCA [47]. The first three layers of the associative network project the original

data on to a curve or surface, providing an activation value for the bottleneck node. The last three layers define the curve and surface. The weights of the associative network are determined by minimizing the following objective function,

$$\min_{\mathbf{x}} \sum_{\mathbf{x}} \|\mathbf{x} - \mathbf{f}\{s_f(\mathbf{x})\}\|^2 \quad (2)$$

where $\mathbf{f}: \mathbb{R}^1 \rightarrow \mathbb{R}^n$ (or $\mathbb{R}^2 \rightarrow \mathbb{R}^n$), the function modelled by the last three layers, defines a curve (or a surface), $s_f: \mathbb{R}^n \rightarrow \mathbb{R}^1$ (or $\mathbb{R}^n \rightarrow \mathbb{R}^2$), the function modelled by the first three layers, defines the projection index.

The kernel-based PCA uses nonlinear mapping and kernel functions to generalise PCA to nonlinear. The nonlinear function $\Phi(\mathbf{x})$ maps data onto high-dimensional feature space, where the standard linear PCA can be performed via kernel functions: $k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$. The projected covariance matrix is then,

$$Cov = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T \quad (3)$$

The standard linear eigenvalue problem can now be written as $\lambda \mathbf{V} = \mathbf{K} \mathbf{V}$, where the columns of \mathbf{V} are the eigenvectors and \mathbf{K} is a $N \times N$ matrix with elements as kernels $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$.

Local linear embedding (LLE) [44] is another way of forming nonlinear principal subspace. The local linearity is defined on a local neighbourhood, ε ball or k nearest neighbors. Then the linear contributions or weightings, W_{ij} , of these neighboring points are calculated through,

$$\min_i \sum_i \left\| \mathbf{x}_i - \sum_{j=1} W_{ij} \mathbf{x}_j \right\|^2 \quad (4)$$

The embedding is computed via,

$$\min_i \sum_i \left\| Y_i - \sum_{j=1} W_{ij} Y_j \right\|^2 \quad (5)$$

where Y is the embedding coordinates. Both steps can be solved by the least square method or eigen problems.

Recent Laplacian eigenmap [3] forms a local linear mapping by converting the problem to a generalized eigen problem and the solution becomes easily traceable. First, the weightings (of local neighboring points) or heat kernels are constructed,

$$W_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}\right) \quad (6)$$

Then the embedding is computed via the generalized eigen problem,

$$L \mathbf{f} = \lambda D \mathbf{f} \quad (7)$$

where $D_{ii} = \sum_j W_{ji}$ and $L = D - W$.

The data is then projected to the subspace spanned by the principal eigen functions ($\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m$). This approach is also related to spectral clustering [51].

These nonlinear PCA methods, though with different approaches, have many common characters. They are all defined on a local neighbourhood so transforming global structure to local linear structures. That is, the manifold is constructed on local (linear) graphs. It has been shown that these methods are closely related [18], or they can be described under the regularization theory [48].

3 Multidimensional Scaling Approaches

Multidimensional scaling (MDS) is a traditional subject related to dimension reduction and data visualization. MDS aims to project or embed high dimensional data points onto a low dimensional (often 2- or 3-D) plane by preserving as closely as possible inter-point metrics [8]. The projection is generally nonlinear and can reveal the overall structure of the data. Let δ_{ij} denote the dissimilarity between data points \mathbf{x}_i and \mathbf{x}_j . δ_{ij} is often calculated (but not necessarily) by the Euclidean distance of data vectors $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$. Let \mathbf{y}_i and \mathbf{y}_j are mapped points or coordinates of points i and j in the visual space, then the distance between mapped points is $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$.

There are three types of MDS for different purposes and effects, classical, metric and nonmetric MDS. *Classical MDS* looks for a configuration so that the distances between projected points match the original dissimilarities, i.e. $d_{ij} = \delta_{ij}, \forall i, j$. *Metric MDS* seeks that dissimilarities are proportional to the distances of projected points, $d_{ij} = f(\delta_{ij}), \forall i, j$; where f is a continuous, monotonic function, or a metric transformation function, that transforms dissimilarities to distance metrics. In practice, exact dissimilarity-distance matches may not be possible due to data noise and imprecision, the equality is replaced by approximation, i.e. “ \approx ”, meaning “as equal as possible” [5].

An MDS configuration is often sought by minimising the following general cost, or the raw *Stress*, function,

$$S = \sum_{i,j} (f(\delta_{ij}) - d_{ij})^2 \quad (8)$$

where f is the metric transformation function. In some cases, the above raw stress is normalised by $\sum_{i,j} d_{ij}^2$ or $\sum_{i,j} \delta_{ij}^2$ to give a relative reading of the overall stress.

Other normalisation scheme is also available. For example, In Sammon mapping [45] intermediate normalisation (pair-wise distance of original space) is used to preserve good local distributions and at the same time maintain a global structure. The Newton optimisation method is used to recursively solve the optimal configuration.

For general metric MDS, especially when original dissimilarities need to be transformed to a distance like form, f is a monotonic transformation function, e.g. a linear function. For classical MDS and many cases of metric MDS, f is simply the identity function, the stress becomes,

$$S = \sum_{i,j} (\delta_{ij} - d_{ij})^2 \quad (9)$$

That is, metric MDS configuration tries to preserve, as well as possible, the pair-wise distances of original data points on the projected space.

Nonmetric MDS deals with rank order of the dissimilarities and seeks a configuration such that distances between pairs of mapped points match order-wise “as well as possible” the original dissimilarities. That is, a nonmetric MDS looks for projection function f that is a monotonic function and satisfies,

$$\text{if } \delta_{ij} \leq \delta_{kl}, \text{ then } d_{ij} \leq d_{kl}, \forall i, j, k, l \quad (10)$$

Nonmetric MDS produces an ordinal scaling rather than a metric one.

MDS relies on an optimization algorithm to search for a configuration that gives as low stress as possible. Inevitably, various computational problems such as local minima and divergence may occur to the process. The methods are often computationally intensive, especially with large number of data points. The final solution depends on the starting configuration and parameters used in the algorithm.

Another major drawback of MDS is the lack of an explicit projection function, as MDS is usually a point-to-point mapping and cannot *naturally* accommodate new data points. Thus for any new input data, the mapping has to be recalculated based on all available data. Although some methods have been proposed to accommodate the new arrivals using triangulation [31][9], the methods are generally not adaptive.

However, such drawbacks can be overcome by implementing or parametrizing MDS using neural networks. In [36], a feed-forward network is used to parametrize the Sammon mapping and a unsupervised training methods has been derived to train the network. The derivation is similar to the back-propagation algorithm, by minimizing the Sammon stress instead of the total errors between desired and actual output. The network takes a pair of input points at each time in training. An evaluation has to be carried out, using all the data points, after a fixed number of iterations. In [32] a radial basis function (RBF) network is used to minimize a simple stress function, so to perform MDS. In [49], Isomap was proposed to use geodesic (curvature) distance for better scaling of nonlinear manifolds. Instead of the direct Euclidean distance, the geodesic distance along the manifold is calculated (or cumulated) via neighborhood graphs or neighboring points. Selecting a suitable neighborhood size can be a difficult task and often needs cross-validation procedure. The Isomap has been reported being unstable [1].

It is worth noting that while MDS general minimizes the difference or squared difference between the dissimilarities in the original and mapped spaces, the c measure proposed as a unified objective for topographic mapping [15] maximizes the correlation between the two. One can argue that when the dissimilarities are normalized, the two are equivalent.

4 Principal Curve and Surface

The principal curve and surface [19][30][50] are the principled nonlinear extension of PCA. The principal curve is defined as a smooth and self-consistency curve, which does not intersect itself, passing through the middle of the data. Denote \mathbf{x} as a random vector in \mathbb{R}^n with density p and finite second moment. Let $f(\cdot)$ be a smooth unit-speed curve in \mathbb{R}^n , parameterized by the arc length ρ (from one end of the curve) over $\Lambda \in \mathbb{R}$, a closed interval.

For a data point \mathbf{x} , its projection index on f is defined as,

$$\rho_f(\mathbf{x}) = \sup_{\rho \in \Lambda} \{ \rho : \|\mathbf{x} - f(\rho)\| = \inf_{\vartheta} \|\mathbf{x} - f(\vartheta)\| \} \quad (11)$$

The curve is called self-consistent principal curve of ρ if,

$$f(\rho) = E[\mathbf{X} \mid \rho_f(\mathbf{X}) = \rho] \quad (12)$$

The principal component is a special case of the principal curves if the distribution is ellipsoidal. Although principal curves have been mainly studied, extension to higher dimension, e.g. principal surfaces or manifolds is feasible in principle. However, in practice, a good implementation of principal curves/surfaces relies on an effective and efficient algorithm. The principal curves/surfaces are more of a concept and practical algorithms are needed for implementation. The HS algorithm is a non-parametric method [19] that directly iterates the two steps of the above definition. It is similar to the standard VQ algorithm combined with some smoothing techniques when only a finite data set is available:

- *Initialization*: Choose the first linear principal component as the initial curve, $f^{(0)}(\mathbf{x})$.
- *Projection*: Project the data points onto the current curve and calculate the projections index, i.e. $\rho^{(t)}(\mathbf{x}) = \rho_{f^{(t)}}(\mathbf{x})$.
- *Expectation*: For each index, take the mean of data points projected onto it as the new curve point, i.e., $f^{(t+1)}(\rho) = E[\mathbf{X} \mid \rho_{f^{(t)}}(\mathbf{X}) = \rho]$.

The projection and expectation steps are repeated until a convergence criterion is met, for instance, when the change of the curve between iterations is below a threshold.

For a finite data set, the density is often unknown; the above expectation is replaced by a smoothing method such as the locally weighted running-line smoother or smoothing splines. For kernel regression, the smoother is,

$$f(\rho) = \frac{\sum_{i=1}^N \mathbf{x}_i \kappa(\rho, \rho_i)}{\sum_{i=1}^N \kappa(\rho, \rho_i)} \quad (13)$$

The arc length is simply computed from the line segments. There are no proofs of convergence of the algorithm, but no convergence problems have been reported, though the algorithm is biased in some cases [19]. In [2] a modified HS algorithm was proposed by taking the expectation of the residual of the projections in order to reduce the bias. An incremental principal curve was proposed in [22]. It is an incremental, or segment by segment, and arc length constrained method for practical construction of principal curves.

In [50] a semi-parametric model for the principal curve was introduced. A mixture model was used to estimate the noise along the curve; and the expectation and maximization (EM) method was employed to estimate the parameters. Other adaptive learning approaches include the probabilistic principal surfaces [7].

5 SOM-Based Approaches

Kohonen's self-organizing map (SOM) is an abstract, simplified mathematical model of the mapping between nerve sensory and cerebral cortex [24][27]. Modeling and analyzing such mappings are important to understanding how the brain perceives, encodes, recognizes, and processes the patterns it receives and thus, are also beneficial to machine-based pattern recognition. External stimuli are received by various sensory or receptive fields (e.g. visual-, auditory-, motor-, or somato-sensory) coded, combined and abstracted by the living neural networks, propagated through axons, and projected onto the cerebral cortex, often to distinct parts of cortex. Different areas of the cortex (cortical maps) respond to different sensory inputs, though many functions and actions require collective responses from various areas. Topographically ordered mappings are widely observed in the cortex. The main structures (primary sensory areas) of the cortical maps are established genetically in a predetermined manner [25]. More detailed areas (associative areas) between the primary sensory areas, however, are developed through self-organisation gradually during life and in a topographically meaningful fashion. Therefore studying such topographic projections, which had been ignored during the early period of neural network research, is undoubtedly fundamentally to understanding of the dimension-reduction mapping for effective representation of sensory information and feature extraction.

Von der Malsburg and Willshaw first developed in mathematical form the self-organising topographic mappings, mainly from two-dimensional presynaptic sheets to two-dimensional postsynaptic sheets, based on retinatopic mapping: the ordered projection of visual retina to visual cortex [52]. Kohonen abstracted this self-organising learning model and proposed a much simplified mechanism which ingeniously incorporates the Hebbian learning rule and lateral interactions [24]. This simplified model can emulate the self-organisation effect. Although the SOM algorithm was more or less proposed in a heuristic manner, it is an abstract and generalised model of the self-organisation or unsupervised learning process.

In the SOM, a set of neurons, often arranged in a 2-D rectangular or hexagonal grid or map, is used to form a discrete, topological mapping of an input space, $\mathbf{X} \in \mathbf{R}^n$. At the start of the learning, all the weights $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M\}$ are initialised to either small random numbers or some specific values (e.g. principal subspace), where \mathbf{w}_i is the weight associated to neuron i and is a vector of the same dimension, n , of the input, M is the total number of neurons. Denote \mathbf{r}_i the discrete vector defining the position (coordinates) of neuron i on the map grid. Then the algorithm iterates the following steps.

- At each time t , present an input, $\mathbf{x}(t)$, select the winner,

$$v(t) = \arg \min_{k \in \Omega} \|\mathbf{x}(t) - \mathbf{w}_k(t)\| \quad (14)$$

- Update the weights of the winner and its neighbours,

$$\Delta \mathbf{w}_k(t) = \alpha(t) \eta(v, k, t) [\mathbf{x}(t) - \mathbf{w}_k(t)] \quad (15)$$

- Repeat until the map converges.

where $\alpha(t)$ is the learning rate and $\eta(v, k, t)$ is the neighbourhood function and Ω is the set of neuron indexes. Although one can use a top-hat type of neighbourhood function, a Gaussian, $\eta(v, k, t) = e^{-\frac{d_{vk}^2}{2\sigma(t)^2}}$, is often used in practice with $\sigma(t)$ representing the effective range of the neighbourhood and $d_{vk} = \|\mathbf{r}_v - \mathbf{r}_k\|$, the distance between neurons v and k on the map grid.

The SOM was proposed to model the sensory-to-cortex mapping thus is an unsupervised, associative memory mechanism. Such a model is also related to vector quantisation (VQ) in coding terms. The SOM has been shown to be an asymptotically optimal VQ [61]. More importantly, with the neighbourhood learning, the SOM is an error tolerant VQ and Bayesian VQ [33, 34]. SOM has been linked with minimal wiring of cortex-like maps [10, 37].

The SOM has been widely used for data visualisation. However the inter-neuron distances, when referred to the data space, have to be crudely or qualitatively marked by colours or grey levels on the trained map. The coordinates of the neurons (the resulting of scaling) are fixed on the lower dimensional (often 2-D) grid and do not resemble the distances (dissimilarities) in the data space.

For metric scaling and data visualisation, a direct and faithful display of data structure and distribution is highly desirable. The ViSOM extends the SOM for distance preservation on the map [54]. For the map to capture the data manifold structure directly, (local) distance quantities must be preserved on the map, along with the topology. The map can be seen as a smooth and graded mesh embedded into the data space, onto which the data points are mapped and the inter-point distances are approximately preserved.

In order to achieve that, the updating force, $[\mathbf{x}(t) - \mathbf{w}_k(t)]$, of the SOM algorithm is decomposed into two elements $[\mathbf{x}(t) - \mathbf{w}_v(t)] + [\mathbf{w}_v(t) - \mathbf{w}_k(t)]$. The first term, $[\mathbf{x}(t) - \mathbf{w}_v(t)]$, represents the updating force from the winner v to the input $\mathbf{x}(t)$, and is the same to the updating force used by the winner v . The second term, $[\mathbf{w}_v(t) - \mathbf{w}_k(t)]$, is a lateral contraction force bringing neighbouring neuron k to the winner. In the ViSOM, this lateral contraction force is regulated in order to help maintain unified inter-neuron distances locally on the map. The update rule is,

$$\Delta \mathbf{w}_k(t) = \alpha(t)\eta(v, k, t)[\mathbf{x}(t) - \mathbf{w}_v(t)] + \beta[\mathbf{w}_v(t) - \mathbf{w}_k(t)] \quad (16)$$

where β is a constraint coefficient -the simplest form being $\beta = \delta_{vk}/(d_{vk}\lambda) - 1$, δ_{vk} is the distance of neuron weights in the input space, d_{vk} is the distance of neuron indexes on the map, and λ is a resolution constant. A further refresh step (using neurons weights are the input) is added to SOM algorithm to ensure smooth expansion of the map in areas where the data is sparse or empty [55, 56].

The ViSOM regularizes the inter-neuron contraction so that local distances between the nodes on the map are analogous to the distances of their weights in the data space. In addition to SOM's objective to minimise the quantisation error, the aim is also to maintain constant inter-neuron distances locally. When the data points are eventually projected onto the trained map, the distance between data points i and j on

the map is proportional to the distance of these two points in the data space, at least locally, subject to the quantisation error (the distance between a data point and its neural representative). That is, $d_{ij} \propto \delta_{ij}$ or $\lambda d_{ij} \approx \delta_{ij}$. This makes data visualisation more direct and quantitatively measurable. The resolution of the map can be enhanced by interpolating a trained (small) map or by incorporating the local linear projection (LLP) method [57]. Instead of projected onto the winning node v (or \mathbf{w}_v), the data point \mathbf{x} is projected to the sub plane spanned by two closest edges. Projected point is therefore,

$$\mathbf{x}' = \mathbf{w}_v + \max_{v'=v\pm 1} \left\{ \frac{(\mathbf{x} - \mathbf{w}_v) \bullet (\mathbf{w}_v - \mathbf{w}_{v'})}{\|\mathbf{w}_v - \mathbf{w}_{v'}\|^2}, 0 \right\} \quad (17)$$

where ‘ \bullet ’ denotes dot-product.

The size or covering range of the neighbourhood function decreases from an initially large value to a final small one. The final neighbourhood, however, should not shrink to contain only the winner, but can be made adaptive. The rigidity or curvature of the map is controlled by the size of the neighbourhood. The larger this size the flatter the final map is in the data space.

Several improvements have since been made to the ViSOM for improved stability and flexibility [53, 11]. A growing ViSOM structure has been proposed recently to effectively extracting highly nonlinear manifold [59].

The similarities between SOMs and MDS in terms of topographic mapping, mostly the qualitative likeness of the mapping results, have been reported before [42]. However limitations of using the SOM for MDS have also been noted [12] – the main one being that SOM does not preserve distance. Many applications combine the SOM and MDS for improved visualisation of the SOM projection results. In [42], it is argued that the SOM is closer to MDS than to principal manifold. In [55], it is shown that the distance preserving ViSOM approximates a discrete principal manifold. One advantage of the ViSOM is that its neighbourhood is usually made adaptive according to data characteristics in various regions, unlike that in other nonlinear PCA or manifold methods, a preset neighbourhood size has to be set empirically. ViSOM has also been shown produces a similar mapping result as to metric MDS. Such a connection has been proven recently [58, 59]. ViSOM is a metric MDS, while order-preserving SOM is a kind of nonmetric MDS.

Other SOM-based or motivated approached for finding the nonlinear manifold includes adaptive subspace SOM (ASSOM) [26], the generative topographic mapping (GTM) [4], self-organizing mixture network (SOMN) [62] and topological product of experts [14]. These methods model the data by a means of a latent space. They belong to the semi-parameterized mixture model, although types and orientations of the local distributions vary from method to method. These approaches also resemble or can produce nonlinear PCA. SOM is used to quantize or segment the input space into local regions (Voronoi tessellation) and local probabilistic models formed in these regions can interpret PCA locally. Other similar neural approaches also include the early elastic net [10] and elastic maps [17].

6 Applications and Discussions

Dimension reduction methods and manifolds have found numerous applications in a number of fields. A large number of articles can be found in the literature. Here we list a sample of typical applications and examples.

Data visualization

Data visualization looks for low dimensional, visual subspaces of the data space in order to observe and display characteristics of the data along these subspaces. Adaptively extracting data's manifold or submanifolds is a principled way to visualize the data set in low dimensional spaces. The approaches described in the previous sections serve as good tools for such applications. For example, SOM and its variants have been widely used for visualizing and organizing sociological data of high dimensional attributes, for instance web documents using WEBSOM) [21] and unstructured text documents using a Windows Explorer alike tree-structure 1-D SOMs [13]. In visualizing and organizing type of data, the ordinal relationship among the data items are important. Therefore topology preserving mapping such as SOMs are useful and effective.

For more numerical visualization, a metric preserving mapping such as metric MDS or nonlinear PCA is essential. ViSOM has been used for visualizing this kind of data, in particular their distribution and structure [55] as it can extract highly nonlinear metric manifolds [59].

Embedding and generative models

Fitting high dimensional data to a low dimensional model can produce an effective generative model of the data. Such models can be used for finding principal, nonlinear latent variables thus underlying models of the data set, as well as filtering noise in the data. Many methods in Sections 2 and 4 are useful tools in this aspect. For instance, Isomap, LLE, GTM and gViSOM [59] have been used to extract highly nonlinear generative, metric manifolds.

Pattern recognition

In many pattern recognition tasks, the first step is to reduce the number of features or dimensionality of the data. For example, both PCA and SOM have been tested on reducing image dimensionality in a face recognition task and nonlinear SOM has been shown to outperform linear PCA [29]. Isomap and LLE have been applied to extract the most important (nonlinear) variations such as pose and lighting in face database decomposition [49, 44].

Feature selection and reduction

Usually feature selection or reduction is performed by supervised learning. However, feature reduction can be carried out in an unsupervised way using a dimensionality reduction method. More recently, semi-supervised methods, which combine supervised and unsupervised procedures, have become an active area of research and has been used for more effective use of both labeled and unlabeled data. For example, a Laplacian score has been proposed to down size and select features [20].

Bioinformatics and neuroinformatics

In bioinformatics, there is a huge demand for analyzing the effect of genes under certain conditions or identifying a subset of genome that contribute to certain biological function or malfunction. The problem is particularly challenging when the number of genes under study is much greater than the number of samples. Microarray data analysis has become increasingly relying on dimensionality reduction techniques and manifold clustering. For instance, in [39], a hybrid dimension reduction method is employed prior to the classification of tumor tissue samples. In [38], a 1-D SOM is utilized to efficiently group yeast *Saccharomyces cerevisiae* cell cycle data, in combination with a novel temporal shape metric.

Neuroinformatics and systems neuroscience also rely intensively on computational techniques for revealing the interaction between neurons as well and between populations or networks. In light of large number of trails and sampling points, decoding experimental data sets becomes particularly challenging. In [60], a topological mapping is applied to the spike trains recorded in rat somatosensory cortex in response to vibrissal stimulations. Combined with the information theoretic framework, it has effectively shown the neuronal response as an energy code [60].

Time series and sequence analysis

Temporal or spatio-temporal manifold is a challenging new direction in this area. Most existing dimension reduction methods regard data as high dimensional, spatial vectors. Such methods may be adopted for temporal signal or sequence when considering time series as vectors of consecutive time points (e.g. using a sliding window over the time series). However such approach is not optimal as spatial vector representation does not take into account the temporal relations of time points. New approaches that naturally consider temporal input are needed. For example, when a temporal metric can be used, then the spatial method can turn into a temporal method [38]. More recently, a self-organizing mixture autoregressive (SOMAR) network has been proposed to cluster and model nonstationary time series [40]. The method regards the entire nonstationary time series as a hybrid of local linear regressive models and it has been shown to produce better forecasting results for financial data over the spatial methods and the GARCH model.

7 Conclusions

In this paper an overview on nonlinear principal manifold for dimensionality reduction and data visualization is presented. The existing methods have been categorized into four groups: nonlinear PCA, MDS-based, principal curve/surface, and SOM-based. While nonlinear PCA approaches are particularly advantageous, especially when they are framed under the kernel method, as the problem is then concerted to a linear eigenvalue problem and a unique solution can be expected, they still need to set a number of kernel or geometrical parameters. They usually are not nonlinear learning methods. Adaptive learning approaches such as SOM-based can gradually extract low dimensional manifolds. SOM-based methods have been shown as either nonmetric or metric MDS. The flexibility and diversity of these adaptive learning approaches make them widely applicable and integrateable with other computational paradigms.

Several typical applications of these methods have been reviewed, together with some challenges ahead. The research and application are set to flourish in many disciplines in this data-rich era.

References

1. Balasubramanian, M., Schwartz, E.L.: The Isomap algorithm and topological stability. *Science* 295, 7a (2002)
2. Banfield, J.D., Raftery, A.E.: Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *J. Amer. Statist. Assoc.* 87, 7–16 (1992)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15, 1373–1396 (2003)
4. Bishop, C.M., Svensén, M., Williams, C.K.I.: GTM: The generative topographic mapping. *Neural Computation* 10, 215–235 (1998)
5. Borg, I., Groenen, P.J.F.: *Modern Multidimensional Scaling: Theory and Applications*, 2nd edn. Springer, Heidelberg (2005)
6. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *PNAS* 103, 1168–1172 (2006)
7. Chang, K.-Y., Ghosh, J.: A unified model for probabilistic principal surfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence* 23, 22–41 (2001)
8. Cox, T.F., Cox, M.A.A.: *Multidimensional Scaling*. Chapman & Hall, London (1994)
9. De Ridder, D., Duin, R.P.W.: Sammon mapping using neural networks: A comparison. *Pattern Recognition Letters* 18, 1307–1316 (1997)
10. Durbin, R., Mitchison, G.: A dimension reduction framework for understanding cortical maps. *Nature* 343, 644–647 (1990)
11. Estévez, P.A., Figueroa, C.J.: Online data visualization using the neural gas network. *Neural Networks* 19, 923–934 (2006)
12. Flexer, A.: Limitations of self-organizing maps for vector quantization and multidimensional scaling. *Advances in Neural Information Processing Systems* 10, 445–451 (1997)
13. Freeman, R.T., Yin, H.: Adaptive topological tree structure for document organisation and visualisation. *Neural Networks* 17, 1255–1271 (2004)
14. Fyfe, C.: Two topographic maps for data visualisation. *Data Mining and Knowledge Discovery* 14, 207–224 (2007)
15. Goodhill, G.J., Sejnowski, T.: A unifying objective function for topographic mappings. *Neural Computation* 9, 1291–1303 (1997)
16. Gorban, A.N., Kégl, B., Wunsch, D.C., Zinovyev, A.: *Principal Manifolds for Data Visualization and Dimension Reduction*. Springer, Heidelberg (2008)
17. Gorban, A., Zinovyev, A.: Method of elastic maps and its applications in data visualization and data modeling. *Int. J. Computing Anticipatory Systems* 12, 353–369 (2001)
18. Ham, J., Lee, D.D., Mika, S., Schölkopf, B.: A kernel view of the dimensionality reduction of manifolds. In: *Proc. 21st Int. Conf. on Machine Learning*, p. 47 (2004)
19. Hastie, T., Stuetzle, W.: Principal curves. *J. Amer. Statist. Assoc.* 84, 502–516 (1989)
20. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. *Advances in Neural Information Processing Systems* 18 (2005)
21. Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: WEBSOM – self-organizing maps of document collections. In: *Proc. Workshop on Self-Organizing Maps*, pp. 310–315 (1997)
22. Karhunen, J., Joutsensalo, J.: Generalisation of principal component analysis, optimisation problems, and neural networks. *Neural Networks* 8, 549–562 (1995)

23. Kégl, B., Krzyzak, A., Linder, T., Zeger, K.: A polygonal line algorithm for constructing principal curves. *Advances in Neural Information Processing Systems* 11, 501–507 (1998)
24. Kohonen, T.: Self-organised formation of topologically correct feature map. *Biological Cybernetics* 43, 56–69 (1982)
25. Kohonen, T.: *Self-organization and associative memory*. Springer, Heidelberg (1984)
26. Kohonen, T.: The adaptive-subspace SOM (ASSOM) and its use for the implementation of invariant feature detection. In: *Proc. Int. Conf. on Artificial Neural Networks*, pp. 3–10 (1995)
27. Kohonen, T.: *Self-Organising Maps*, 2nd edn. Springer, Heidelberg (1997)
28. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. *AICHE (American Institute of Chemical Engineers) Journal* 37, 233–243 (1991)
29. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: A convolutional neuralnetwork approach. *IEEE Trans. Neural Networks* 8, 98–113 (1997)
30. LeBlanc, M., Tibshirani, R.J.: Adaptive principal surfaces. *J. Amer. Statist. Assoc.* 89, 53–64 (1994)
31. Lee, R.C.T., Slagle, J.R., Blum, H.: A triangulation method for the sequential mapping of points from n-space to two-space. *IEEE Trans. Computers* 27, 288–292 (1977)
32. Lowe, D., Tipping, M.E.: Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications* 4, 83–95 (1996)
33. Luttrell, S.P.: Derivation of a class of training algorithms. *IEEE Trans. Neural Networks* 1, 229–232 (1990)
34. Luttrell, S.P.: A Bayesian analysis of self-organising maps. *Neural Computation* 6, 767–794 (1994)
35. Malthouse, E.C.: Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Trans. Neural Networks* 9, 165–173 (1998)
36. Mao, J., Jain, A.K.: Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. Neural Networks* 6, 296–317 (1995)
37. Mitchison, G.: A type of duality between self-organizing maps and minimal wiring. *Neural Computation* 7, 25–35 (1995)
38. Möller-Levet, C.S., Yin, H.: Modeling and analysis of gene expression time-series based on co-expression. *Int. J. Neural Systems* 15, 311–322 (2005)
39. Nguyen, D.V., Rockeb, D.M.: On partial least squares dimension reduction for microarray-based classification: a simulation study. *Computational Statistics & Data Analysis* 46, 407–425 (2004)
40. Ni, H., Yin, H.: Time-series prediction using self-organising mixture autoregressive network. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) *IDEAL 2007. LNCS*, vol. 4881, pp. 1000–1009. Springer, Heidelberg (2007)
41. Oja, E.: Neural networks, principal components, and subspaces. *Int. J. Neural Systems* 1, 61–68 (1989)
42. Ripley, B.D.: *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge (1996)
43. Ritter, H., Martinetz, T., Schulten, K.: *Neural Computation and Self-organising Maps: An Introduction*. Addison-Wesley Publishing Company, Reading (1992)
44. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 2323–2326 (2000)
45. Sammon, J.W.: A nonlinear mapping for data structure analysis. *IEEE Trans. Computer* 18, 401–409 (1969)
46. Sanger, T.D.: Optimal unsupervised learning in a single-layer linear feedforward network. *Neural Networks* 2, 459–473 (1991)

47. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10, 1299–1319 (1998)
48. Smola, A.J., Williamson, R.C., Mika, S., Schölkopf, B.: Regularized principal manifolds. In: Fischer, P., Simon, H.U. (eds.) *EuroCOLT 1999. LNCS (LNAI)*, vol. 1572, pp. 214–229. Springer, Heidelberg (1999)
49. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)
50. Tibshirani, R.: Principal curves revisited. *Statistics and Computation* 2, 183–190 (1992)
51. Weiss, Y.: Segmentation using eigenvectors: a unified view. In: *Proc. IEEE Int. Conf. on Computer Vision*, pp. 975–982 (1999)
52. Willshaw, D.J., von der Malsburg, C.: How patterned neural connections can be set up by self-organization. *Proc. Royal Society of London Series B* 194, 431–445 (1976)
53. Wu, S., Chow, T.W.S.: PRSOM: A new visualization method by hybridizing multidimensional scaling and self-organizing map. *IEEE Trans. Neural Networks* 16, 1362–1380 (2005)
54. Yin, H.: ViSOM-A novel method for multivariate data projection and structure visualisation. *IEEE Trans. Neural Networks* 13, 237–243 (2002)
55. Yin, H.: Data visualisation and manifold mapping using the ViSOM. *Neural Networks* 15, 1005–1016 (2002)
56. Yin, H.: Nonlinear multidimensional data projection and visualisation. In: Liu, J., Cheung, Y.-m., Yin, H. (eds.) *IDEAL 2003. LNCS*, vol. 2690, pp. 377–388. Springer, Heidelberg (2003)
57. Yin, H.: Resolution enhancement for the ViSOM. In: *Proc. Workshop on Self-Organizing Maps, Kitakyushu, Japan, Kyushu Institute of Technology*, pp. 208–212 (2003)
58. Yin, H.: Connection between self-organising maps and metric multidimensional scaling. In: *Proc. Int. Joint Conf. on Neural Networks, Orlando*, pp. 1025–1030. IEEE Press, Los Alamitos (2007)
59. Yin, H.: On multidimensional scaling and the embedding of self-organizing maps. *Neural Networks* 21, 160–169 (2008)
60. Yin, H., Panzeri, P., Mehboob, Z., Diamond, M.: Decoding population neuronal responses by topological clustering. In: *Proc. Int. Conf. on Artificial Neural Networks. LNCS*, vol. 5164, pp. 547–557. Springer, Heidelberg (2008)
61. Yin, H., Allinson, N.M.: On the distribution and convergence of the feature space in self-organising maps. *Neural Computation* 7, 1178–1187 (1995)
62. Yin, H., Allinson, N.M.: Self-organising mixture networks for probability density estimation. *IEEE Trans. Neural Networks* 12, 405–411 (2001)

Multi-agent ERA Model Based on Belief Interaction Solves Wireless Sensor Networks Routing Problem

Yanbin Liu, Chunguang Zhou, Kangping Wang, Dan Li, and Dongwei Guo

College of computer science and technology, Jilin University, Changchun 130012,
China
skywind.guo@gmail.com

Abstract. ERA is the acronym of environment, reactive rules and agent. The model is a new effective multi-agent cooperation framework, which has been successfully applied in a wide range of fields. This paper presents the concept of belief into multi agent system. Each agent stands for a feasible routing, it moves according to its own belief matrix in the sensor networks environment. Agent has the ability to evaluate and adjust its belief matrix according to the historical routing. Agents will interact their believes to each other when they meet at the same sensor node. This paper employs a large number of data and other classical models to evaluate the algorithm, the experimental results show that this model prolongs the lifetime of wireless sensor network by reducing energy dissipation, which validates the algorithm's feasibility and availability.

1 Introduction

As a sub-field of distributed artificial intelligence, agent theory has developed fast since it was first proposed in 1970s (see [1]). Multi-Agent System (MAS) is a multi-knowledge formalization model, which has become a hotspot of investigation in recent years. Generally speaking, an autonomous agent is a system situated within and a part of an environment that senses environment and acts on it, over time, in pursuit of its own agenda and so as to affect what it senses in the future (see [2]). Agent has the knowledge, target and abilities. MAS has been widely applied in many areas of artificial intelligence, for example, control process (see [3]), mobile robot (see [4]), air-traffic management (see [5]) and intelligence information recycle (see [6])etc.

ERA model is a new multi-agent collaboration architecture of MAS, proposed by Jiming Liu (see [7]). ERA contains environment, reactive rules and agent, in which agents collaborate to achieve a global solution. It has successfully solved lots of problems, for example, Constraint Satisfaction Problem (CSP) (see [7])etc.

This paper is organized as follows. Section 2 introduces the routing problem of wireless sensor networks (WSNs) and presents a model of WSNs. We then define multi-agent ERA model in Section 3, which contains environment, agent and rules. The algorithm description in detail is proposed in Section 4. Before the conclusion, we give the experimental results of this paper.

2 Problem Descriptions and Model

2.1 Routing Problem in WSNs

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered in short distances. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of sensor networks based on collaborative effort of a large number of nodes. Sensor networks represent a significant improvement over traditional sensors (see [8]).

Wireless sensor networks, which is utilized in a broad foreground, is able to instantly monitor and collect various information of different objects in the networks distributed region (see [9]). Figure 1 shows a classical architecture of WSNs (see [10]), The main techniques are to design an effective routing algorithm, which will advance the communication connectivity. The objective is to decrease the energy dissipation and prolong the lifetime of sensor networks. A large number of papers are related to energy issues in WSNs(see [11], [12] and [13]).

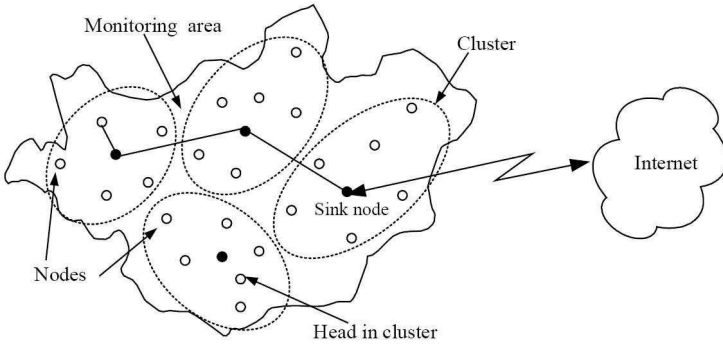


Fig. 1. Architecture of sensor networks

2.2 Model of Wireless Sensor Networks in This Paper

A wireless sensor network is constituted by a large scale sensor nodes, which are divided into clusters. The sensor nodes do not send data directly, but, there is a sink node with more energy than others to collect data from others nodes in the same cluster. Because of the signal limitation, a sensor node can connect with the node which is within a distance shorter than the launch radius. This paper focuses on the data collection in a cluster.

S is the set of sensor nodes and E is the connectivity set of one sensor. $s \in S$ is a sensor node of cluster, $(u, v) \in E$ represents the energy dissipation of transmitting 1k bit data between sensor node u and v . This paper uses the following wireless communication model (see [14]): receiving a 1k bit data packet will dissipate 1k elec energy of the sensor. The unit energy dissipation of sender amplifier is distance-related, which is the value of (u, v) represented by $C(u, v)$. The routing starts from the sink

sensor node, and ends in the same node after finishing data collection. $P = \{s_0, \dots, s_i, (s_i, s_j), s_j, \dots, s_0\}$ stands for a routing path, s_0 is the sink sensor. The energy dissipation formula of the routing is defined as Eq.(1):

$$EC(P) = \sum_{s \in P} D + \sum_{(u,v) \in P} C(u,v)D \quad (1)$$

Here, D is the data size of transmittal data packet, it is a dynamic value. This paper will find a routing for data collection, based on this routing, the core sensor will get all data from other sensors with dissipating the minimal energy.

3 Multi-agent ERA Model

The ERA model is a kind of distributed MAS, similar to Swarm (see [15]) architecture, it has the self-organization ability. Agent changes according to predefined rules of conduct enable the system to achieve a global solution. This paper presents a MAS model based on belief, in which each agent manages a feasible routing.

3.1 Environment

The environment is defined as the region of wireless sensor networks. It is a two-dimensional space, sensors can be ascertained in this space with x and y coordinates.

Sensor is located in the environment randomly. Each pair of sensors has their energy dissipation for communication, but, sensor can only connect to other sensors within its launch radius, which is an initialize property of sensor.

3.2 Agent

Agents exist in the environment, each one stands for a total routing. It has intelligent behavior, and a target is given to the individual differently based on different concrete problems. Agent will decide its own action according to its belief matrix in the process of iteration to attain the purpose of tending target (see [16] and [17]). In this paper, agent's purpose is defined as the minimum of energy dissipation of its routing.

Belief is an abstract concept of intelligence, which will affect the movement of agents. Each agent has a two-dimensional belief matrix, which is created according to sensor connection status in environment, and matrix $[s_1][s_2]$ value is set by a random method based on energy dissipation of communication between sensor node s_1 and s_2 .

3.3 Reactive Rules

Selection rule. This effects the action of agent in choosing a sensor to visit next. The flow in detail is as follows:

- (1) Confirm the sensor node which agent is attaching on currently, assumed it to be sensor node s .
- (2) Get the connective sensors of sensor s and ordering these sensors descending by their values in belief matrix of the agent.
- (3) Get the former m sensors of (2), and choose a sensor to go ahead randomly

Adjustment rule. Agent will adjust its belief matrix values relative to sensors in the previous routing according to the changing of energy dissipation, which is belief adjustment rule.

Let matrix b_i stand for belief strength of agent i . α , β and γ are parameters.

Here, α and β are real numbers, γ is a percentage, which measures the change degree of the belief. b_i is adjusted as follows :

- If energy dissipation reduces and the change degree is stronger than γ , the adjustment uses formula $b_i[x, y] = b_i[x, y] + \alpha$, otherwise $b_i[x, y] = b_i[x, y] + \beta$.
- If energy dissipation increases and the change degree is stronger than γ , the adjustment uses formula $b_i[x, y] = b_i[x, y] - \alpha$, otherwise $b_i[x, y] = b_i[x, y] - \beta$.

Where, x and y directly connect in the best routing of agent i .

Interaction rule. The rule of interaction defines the interactive method of agents meeting at the same sensor node. If routing energy dissipation of agent j is smaller than i , agent i will adjust its belief matrix according to b_j as the following equation:

$$b_i[x, y] = \lambda b_j[x, y] + (1 - \lambda) b_i[x, y] \quad (2)$$

4 Algorithm Description

This paper constructs algorithm based the proposed MAS model, which creates the environment and sensors as required, and then creates agents and initializes their random belief matrices. In the process of system run, agents change their positions and beliefs gradually to get routing, the system records the optimal solution.

The detailed algorithm is shown as follows:

-
- 1: Initialize swarm. Create sensor nodes' positions randomly. Set launch radius for each node. Initialize the sensor communication energy dissipation matrix M_s . { M_s is exclusive in the system. }
 - 2: Create and locate agents. Initialize agent's belief matrix M_a itself, which is according to M_s . { The smaller the energy dissipation, the stronger the belief. }
 - 3: Set $m = 3$ in *selection rules*.
 - 4: **while** NOT arriving at iteration times AND NOT convergence principle **do**
 - 5: **for all** Agent i **do**
 - 6: Choose the target node from the m sensors that selected by *selection rules*, and it is required that the target node has not been visited by agent i
 - 7: Agent moves to the target node and adds it to session routing.
 - 8: Get the lowest energy dissipation of other agents at the same node, and compares itself's with it.
 - 9: **if** Agent i 's energy dissipation of STEP 8 is more **then**
 - 10: Adjust the belief matrix M_a according to the Eq.(2)
 - 11: **else**
 - 12: Keep M_a steady
 - 13: **end if** { Where $\lambda = 0.7$. }
 - 14: **if** The movement step of agent is the integral multiple of sensor's quantity **then**
 - 15: **if** The session routing is better **then**

```

16:         Increase its belief matrix values
17:     else
18:         Decrease its belief matrix values
19:     end if
20:     Agent calculates and updates its best routing, then, clears its session
    routing.
21:     System gets the global best routing.
22: end if
23: end for
24: end while

```

In step 14, the movement step is the integral multiple of sensor's quantity ensures that the agent exactly constructs a complete path. The individual gets a new feasible solution.

5 Experimental Results

This paper solves data collection problem within a cluster of sensor networks. The experiments set the number of sensor nodes as 100, 200, 300, 400, and 500. The environment of wireless sensor networks is a 200m*200m square. Let the transmitted data size of one sensor be 1k bit, launch radiuses are 20m and 30m. The experimental results are the average value of 50 runs. Figure 2 and figure 3 show the trend of optimal solution in different data sets. We can see from these figures that the energy dissipation decreases gradually, the curve is smoother because of the bigger launch radius in figure 3.

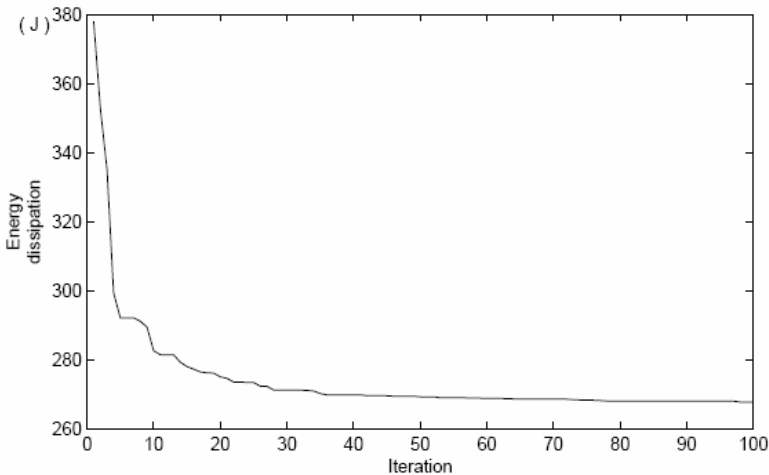


Fig. 2. The trend of energy dissipation (*sensor = 400, R = 20*)

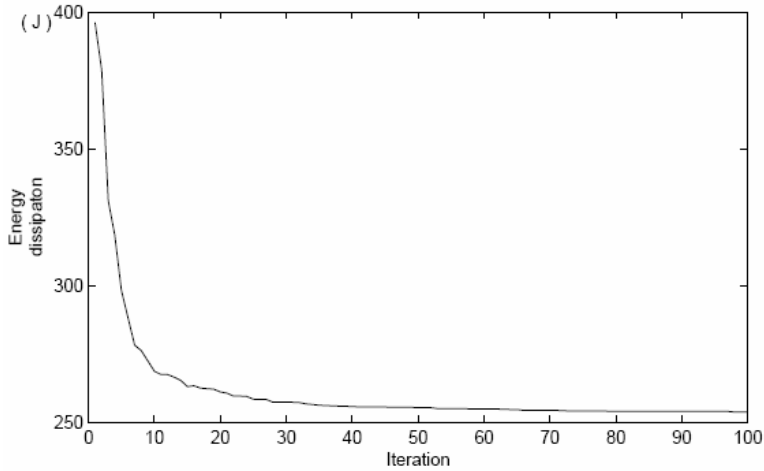


Fig. 3. The trend of energy dissipation ($sensor = 400, R = 30$)

The figures are only two cases of the experiments, table 1 is the summary of experiment results. CS (see [18]) and GK (see [19]) algorithm solved wireless sensor networks routing problem using mathematic method, their results are outstanding in algorithms solving this problem. According to the data in table 1, the results in this paper are better than CS and GK algorithm in small-scale and medium-scale networks, and a slightly inferior in large-scale networks. In practical, there are a lot of applications of medium-scale wireless sensor networks, so, the algorithm in this paper can be more applied in actual. The energy dissipation is smaller, because of the improvement of the connectivity of sensor networks, which is brought by the bigger launch radius.

Table 1. The relationship between energy dissipation and the number of sensor nodes (J)

Sensor nodes	CS		GK		This paper	
	$R=20m$	$R=30m$	$R=20m$	$R=30m$	$R=20m$	$R=30m$
100	101.02	98.00	102.64	99.05	99.04	95.55
200	132.01	108.06	132.00	108.48	126.00	104.67
300	187.32	165.96	187.46	167.88	186.89	160.33
400	267.24	251.24	269.15	254.56	268.09	254.72
500	385.80	357.96	386.63	363.37	391.60	370.82

Another evaluation of routing algorithm is the lifetime of wireless sensor networks, which can be calculated by energy dissipation indirectly.

The life ratio of our work and other algorithms is displayed in table 2. The network's life of this paper has a greater improvement under most circumstances.

Table 2. The wireless sensor networks life ratio of different routing algorithms

Sensor nodes	This paper/CS		This paper/GK	
	$R=20m$	$R=30m$	$R=20m$	$R=30m$
100	1.020	1.026	1.036	1.037
200	1.047	1.032	1.047	1.036
300	1.002	1.035	1.003	1.047
400	0.996	0.986	1.004	0.999
500	0.985	0.965	0.987	0.980

6 Conclusion and Further Work

This paper uses multi-agent system based on belief to solve wireless sensor networks routing problem, our model represents the concept of belief to agent and constructs environment and rules. The experiments use different scales sensor networks and launch radius, the analysis of the optimal solution and the comparisons with other classical algorithms prove the validity of this model.

The further work will update the mechanism and rules of this multi-agent model to improve the issues of large-scale wireless sensor networks problem.

Acknowledgment

This research works are supported by European Commission under grant No.TH/Asia Link/010 (111084), National Natural Science Foundation of China under Grant No.60433020, 60673099, 60773095 and 863 project 2007AA04Z114.

References

1. Weiss, G.: Multi-agent systems: A modern approach to distributed artificial intelligence. MIT Press, Cambridge (1999)
2. Petrie, C.J.: Agent-based engineering: the web and intelligence. IEEE Intelligent Systems and Their Applications, 24–29 (1996)
3. Jennings, N.R., Corera, J.M., Laresgoiti: Developing industrial multi-agent systems. In: Proceedings of the First International Conference on multi-agent Systems, ICMAS, pp. 423–430
4. Neves, M.C., Oliveira, E.: A Control Architecture for an Autonomous Mobile Robot. In: Agents 1997. ACM, New York (1997)
5. Ljungberg, M., Lucas, A.: The OASIS air-traffic management system. In: Proceedings of the Second Pacific Rim International conference on Artificial Intelligence, PRI-CAI 1992, Seoul, Korea (1992)
6. Julian, V., Carrascosa, C., Soler, J.: Una Arquitectura de Sistema Multi-Agente para la Recuperación y Presentación de Información. In: IV Congreso ISKO-España, EOCONSID 1999, pp. 291–296 (1999)

7. Liu, J., Jing, H., Tang, Y.Y.: Multi-agent oriented constraint satisfaction. *Artificial Intelligence*, S0004-3702(01)00174-6/FLA, 1–41 (2004)
8. Akyildiz, I.F., et al.: Wireless sensor networks: a survey. *Computer Networks* 38, 393–422 (2002)
9. Wei, Z.X., Bin, L., Ping, Q.B.: Research on Routing Algorithm for wireless Sensor Network. *Chinese journal of sensors and actuators* 19(2), 463–467 (1999)
10. Feng-Yuan, R., Hai-Ning, H., Chuang, L.: Wireless Sensor Networks. *Journal of Software* 7, 1282–1292 (2003)
11. Mainwaring, A.: Wireless Sensor Networks for Habitat Monitoring. In: *WSNA 2002*, Atlanta, Georgia, USA (2002)
12. Park, J., Sahni, S.: An Online Heuristic for Maximum Lifetime Routing in Wireless Sensor Networks. *IEEE Transactions On Computers* 55(8), 1048–1057 (2006)
13. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks*, 293–315 (2003)
14. Tao, C.Y., Chen, H., Jun, W., Quasn, W.W.: An Energy Efficient Routing Algorithm for Mobile Agents in Wireless Sensor Networks. *Journal of shanghai jiao tong university* 40(3), 520–523 (2006)
15. Swarm Development Group, Swarm simulation system, <http://www.swarm.org/index.html>
16. Rao, A.S., Georgeff, M.P.: BDI Agents, From Theory to Practice. In: *The Proceedings of the First International Conference on Multi-Agent-System (ICMAS)*, SanFrancisco (1995)
17. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: *Proceedings of the 2rd International Conference on Principles of Knowledge Representation and Reasoning*, pp. 473–484. Morgan Kaufmann Publishers, San Mateo (1991)
18. Current, J.T., Schilling, D.A.: The covering salesman problem. *Transportation Science* 24(3), 208–213 (1989)
19. Guha, S., Khuller, S.: Approximation algorithms for connected dominating sets. *Algorithmica* 20(4), 374–387 (1998)

Multi-agent System for Management and Monitoring of Routes Surveillance

Sara Rodríguez and Javier Bajo

Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, España
{srg, jbajope}@usal.es

Abstract. This paper presents a multi-agent system for security control on industrial environments. The system uses a set of wireless technologies and software agents which integrate reasoning and planning mechanisms. It has the ability to obtain automatic and real-time information about the context to schedule the security guards activities. The combination of technologies enables users to interact with the system in a simple, natural and intuitive way.

Keywords: Industrial Security, Agents, Surveillance Routes Calculation, Monitoring, Radio-Frequency Identification.

1 Introduction

In recent years there has been an expansion in the industrial sector, especially in developed countries. In such an important and growing sector, it is necessary to establish security policies to manage risks and control hazardous events, providing better working conditions and an increase in productivity.

Recent studies [4] have revealed that at least 3% of the working shifts time is spent because of the lack of time control systems that supervise the real working time. Implementation of time control systems have a good influence in productivity, since the workers optimize their potential and enhance the process where they collaborate.

Multi-agent systems and intelligent mobile devices architectures are suitable to handle complex and highly dynamic problems in execution time. Agents and multi-agent systems are successfully implemented in areas such as e-commerce, medicine, oceanography, robotics, etc. [2][3]. They have been recently explored as supervision systems, with the flexibility to be implemented in a wide diversity of scenarios, including industrial sector. The current application of multi-agent systems in real-time environments is an area of increasing interest. In general, the multi-agent system represents an appropriate approach for solving inherently distributed problems, whereby clearly different and independent processes can be distinguished. The use of wireless technologies, such as GPRS (General Packet Radio Service), UMTS (Universal Mobile Telecommunications System), RFID (Radio-frequency identification), Bluetooth, etc., make possible to find better ways to provide mobile services and also give the agents the ability to communicate using portable devices (e.g. PDA's and cellular phones) [10]. Nowadays, there is a great growth in the development of

agents-based architectures, evolved in part because of the advances on intelligent environments and computational networks [10].

This paper presents the application of a novel hybrid artificial intelligence (AI) system to manage and monitor surveillance routes for security guards on industrial environments. The system uses a set of wireless technologies: Wi-Fi, GPRS and RFID. These technologies, increase the mobility, flexibility and efficiency of users, allowing them to access resources (programs, equipment, services, etc.) remotely, no matter their physical location.

The rest of the article is structured as follows: in Section 2 a case study is defined, describing the development of a multi-agent system designed to solve some of the problems that affect the industrial sector. Sections 3 and 4 present the main characteristics of the system, describing its architecture and the surveillance routes planning mechanism; and, lastly, the evaluation is presented and the results obtained are analysed.

2 Multi-agent System for Industrial Security

A multi-agent system has been designed to allow scheduling and distribution of surveillance routes among available security guards in order to provide better control over the activities performed by the staff responsible for overseeing the industrial environments. The routes assigned are automatically and real-time monitored to ensure the accomplishment of the security guards working shifts. The system interacts with users through a set of mobile devices (PDA's) and wireless communication technologies (Wi-Fi, GPRS and RFID). These technologies and devices work in a distributed way, providing the users a flexible and easy access to the system services.

Depending on the security guards available, the working shifts and the distance to be covered in the facilities, the agents in the system calculate the surveillance routes. A supervisor (person) can set the possible routes, defining the areas that must be supervised, which can be modified according the scenario or changes in the environment. The system has the ability to re-plan the routes automatically considering the security guards available. It is also possible to track the workers activities (routes completion) over the Internet.

Radiofrequency Identification (RFID) is a key technology in this development. RFID is an automated data-capture technology, relying on storing and remotely retrieving data. It is most frequently used in industrial/manufacturing, transportation, distribution, and warehousing industries, but there are other growth sectors including health care [11]. As can be seen in Figure 1, the RFID configuration for the system presented within this paper consists of a mesh of tags distributed all over the building. Each tag, named "control point" is related to an area which must be covered by the security guards. Each security guard carries on a PDA with a RFID reader to register the completion of each control point. The information is sent via wireless technology to a central computer where it is processed.

The case study has been successfully implemented in a real environment. Sections 3 and 4 present the main characteristics of the system, describing its architecture as well as the surveillance routes planning mechanism.

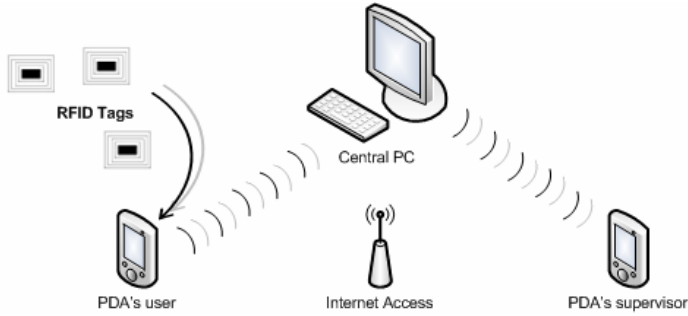


Fig. 1. Basic Monitoring Schema

3 Multi-agent Architecture

The core of the hybrid system presented in this work is a multi-agent architecture specifically designed to facilitate the development of ubiquitous environments. The basis of the multi-agents systems is the cooperation of multiple autonomous agents [2]. Multi-agent architectures comprise plenty errors recovery, with the ability to initialize or end separate agents without the need to restart the entire system. The developed system presented on this paper has these features, so it is possible to initialize multiple services on demand. The agents' behaviour is affected when it is necessary to schedule a new surveillance route. Besides, the system store continuously its status information, so it can be restarted and recover the last backup information. The analysis and modelling of the system has been done using a combination of the Gaia methodology [12] and the AUMML language [1]. Once defined the system structure, shown on Figure 2, it is possible to appreciate that the system is composed of five different kinds of agents:

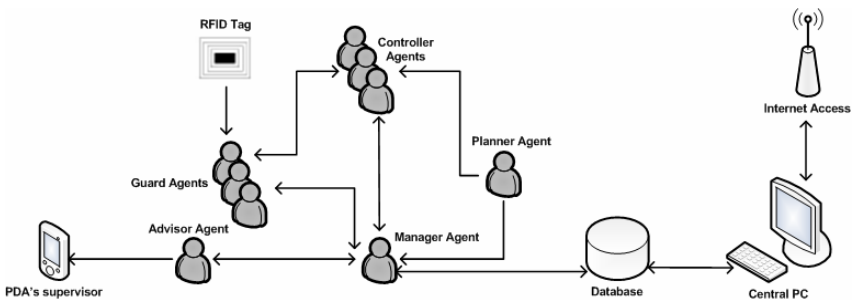


Fig. 2. Multi-agent architecture describing the system structure

- *Guard Agent* manages the portable RFID readers to get the RFID tags information on every control point. Communicates with Controller Agents to check the accomplishment of the assigned surveillance routes, to obtain new routes, and also to send the RFID tags information via Wi-Fi.

- *Manager Agent* controls all the rest of agents in the system. Manages the connection and disconnection of Guard Agents to determine the available security guards available. The information is sent to the Planner Agent to generate new surveillance routes. Manager Agent also receives incidences (omitted control points, route changes, new security guard connected/disconnected, security guards notifications, etc.) from the Controller Agents and Guard Agents and, depending its priority, informs the Advisor Agent. Manager Agent stores all the system information (incidences, time, data, control points, route status, etc.) into a database. Information can be accessed via Internet.
- *Planner Agent* generates automatically the surveillance routes. The routes are sent to the Manager Agent to distribute them among the security guards.
- *Controller Agent* monitors the security guards activities by means of the control points checked. Once a surveillance route is generated by the Planner Agent, the average time to reach each control point is calculated. The Controller Agent also handles the associated route incidences and sends them to the Manager Agent.
- *Advisor Agent* administers the communication with the supervisors (person). Receive from the Manager Agent the incidences, and decide if are sent to the supervisor. Incidences can be sent via Wi-Fi, SMS or GPRS. If a communication problem is detected, the incidence is sent back to the Manager Agent and stored.

4 Surveillance Routes Planning Mechanism

The system, in particular the Planner Agent, makes a route planning, delivering the control points between the security guards. For surveillance routes calculation, the system takes into account the time and the minimum distance to be covered. So it is necessary a proper control points grouping and order on each group. The planning mechanism uses Kohonen SOM (Self Organizing Maps) neural networks with the k-means learning algorithm [8] to calculate the optimal routes and assign them to the available security guards. Once distributed control points, it sends distribution of the points each agent controller, which calculates the route to be followed by every security guard. Neural networks allow the calculus of variable size data collections, and reduce the time and distances to be covered. In addition, the control points can be changed on each calculation, so the surveillance routes are dynamic, avoiding repetitive patterns.

The mechanism starts spreading the control points among the available security guards. Then, the optimal route for each one is calculated using a modified SOM neural network. The modification is done through a FYDPS neural network, changing the neighbourhood function defined in the learning stage of the Kohonen network. The new network has two layers: IN and OUT. The IN layer has two neurons, corresponding the physical control points coordinates. The OUT layer has the same number of control points on each route [6][9].

Be $x_i \equiv (x_{i1}, x_{i2})$ $i = 1, \dots, N$ the i control point coordinates and $n_i \equiv (n_{i1}, n_{i2})$ $i = 1, \dots, N$ the i neuron coordinates on \mathfrak{R}^2 , being N the number of control points in the route. So, there are two neurons for the IN layer and N neurons for the OUT layer. The weight actualization formula is defined by the following equation:

$$w_{ki}(t+1) = w_{ki}(t) + \eta(t)g(k, h, t)(x_i(t) - w_{ki}(t)) \tag{1}$$

Be w_{ki} the weight that connect the IN layer i neuron with the OUT layer k neuron. t represents the interaction; $\eta(t)$ the learning rate; and finally, $g(k, h, t)$ the neighbourhood function, which depends on three parameters: the winner neuron, the actual neuron, and the interaction.

A decreasing neighbourhood function is considered with the number of interactions and the winner neuron distance.

$$g(k, h, t) = \text{Exp} \left[\left(-\frac{|k-h|}{N/2} \right)^{\text{Max}_{i,j} \{f_{ij}\} - \sqrt{(n_{k1} - n_{h1})^2 + (n_{k2} - n_{h2})^2}} - \lambda \frac{|k-h|t}{\beta N} \right] \tag{2}$$

λ and β are determined empirically. The value of λ is set to 1 by default, and the values of β are set between 5 y 50. t is the current interaction. Its value is obtained by means of βN . $\text{Exp}[x] = e^x$, where N is the number of control points. f_{ij} is the distance between two points i and j . Finally, $\text{Max}\{f_{ij}\}$ represents the maximum distance that joins those two points. Each parameter of the neighbourhood function can be 0 or 1. The neighbourhood function radius value must be close to 0 to update just the winner neuron.

To train the neural network, the control points groups are passed to the IN layer, so the neurons weights are similar to the control points coordinated. When all the process concludes, there is only one neuron associated to each control point. To determine the optimal route, the i neuron is associated with the $i+1$ neuron, from $i=1, 2, \dots, N$, covering all the neurons vector. A last interval is added to complete the route, associating the N neuron with the i neuron. The total distance is calculated adding the distances between two points. The learning rate depends on the number of interactions, as can be seen on the following equation:

$$\eta(t) = \text{Exp} \left[-\sqrt[4]{\frac{t}{\beta N}} \right] \tag{3}$$

The neurons activation function is the identity. When the learning stage ends, the winner neuron for each point is determined, so each point has only one neuron associated. The optimal route is then calculated following the weights vector. This vector is actually a ring, where the n_1 neuron is the next n_N neuron. Initially considering a high neighbourhood radius, the weights modifications affect the nearest neurons. Reducing the neighbourhood radius, the number of neurons affected decrease, until just the winner neuron is affected. The learning stage is stopped when the distance between two points cannot be optimized any more.

The initial number of interactions is $T_1 = \beta N$ in the first stage. When $t = \beta N$, the weights of the possible couple of neurons are changed from the neurons ring obtained. If the distance is optimized, the number of interactions is reduced to continue the learning. In the Z phase, the total number of interactions is:

$$T_Z = T_{Z-1} - \frac{T_{Z-1}}{Z} \quad (4)$$

The objective of these phases is to avoid the crossings. Once all interactions are concluded, the distance obtained is analyzed to determine if it is the optimal distance. So, the recoil in the number of interactions is reduced each time, obtaining a maximum number of interactions, although the value is variable. Figure 3 shows the routes planned by the Controller Agent for one and two security guards, with the grouping of checkpoints done by the Planner Agent.

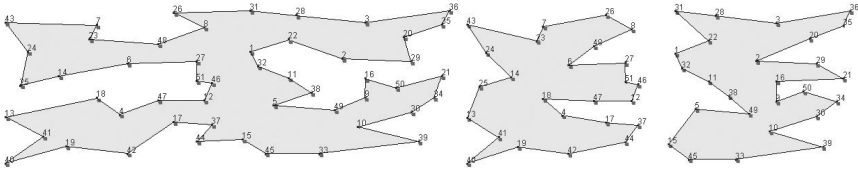


Fig. 3. Routes for one (left) and two (right) security guards.(175, 79 and 76 minutes)

5 Results and Conclusions

The hybrid system presented in this paper has been implemented and tested over real and controlled scenarios. The main scenario used for tests has been an environment where was introduced a variable number of control points. In addition to vary the control points, the number of guards switches between the different tests. They were conducted with a maximum of 4 people with the role of security guard.

Each control point contained an RFID tag that would be activated by the PDA's security guard. Information obtained at each control point was sent a central computer via Wi-Fi. The aim of the tests was to check viability from the combination of technologies (wireless technologies such as RFID, software agents and planning mechanism) in the chosen environment, in a scenario reduced and controlled.

The main input data used by the system were: (i) the number of control points; (ii) the distance between the control points; (iii) the average time to go from one control point to another, (iv) the number of guards available to calculate routes.

The results obtained have shown that it is possible to find out the necessary number of security guards depending on the surveillance routes calculated by the system. For example, Figure 3 (left) shows that the calculated route for a single security guard is relatively large for an 8-hour workday and it can visit once. By adding a new security guard, the new routes, as shown in Figure 3 (right), can be completed over 2 and a half times in 8 hours.

To evaluate the system efficiency, a comparison after and before the prototype implementation was done, defining multiple control points sets and just one security guard. The results of times and distances calculated by the users and the system are shown on Table 1.

Table 1. Time and distance calculated for security guards and multiple control points sets

Security guards	Control Points	Users		System	
		Time (min)	Distance (m)	Time (min)	Distance (m)
1	15	39	1285	28	944
1	20	64	2124	44	1451
1	25	76	2535	53	1761
1	30	79	2617	63	2088
1	35	96	3191	77	2551
3	100	357	11900	253	8415

The system provides optimized calculations, so the time and distance are reduced. A complete working day shift can be fixed according the system results, for example, if the route calculated is too long or the time exceeds eight working hours, a new guard must be incorporated.

Extending these results, Figure 4 (left) shows the average number of estimated security guards needed to cover an entire area, which consisted on a mesh from 20 to 100 control points, with an increment of 5 control points. The results are clear, for example, for 80 control points, the users estimated 4 security guards, but the system recommended only 3.

As shown on Figure 4 (right), the differences are bigger when there are 3 security guards and 100 control points to determine the level of accuracy compared with the users' predictions. The reason is that the system calculates the optimum route for each security guard and not for the entire control points set.

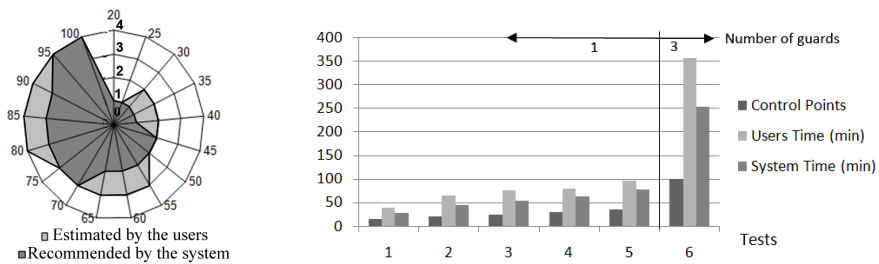


Fig. 4. Average number of estimated security guards (left) and time calculated by the users and the system with different number of control points (right)

The results obtained so far are positive. It is possible to determine the number of security guards needed to cover an entire area and the loops in the routes, so the human resources are optimized. In addition, the system provides the supervisors relevant information to monitor the workers activities, detecting incidences in the surveillance routes automatically and in real-time.

The use of wireless technologies, such as Wi-Fi, RFID, or GPRS provides an adequate communication infrastructure that the agents can use to obtain information about the context. With this information, the system can adapt services and interact

with users according a specific situation in an easy, natural and ubiquitous way to solve some of daily life problems.

The system presented can be easily adapted to other scenarios with similar characteristics, providing a simple but powerful tool to optimize human resources and monitor the staff activities. However, this system is still under development, continuously adding new capabilities and services to have the enough robustness to implement it on other scenarios.

Acknowledgments. This development has been supported by the project GERMAP (IMSERSO 137/2007). Special thanks to Telefónica Móviles and Sokymat for the technology provided.

References

1. Bauer, B., Huget, M.P.: FIPA Modeling: Agent Class Diagrams (2003)
2. 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* 44(2), 382–396 (2008)
3. Corchado, J.M., Bajo, J., Abraham, A.: GERAmI: Improving the delivery of health care. *IEEE Intelligent Systems. Special Issue on Ambient Intelligence* 23(2), 19–25 (2008)
4. Inology: Press note (June 9, 2005),
http://www.controldetiempos.com/sala_de_prensa.htm#absentismo
5. Jin, H.D., Leung, K.S., Wong, M.L., Xu, Z.B.: An Efficient Self-Organizing Map Designed by Genetic Algorithms for the Traveling Salesman Problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 33(6), 877–888 (2003)
6. Kleindienst, J., Macek, T., Seredi, L., Sedivy, J.: Vision-enhanced multi-modal interactions in domotic environments. *IBM Voice and systems technologies, Czech Republic* (2004)
7. Kohonen, T.: *Self-Organizing Maps*. Springer, New York (1997)
8. Leung, K.S., Jin, H.D., Xu, Z.B.: An expanding Self-organizing Neural Network for the Traveling Salesman Problem. *Neurocomputing* 62, 267–292 (2004)
9. Martín, Q., Santos, M.T., De Paz, Y.: *Operations research: Resolute problems and exercises*, Pearson, pp. 189–190 (2005)
10. Rigole, P., Holvoet, T., Berbers, Y.: Using Jini to integrate home automation in a distributed software-system. *Computational Sciences Department, Belgium* (2002)
11. Garfinkel, S., Rosenberg, B.: *RFID: Applications, security, and privacy*, pp. 15–36. Addison-Wesley Professional, Reading (2005)
12. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems* 3, 285–312 (2000)

Classification Agent-Based Techniques for Detecting Intrusions in Databases

Cristian Pinzón¹, Yanira De Paz², and Rosa Cano³

¹ Universidad Tecnológica de Panamá, Av. Manuel Espinosa Batista, Panama

² Universidad Europea de Madrid, Tajo s/n 28670, Villaviciosa de Odón, Spain

³ Instituto Tecnológico de Colima, Av. Tecnológico s/n, 28976, Mexico

cristian.pinzon@utp.ac.pa, yanirarosario.depaz@uem.es,
rdegca@gmail.com

Abstract. This paper presents an agent specially designed for the prevention and detection of SQL injection at the database layer of an application. The agent incorporates a Case-based reasoning mechanism whose main characteristic involves a mixture of neural networks that carry out the task of filtering attacks. The agent had been tested and the results obtained are presented in this study.

Keywords: SQL injection, multiagent systems, case-based reasoning, neural networks.

1 Introduction

Database security is a fundamental aspect of all current information systems. There are many ways of exploiting the security vulnerability in a relational database. SQL injection is one of more common types of attacks at the database layer of desktop and Web applications. SQL injection occurs when the intended effect of the SQL sentence is changed by inserting SQL keywords or special symbols [1]. The problem of SQL injection attacks has been traditionally addressed by using centralized architectures [2], [3]. Because this type of solution is incomplete, several types of intrusion detection system (IDS) solutions have been proposed [4]. Although IDSs are effective, there are a number of drawbacks such as a large number of false positives and negatives, limited learning capacity, and limited ability in adapting to changes in attack patterns.

This article presents a CBR-BDI [5] deliberative agent based on the BDI (*Belief, Desire, Intention*) [6] model specifically designed for the detection and prevention of SQL injection attacks in database layers. Our study applies a novel case-based reasoning (CBR) [7] [8] classification mechanism that incorporates a mixture of neural networks capable of making short term predictions [9].

This proposal is an innovative approach that addresses the problem of SQL injection attacks by means of a distributed artificial intelligence technique. Specifically, it combines the characteristics of multiagent systems such as autonomy, pro-activity, social relations, etc., [5] with CBR [7]. CBR Systems are adequate in dealing with

SQL injection attacks, inasmuch as these systems find solutions to new problems by using previous experiences. This fact allows us to equip our classifier agents with a great capacity for adapting and learning, thus making them very adept in resolving problems in dynamic environments. The system developed within the scope of this work proposes a solution which combines a distributed approach and an advanced classification system, incorporating the best of both approaches.

The rest of the paper is structured as follows: section 2 presents the problem that has prompted most of this research work. Section 3 focuses on the structure of the classifier agent which facilitates the detection and prevention of malicious injection attacks, and section 4 explains in detail the classification model integrated within the classifier agent. Finally, section 5 describes how the classifier agent has been tested inside a multi-agent system and presents the results obtained.

2 SQL Injection Problem Description

A SQL injection attack affects the security of personal, social, financial and legal information for both individuals and organizations. A SQL injection attack takes place when a hacker changes the semantic or syntactic logic of a SQL text string by inserting SQL keywords or special symbols within the original SQL command that will be executed at the database layer of an application [1]. SQL injection attacks occur when user input variables are not strongly typed, thus making them vulnerable to attack. As a result, these attacks can produce unauthorized handling of data, retrieval of confidential information, and in the worst possible case, taking over control of the application server [2]. One of the biggest problems with SQL injection is the various forms of vulnerabilities that exist. Some of the better known strategies, such as tautologies, syntax errors or illegal queries, and *union* operators, are easy to detect. However other strategies can be extremely complex due to the high number of variables that they can generate, thus making their detection very difficult. Some examples of these strategies are inference mechanisms, data storage procedures, and alternative encoding.

Traditional security mechanisms such as firewalls or IDSs are not very efficient in detecting and preventing these types of attacks. Other approaches based on string analysis, along with dynamic and static analyses such as AMNESIA (Analysis and Monitoring for Neutralizing SQL Injection Attacks) [2], have the disadvantage of addressing just one part of the problem, and therefore deliver only a partial solution. Moreover, the approaches based on models for detecting SQL injection attacks are very sensitive. With only slight variations of accuracy, they generate a large number of false positive and negatives.

Some innovative proposals are incorporating artificial intelligence and hybrid systems. Web Application Vulnerability and Error Scanner (WAVES) [10] uses a black-box technique which includes a machine learning approach. Valeur [4] presents an IDS approach which uses a machine learning technique based on a dataset of legal transactions. These are used during the training phase prior to monitoring and classifying malicious accesses. Rietta [11] proposed an IDS at the application layer using an anomaly detection model which assumes certain behaviour of the traffic generated by the SQL queries; that is, elements within the query (sub-queries, literals, keyword

SQL). It also applies general statistics and proposes grouping the queries according to SQL commands and then comparing them against a previously built model. The SQL query that deviates from the normal profile is rejected. Finally, Skaruz [12] proposes the use of a neural network. The detection problem becomes a time serial prediction problem. All of these approaches have the problem of producing a large number of false positive and false negative. In the case of the IDS systems, they are unable to recognize unknown attacks because they depend on a signature database.

Our technique is based on BDI agents that incorporate a CBR mechanism. This is a novel approach in detecting and preventing SQL injection attacks. This strategy, which is depicted in the following sections, offers a robust and flexible solution for confronting SQL injection attacks, thus improving and surpassing previous approaches.

3 Classifier Agent Internal Structure

Agents are characterized by their autonomy; which gives them the ability to work in independently and real-time environments [13]. Because of this and other capacities they have, agents are being integrated into security approaches, such as is the case with IDS. Some applications of these systems can be found in [14], [15]. However, the use of agents in these systems is geared towards the retrieval of information in distributed environments, thus taking advantage of their mobility capacity. The classifier agent depicted in this work is the core of a multi-agent architecture, focused on detecting and preventing SQL injection attacks. The classifier agent interacts with other complementary agents, which are dedicated to tasks such as monitoring traffic, pattern matching, manage and interacting with both the user and the database. This agent is in charge of classifying SQL queries by means of an anomaly detection approach.

In our work, the agents are based on a BDI model in which beliefs are used as cognitive aptitudes, desires as motivational aptitudes, and intentions as deliberative aptitudes in the agents [5]. However, in order to focus on the problem of SQL injection, it was necessary to provide the agents with a greater capacity of learning and adaptation, as well as, a greater level of autonomy than a pure BDI model possesses. This is possible by providing the classifier agents with a CBR mechanism [7], which allows them to “reason” on their own and adapt to changes in the patterns of attacks. Working with this type of systems, the key concept is that of “case”. A case is defined as a previous experience and is composed of three elements: a description of the problem that depicts the initial problem; a solution that describes the sequence of actions performed in order to solve the problem; and the final state, which describes the state that has been achieved once the solution is applied. To introduce a CBR motor into a BDI agent, we represent CBR system cases using BDI and implement a CBR cycle which consists of four steps: retrieve, reuse, revise and retain [7][16].

The classifier CBR-BDI [5] agent analyses a new query at the data base layer, executing a CBR cycle. In the retrieval phase, the cases that present a description similar to that of the new problem are recovered. The case description is based on elements of the SQL query that are extracted through a syntactic analyze process applied to the SQL text string. The process of retrieving the cases involves a similarity algorithm.

Once the similar cases have been recovered, the reuse phase starts adapting the solution in order to obtain the best case output for the given object of study. In this phase, two neural networks are responsible for receiving the variables as input data in order to produce an output response. This is done by a mixture of neural networks which use both sigmoidal activation functions and tangential functions. The next phase of the CBR cycle is the revise phase in which an expert evaluates the proposed solution. Finally, the retain phase is carried out where the system learns from its own experiences from each of the previous phases and updates the database with the solution obtained in the query classification.

As has been mentioned above, the classifier CBR-BDI agent is the core of a multi-agent architecture geared towards to detect and classify SQL injection attack in distributed environments. Figure 1 shows the classifier CBR-BDI agent in the multi-agent architecture.

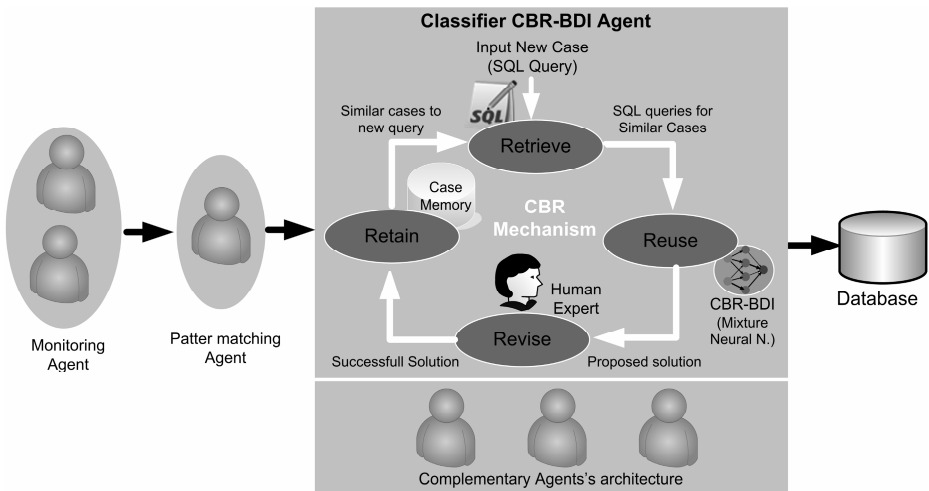


Fig. 1. CBR-BDI agent in the multi-agent architecture

All the SQL queries are filtered by the classifier CBR-BDI agent. The query considered legal is executed on the database otherwise the query considered malicious is rejected. In the next section, the classification mechanism used in the reuse phase by the agent is presented.

4 Mechanism for the Classification of SQL Injection Attacks

The classifier agent presented in section 3 incorporates a case-based reasoning system that allows the prevention and detection of anomalies by means of a prediction model based on neural networks, configured for short-term predictions of intrusions by SQL injections. This mechanism uses a memory of cases which identifies past experiences with the corresponding indicators that characterize each of the attacks. This paper presents a novel classification system that combines the advantages of the CBR systems, such as learning and adaptation, with the predictive capabilities of a mixture of

neural networks. These features make the system innovative for this type of attack, and make it very appropriate for its use in different environments. The proposed mechanism is responsible for classifying SQL database requests made by users. When a user makes a new request by means of a SQL query, it is checked by an agent in charge of detection by pattern matching. If it is found an attack firm on the SQL string, then the SQL query is labelled as suspicious. Independently of the results of the pattern matching, the SQL query is send to a CBR-BDI agent. This query is transformed to a new case of input for a CBR mechanism. The case is constituted by elements extracted of the SQL string (Affected_Table, Affected_Field, Command_Type, Word_GroupBy, Word_OrderBy, Word_Having, Number_And, Number_Or, Number_Literales, Number_LOL, Length_SQL_String, Cost_Time_CPU, Start_time, End_time, Query_Category). The CBR mechanism needs a memory of cases dating back at least 4 weeks for the training stage of the neural networks.

The first phase of the CBR cycle consists of recovering past experience from the memory of cases, specifically those with a problem description similar to the current request, In order to do this, a cosine similarity-based algorithm is applied, allowing the recovery of those cases which are at least 90% similar to the current request. The cases recovered are used to train the mixture of neural networks implemented in the recovery phase; the neural network with the sigmoidal function is trained with the recovered cases that were an attack or not, whereas the neural network with hyperbolic function is trained with all the recovered cases (including the suspects). A preliminary analysis of correlations is required to determine the number of neurons of the input layer of the neuronal networks. Additionally, it is to normalize the data (i.e., all data must be values in the interval [0.1]). The data used to train the mixture of networks must not be correlated. With the cases stored after eliminating correlated cases, the entries for training the mixture of networks are normalized. It is considered to be two neural networks. The result obtained using a mixture of the outputs of the networks provides a balanced response and avoids individual tendencies (always taking into account the weights that determine which of the two networks is more optimal).

La mixture of the neural networks intents to solve the individual tendencies by use an only neural network. If one only network with a sigmoidal activation function is used, then the result provided by the network would tend to be attack or not attack, and no suspects would be detected. On the other hand, if only one network with a hyperbolic tangent activation is used, then a potential problem could exist in which the majority of the results would be identified as suspect although they are being clearly attack or not attack. Figure 2 shows the mixture of the neural networks with their activation function.

The mixture provides a more efficient configuration of the networks, since the global result is determined by merging two filters. This way, if the two networks classify the user request as an attack, so too will the mixture; and if both agree that it is not an attack, the mixture will as well. If there is no concurrence, the system uses the result of the network with the least error in the training process or classifies it as a suspect. In the reuse phase the two networks are trained by a back-propagation algorithm for the same set of training patterns (in particular, these neural networks are named Multilayer Perceptron), using a sigmoidal activation function (which will take values in [0.1], where 0 = Illegal and 1 = legal) for a Multilayer Perceptron and a hyperbolic tangent activation function for the other Multilayer Perceptron (which take values in [-1.1], where -1 = Suspect, 0 = illegal and 1 = legal).

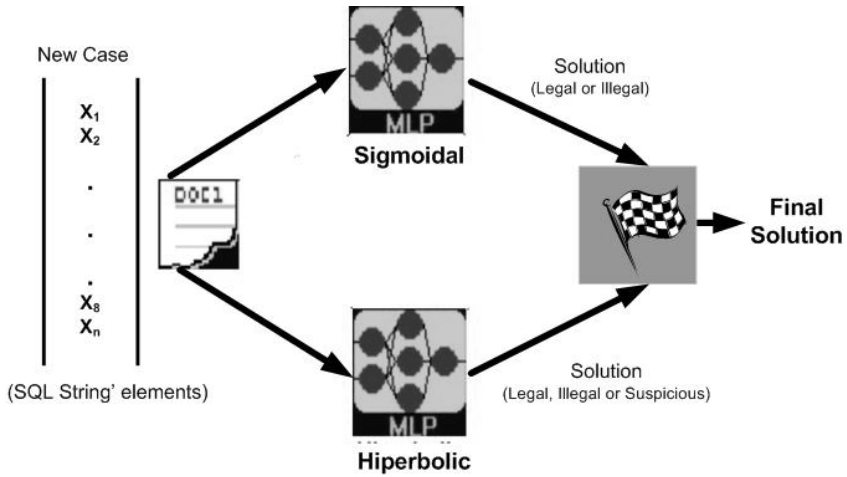


Fig. 2. Mixture of the Neural Networks

The response of both networks is combined, obtaining the mixture of networks denoted by y^2 ; where the superscript indicates the number of mixed networks

$$y^2 = \frac{1}{\sum_{r=1}^2 e^{-|1-r|}} \sum_{r=1}^2 e^{-|1-r|} y^r \quad (1)$$

The number of neurons in the output layer for both Multilayer Perceptrons is 1, and is responsible for deciding whether or not there is an attack. The error of the training phase for each of the neural networks, can be quantified with formula (2), where P is the total number of training patterns.

$$Error = \frac{1}{P} \sum_{i=1}^P \left| \frac{Forecast_p - Target_p}{Target_p} \right| \quad (2)$$

The review stage is performed by an expert, and depending on his opinions, a decision is made as to whether the case is stored in the memory of cases and whether the list of well-known patterns has to be updated in the retain phase.

5 Results and Conclusions

The problem of SQL injection attacks on databases supposes a serious threat against information systems. This paper has presented a new classification system for detecting SQL injection attacks which combines the advantages of multi-agent systems, such as autonomy and distributed problem solving, with the adaptation and learning capabilities of CBR systems. Additionally, the system incorporates the prediction capabilities that characterize neural networks. An innovative model has been presented that provides a significant reduction of the error rate during the classification of attacks. To check the validity of the proposed model, a series of test were

elaborated which were executed on a memory of cases, specifically developed for these tests, and which generated attack consults. The results shown in Table 1 are promising: it is possible to observe different techniques for predicting attacks at the database layer and the errors associated with misclassifications. All the techniques presented in Table 1 have been applied under similar conditions to the same set of cases, taking the same problem into account in order to obtain a new case common to all the methods. Note that the technique proposed in this article provides the best results, with an error in only 0.537% of the cases.

Table 1. Results obtained after testing different classification techniques

Forecasting Techniques	Successful (%)	Approximated Time (secs)
CBR-BDI Agent (mixture NN)	99.5	2
Back-Propagation Neural Networks	99.2	2
Bayesian Forecasting Method	98.2	11
Exponential Regression	97.8	9
Polynomial Regression	97.7	8
Linear Regression	97.6	5

As shown in Table 2, the Bayesian method is the most accurate statistical method since it is based on the likelihood of the events observed. But it has the disadvantage of determining the initial parameters of the algorithm, although it is the fastest of the statistical methods. Taking the errors obtained with the different methods into account, after the CBR-BDI Agent together with the mixture of neural networks and Bayesian methods we find the regression models. Because of the non linear behaviour of the hackers, linear regression offers the worst results, followed by the polynomial and exponential regression. This can be explained by looking at hacker behaviour: as the hackers break security measures, the time for their attacks to obtain information decreases exponentially. The empirical results show that the best methods are those that involve the use of neural networks and, if we consider a mixture of two neural networks, the predictions are notably improved. These methods are more accurate than statistical methods for detecting attacks to databases because the behaviour of the hacker is not linear. The solution presented is an innovative approach to detect SQL injection attack, when it is based on multi-agent system, CBR mechanism and a sophisticated technique using a mixture of neural networks.

Acknowledgments. This development has been partially supported by the Spanish Ministry of Science project TIN2006-14630-C03-03.

References

1. Anley, C.: Advanced SQL Injection In SQL Server Applications (2002), <http://www.nextgenss.com/papers/advanced-sql-injection.pdf>
2. Halfond, W., Orso, A.: AMNESIA: analysis and monitoring for neutralizing SQL-injection attacks. In: ASE 2005: 20th IEEE/ACM international Conference on Automated software engineering, pp. 174–183. ACM, New York (2005)

3. Wassermann, G., Gould, C., Su, Z., Devanbu, P.: Static Checking of Dynamically Generated Queries in Database Applications. *ACM Transactions on Software Engineering and Methodology* 16, 14 (2007)
4. Valeur, F., Mutz, D., Vigna, G.: A Learning-Based Approach to the Detection of SQL Attacks. In: Julisch, K., Krügel, C. (eds.) *DIMVA 2005*. LNCS, vol. 3548, pp. 123–140. Springer, Heidelberg (2005)
5. Corchado, J.M., Pavón, J., Corchado, E.S., Castillo, L.F.: Development of CBR-BDI Agents. In: *Advances in Case-Based Reasoning*. Springer, Heidelberg (2004)
6. Woolridge, M., Wooldridge, M.J.: *Introduction to Multiagent Systems*. John Wiley & Sons, New York (2002)
7. Corchado, J.M., Laza, R., Borrajo, L., De Luis, Y.A., Valiño, M.: Increasing the Autonomy of Deliberative Agents with a Case-Based Reasoning System. *International Journal of Computational Intelligence and Applications* 3(1), 101–118 (2003)
8. Fdez-Riverola, F., Iglesias, E.L., Daz, F., Méndez, J.R., Corchado, J.M.: SpamHunting: An instance-based reasoning system for spam labelling and filtering. *Decision Support System* 43(3), 722–736 (2007)
9. Ramasubramanian, P., Kannan, A.: Quickprop Neural Network Ensemble Forecasting a Database Intrusion Prediction System. *Neural Information Processing* 5, 847–852 (2004)
10. Huang, Y., Huang, S., Lin, T., Tsai, C.: Web application security assessment by fault injection and behavior monitoring, pp. 148–159. ACM, New York (2003)
11. Rietta, F.: Application layer intrusion detection for SQL injection. In: 44th annual Southeast regional conference, pp. 531–536. ACM, New York (2006)
12. Skaruz, J., Serebinski, F.: Recurrent neural networks towards detection of SQL attacks. In: *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007*, pp. 1–8. IEEE International, Los Alamitos (2007)
13. Carrascosa, C., Bajo, J., Julian, V., Corchado, J.M., Botti, V.: Hybrid multi-agent architecture as a real-time problem-solving model. *Expert System with Application* 34, 2–17 (2008)
14. Kussul, N., Shelestov, A., Sidorenko, A., Skakun, S., Veremeenko, Y.: Intelligent multi-agent information security system, *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. In: *Proceedings of the Second IEEE International Workshop*, pp. 120–122 (2003)
15. Abraham, A., Jain, R., Thomas, J., Han, S.Y.: D-SCIDS: distributed soft computing intrusion detection system. *J. Netw. Comput. Appl.* 30, 81–98 (2007)
16. Corchado, J.M., Bajo, J., Abraham, A.: GerAmi: Improving Healthcare Delivery in Geriatric Residences. *Intelligent Systems* 23, 19–25 (2008)

Hybrid Multi-Agent Architecture (HoCa) Applied to the Control and Supervision of Patients in Their Homes

Juan A. Fraile¹, Dante I. Tapia², and Miguel A. Sánchez³

¹ Centro de Proceso de Datos,

² R&D Department, Tulecom Group S.L.
Hoces del Duratón 57, 37008, Salamanca, Spain

³ Escuela Universitaria de Informática,
Pontifical University of Salamanca,
Compañía 5, 37002 Salamanca, Spain

jafraileni@upsa.es, dante.tapia@tulecom.com, masanchezvi@upsa.es

Abstract. This paper presents a Hybrid Multi-Agent Architecture based on an Ambient Intelligence model, for the control and supervision of dependent environments. HoCa architecture incorporates a management system of alerts based on SMS and MMS technologies, and an automated identification, localization, and movement control system based on Java Card and RFID technologies. HoCa is independent from the programming language and operating system in that it is executable. The core of the architecture is formed by both deliberative agents and reactive agents that interact to offer efficient services. The architecture has been tested in a real environment and the results obtained are presented in this paper.

Keywords: Dependent environments, Ambient Intelligence, Multiagent Systems, Home Care.

1 Introduction

There is currently considerable growth in the development of automation technologies, such as home automation and Ambient Intelligence (AmI) [1] [14]. One of their principal objectives is to look after the user's well-being and obtain a more friendly, rational, productive, sustainable and secure relationship for users within their environments. Several architectures based on agent utilization have emerged thanks to the appearance of intelligent spaces and the integration of devices that are programmable via computer networks [15]. These have stimulated the development of ubiquitous computation [9], which is the most promising technological approximation for resolving the challenge of developing strategies that allow the early detection and prevention of problems in an automated environment.

The main objective of this paper is to define a hybrid Multi-Agent Architecture for the control and the supervision of open environments. It involves developing an architecture that allows automated identification, localization, alarms management and control of movement. The users who utilize the system in which this architecture is applied will be able to gain wireless access to all the information that they need to

perform their work. The novel innovation at the core of the architecture is a real time communication protocol that allows secure and rapid communication between the reactive agents and the system sensors. These reactive agents, whose response time is critical, are influenced by deliberative BDI agents, which are located inside the platform given that a very fluid communication already exists between them. Additionally, the architecture manages an alert or alarm system across the agents' platform specially designed to work with mobile devices. The alert system contains different levels of urgency. The alert level is determined by the deliberative agent who, depending on the alert level, then emits the alert to either a reactive agent or a deliberative agent.

The paper is organized as follows: The second section presents the problem that prompted this work. The third section presents the proposed architecture, and the fourth section gives the results and conclusions obtained after applying the proposed architecture to a real case in an environment of dependence.

2 General Description of the Problem

The use of intelligent agents is an essential component for analyzing information on distributed sensors [16] [18]. These agents must be capable of both independent reasoning and joint analysis of complex situations in order to be able to achieve a high level of interaction with humans [3] [4]. Although multi-agent systems already exist and are capable of gathering information within a given environment in order to provide medical care [12], there is still much work to be done. It is necessary to continue developing systems and technology that focus on the improvement of services in general. After the development of the internet there has been continual progress in new wireless communication networks and mobile devices such as mobile telephones and PDAs. This technology can help to construct more efficient distributed systems capable of addressing new problems [7].

Hybrid architectures try to combine deliberative and reactive aspects, by combining reactive and deliberative modules. The reactive modules are in charge of processing stimuli that do not need deliberation, whereas the deliberative modules determine which actions to take in order to satisfy the local and cooperative aims of the agents. The aim of modern architectures like Service Oriented Architecture (SOA) is to be able to interact among different systems by distributing resources or services without needing to consider which system they are designed for. An alternative to these architectures are the multi-agent systems, which can help to distribute resources and to reduce the centralization of tasks. Unfortunately the complexity of designing multi-agent architecture is great since there are no tools to either help programme needs or develop agents.

Multi-agent systems combine aspects of both classic and modern architectures. The integration of multi-agent systems with SOA and web services has been recently investigated [2]. Some investigators focus on the communication among these models, whereas others focus on the integration of distributed services, especially web services, in the agents' structure [5] [13] [17].

These works provide a good base for the development of multi-agent systems. Because the majority of them are in the development stage, their full potential in a real

environment is not known. HoCa has been implemented in a real environment and not only does it provide communication and integration among distributed agents, services and applications, but it also provides a new method for facilitating the development of multi-agent systems, thus allowing the agents and systems to function as services.

HoCa implements an alert and alarm system across the agent's platform, specially designed to be used by mobile devices. The platform agents manage this service and determine the level of alert at every moment so that they can decide who will receive the alert and when. In order to identify each user, HoCa implements a system based on Java Card and RFID (Radio Frequency IDentification) microchip technology in which there will be a series of distributed sensors that provide the necessary services to the user.

3 Proposed Architecture

The HoCa model architecture uses a series of components to offer a solution that includes all levels of service for various systems. It accomplishes this by incorporating intelligent agents, identification and localization technology, wireless networks and mobile devices. Additionally, it provides access mechanisms to multi-agent system services, through mobile devices, such as mobiles phones or PDAs. Access is provided via wi-fi wireless networks, a notification and alarm management module based on SMS and MMS technologies, and user identification and localization system based on Java Card and RFID technologies. This system is dynamic, flexible, robust and very adaptable to changes of context.

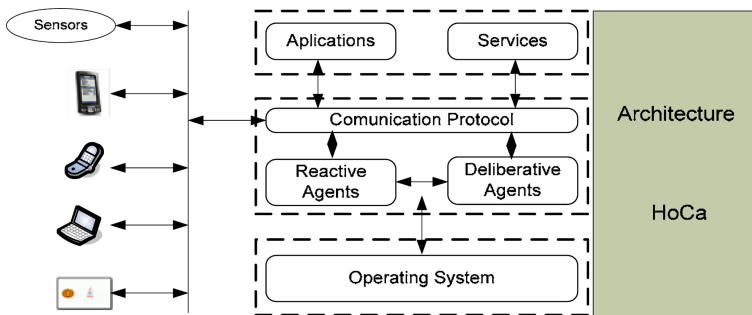


Fig. 1. HoCa Framework

HoCa architecture describes four basic blocks that can be seen in Figure 1: Applications, Services, Agents Platform and Communication Protocol. These blocks constitute the whole functionality of the architecture.

3.1 Agents Platform in HoCa

The agents platform is the core of the architecture and integrates two types of agents, each of which behaves differently for specific tasks.

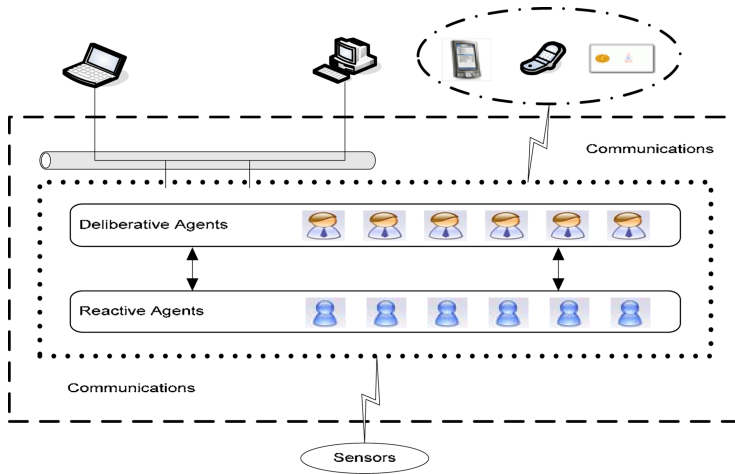


Fig. 2. Agents platform structure in the HoCa architecture

The first group of agents is made up of deliberative BDI agents, which are in charge of the management and coordination of all system applications and services. These agents are able to modify their behaviour according to the preferences and knowledge acquired in previous experiences, thus making them capable of choosing the best solution. Deliberative agents constantly deal with information and knowledge. Because they can be executed on mobile devices, they are always available for the users regardless of physical location.

The second group is made up of reactive agents. Most of the research conducted within the field of multi-agent systems focuses on designing architectures that incorporate complicated negotiation schemes as well as high level task resolution, but don't focus on temporal restrictions. In general, the multi-agent architectures assume a reliable channel of communication and, while some establish deadlines for the interaction processes, they don't provide solutions for limiting the time the system may take to react to events.

It is possible to define a real-time agent as an agent with temporal restrictions for some of its responsibilities or tasks [11]. From this definition, we can define a real-time multi-agent system (Real Time Multi-Agent System, RT-MAS) as a multi-agent system in which at least one of the agents is a real-time agent [6]. The use of RT-MAS makes sense within an environment of critical temporal restrictions, where the system can be controlled by autonomous agents that need to communicate among themselves in order to improve the degree of system task completion. In this kind of environments every agent requires autonomy as well as certain cooperation skills to achieve a common goal.

3.2 HoCa Communication Protocol

Communication protocol allows applications, services and sensors to be connected directly to the platform agents. The protocol presented in this work is open and independent of programming languages. It is based on the SOAP standard and allows messages to be exchanged between applications and services as shown in Figure 3.

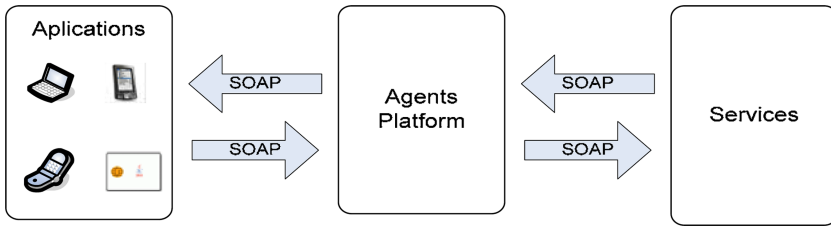


Fig. 3. Communication using SOAP messages in HoCa

However, interaction with environmental sensors requires Real-time Transport Protocol (RTP) [10] [6] which provides transport functions that are adapted for applications that need to transmit real-time data such as audio, video or simulation data, over multicast or unicast network services. The RTCP protocol is added to RTP, allowing a scalable form of data supervision. Both RTP and RTCP are designed to work independently from the transport and lower network services. They are in charge of transporting data with real-time characteristics, and of supervising the quality of service, managing the information for all the entities taking part in the current session.

The communications between agents within the platforms follows the FIPA ACL (Agent Communication Language) standard. This way, the applications can use the platform to communicate directly with the agents.

3.3 Location and Identification System in HoCa

This system incorporates Java Card [19] and RFID [8] technologies. The primary purpose of the system is to convey the identity of an object or person, as with a unique serial number, using radio waves. Java Card is a technology that permits small Java applications (applets) to be run safely in microchip smart cards and similar embedded devices. Java Card gives the user the ability to program applications that can be run off a card so that it has a practical function in a specific application domain. The main features of Java Card are portability and security. The data are stored in the application and the Java Card applications are executed in an isolated environment, separate from the operating system and from computer that reads the card. The most commonly used algorithms, such as DES, 3DES, AES, and RSA, are cryptographically implemented in Java Card. Other services such as electronic signature or key generation are also supported.

RFID technology is grouped into the so-called automatic identification technologies. But RFID provides more information than other auto-identification technologies, speeds up processes without losing reliability, and requires no human intervention.

The combination of these two technologies allows us to both identify the user or identifiable element, and to locate it, by means of sensors and actuators, within the environment, at which time we can act on it and provide services. The microchip, which contains the identification data of the object to which it is adhered, generates a radio frequency signal with this data. The signal can be picked up by an RFID reader, which is responsible for reading the information and sending it, in digital format, to the specific application.

3.4 Alerts System in HoCa

The alert system is integrated into the HoCa architecture and uses mobile technology to inform users about alerts, warnings and information specific to the daily routine of the application environment.

This is a very configurable system that allows users to select the type of information they are interested in, and to receive it immediately on their mobile phone or PDA. It places the information to be sent into categories and subcategories of information. The users determine the information they are interested in. The system automatically sends the information to each of the users as soon as it is available.

4 Results and Conclusions

HoCa has been used to develop a multi-agent system for monitoring dependent patients at home. The main features of this system include reasoning and planning mechanisms, and alert and response management. Most of these responses are reactions in real time to certain stimuli, and represent the abilities that the reactive agents have in the HoCa architecture based platform. To offer all these features the system uses various technologies and acquires information from users and the surrounding environment. Some of the technologies used to test the system include mobile technology for managing service alerts through PDAs and mobile phones, and Java Card technology for identification and access control.

One of the main contributions of the HoCa architecture is the alert system. We implemented several test cases to evaluate the management of alerts integrated into the system. This allowed us to determine the response time for warnings generated by the users, for which the results were very satisfactory, with response times shorter than those obtained prior to the implementation of HoCa. In addition, the system studies the information collected, and applies a reasoning process which allows alerts to be automatically generated. For these alerts, the system does not only take response time into account, but also the time elapsed between alerts, and the user's profile and reliability, in order to generalize reactions to common situations. The results show that HoCa fits perfectly within complex systems by correctly exploiting services and planning mechanisms.

Table 1. Comparison between the HoCa and the ALZ-MAS architectures

Factor	HoCa	ALZ-MAS
Average Response Time to Incidents(min.)	8 minutes	14 minutes
Assisted Incidents	12	17
Average number of daily planned tasks	12	10
Average number of services completed daily	46	32
Time employed by the medical staff to attend to an alert (min.)	75 minutes	90 minutes

Table 1 presents the results obtained after comparing the HoCa architecture to the previously developed ALZ-MAS architecture [7] in a case study on medical care for patients at home. The ALZ-MAS architecture allows the monitoring of patients in geriatric residences, but home care is carried out through traditional methods. The case study presented in this work consisted of analysing the functioning of both architectures in a test environment. The HoCa architecture was implemented in the home of 5 patients and was tested for 30 days. The results were very promising. The data shown in Table 1 are the results obtained from the test cases. They show that the alert system improved the communication between the user and the dependent care services providers, whose work performance improved, allowing them to avoid unnecessary movement such as travels and visits simply oriented to control or supervise the patient. The user identification and location system in conjunction with the alert system has helped to notably reduce the percentage of incidents in the environment under study. Moreover, in addition to a reduction in the number of incidents, the time elapsed between the generation of a warning and solution decreased significantly. Finally, due to the many improvements, the level of user satisfaction increased with the introduction of HoCa architecture since patients can live in their own homes with the same level of care as those offered at the residence.

Acknowledgements. This development has been partially supported by the Spanish Ministry of Science project TIN2006-14630-C03-03.

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. Ardissono, L., Petrone, G., Segnan, M.: A conversational approach to the interaction with Web Services. In: Computational Intelligence, vol. 20, pp. 693–709. Blackwell Publishing, Malden (2004)
3. Bahadori, S., Cesta, A., Grisetti, G., Iocchi, L., Leone, R., Nardi, D., Oddi, A., Pecora, F., Rasconi, R.: RoboCare: an Integrated robotic system for the domestic care of the elderly. In: Proceedings of workshop on Ambient Intelligence, Pisa, Italia (2003)
4. Bahadori, S., Cesta, A., Grisetti, G., Iocchi, L., Leone, R., Nardi, D., Oddi, A., Pecora, F., Rasconi, R.: RoboCare: Pervasive Intelligence for the Domestic Care of the Elderly. In: AI*IA Magazine Special Issue (January 2003)
5. Bonino da Silva, L.O., Ramparany, F., Dockhorn, P., Vink, P., Etter, R., Broens, T.: A Service Architecture for Context Awareness and Reaction Provisioning. In: IEEE Congress on Services (Services 2007), pp. 25–32 (2007)
6. Carrascosa, C., Bajo, J., Julian, V., Corchado, J.M., Botti, V.: Hybrid multi-agent architecture as a real-time problem-solving model. *Expert Systems With Applications* 34(1), 2–17 (2008)
7. Corchado, J.M., Bajo, J., Abraham, A.: GERAmI: Improving the delivery of health care. *IEEE Intelligent Systems. Special Issue on Ambient Intelligence* 23(2), 19–25 (2008)
8. Garfinkel, S., Rosenberg, B.: RFID: Applications, security, and privacy, pp. 15–36. Addison-Wesley Professional, Reading (2005)

9. Kleindienst, J., Macek, T., Seredi, L., Sedivy, J.: Vision-enhanced multi-modal interactions in domotic environments. IBM Tecnologías de Voz y Sistemas. República Checa (2004)
10. Jacobsen, V., Fredrick, R., Casner, S., Schulzrinne, H.: RTP: A transport Protocol for Real-Time Applications. RFC 1889. Lawrence Berkeley National Laboratory, Xerox PARC, Precept Software Inc., GMD Fokus (January 1996) (Accessed October 16, 1996), <http://www.connect.org.uk/teckwatch/cgi-bin/rfcshow?1889>
11. Julian, V., Botti, V.: Developing real-time multi-agent systems. In: Proceedings of the Fourth Iberoamerican Workshop on Multi-Agent Systems (Iberagents 2002), Málaga (2004)
12. Mengual, L., Bobadilla, J., Triviño, G.: A fuzzy multi-agent system for secure remote control of a mobile guard robot. In: Favela, J., Menasalvas, E., Chávez, E. (eds.) AWIC 2004. LNCS (LNAI), vol. 3034, pp. 44–53. Springer, Heidelberg (2004)
13. 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)
14. Richter, K., Hellenschmidt, M.: Interacting with the Ambience: Multimodal Interaction and Ambient Intelligence. In: Position Paper to the W3C Workshop on Multimodal Interaction, July 19-20 (2004)
15. Rigole, P., Holvoet, T., Berbers, Y.: Using Jini to integrate home automation in a distributed software-system. Departamento de Ciencias Computacionales. Leuven, Bélgica (2002)
16. Tapia, D.I., Bajo, J., De Paz, F., Corchado, J.M.: Hybrid Multiagent System for Alzheimer Health Care. In: Rezende, S.O., da Silva Filho, A.C.R. (eds.) Proceedings of HAIS 2006. Ribeiro Preto, Brasil (2006)
17. Walton, C.: Agency and the Semantic Web. Oxford University Press, Oxford (2006)
18. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons, Chichester (2002)
19. ZhiqunChen (Sun Microsystems). Java Card Technology for Smart Cards. Addison Wesley Longman. ISBN 0201703297

JADE/LEAP Agents in an Aml Domain*

Nayat Sánchez-Pi, Javier Carbó, and José Manuel Molina

Computer Science Department. Carlos III University of Madrid
Av. de la Universidad Carlos III, 22 Madrid, Spain
{Nayat.Sanchez, Javier.Carbo, Jose.Molina}@uc3m.es

Abstract. Ambient intelligence, ubiquitous computing and other new concepts related are in some sense a projection of the new wireless technologies and the mobile phenomenon where the user mobility and the communicating smart devices are immersed nowadays. In this paper, we propose a multi-agent system architecture and its implementation for the provisioning of context-aware services in an ambient intelligence domain: an airport.

Keywords: Multi-agent system, ambient intelligence, context-aware services.

1 Introduction

Ambient intelligence (Aml) was first introduced by the European Community [1] to identify the paradigm to equip environments with advanced technology to create an ergonomic space for the user where they could interact with their digital environments the same way they interact each other. If we assume that agents are abstractions for the interaction within an ambient intelligent environment, one aspect that we need to ensure is that their behavior is regulated and coordinated. For this purpose, we need rules that take into consideration the context (location, user profile, type of device, etc...) in which these interactions take place. Taking care this, the system needs an organization similar to the one envisaged by artificial agent societies. The society is there not only to regulate behavior but also to distribute responsibility amongst the member agents. In [2], O'Hare et al. advocate the use of agents (as well as we do) as a key enabler in the delivery of ambient intelligence.

There are several approaches developing platforms, frameworks and applications for offering context-aware services where agent technology has been applied in order to provide the right information at the right time to its users. Tourism, Healthcare, Education, Transportation, etc..., are some of the sectors where has been developed context-aware systems and applications. Such applications include location-based services like mapping and points of interest search, travel planning and, recently, pushing information and events to the user. Works like the ones carried out in CRUMPET [3] projects have addressed the state of the art and even furthered it. Recently, another clear application was presented for supporting virtual elderly assistance communities, in the context of the TeleCARE project [4]. Also [5] proposed a planning agent AGALZ which uses a

* Funded by projects MADRINET, TEC 2005-07186-C03-02, SINPROB, TSI2005-07344-C02-02 and CAM CCG06-UC3M/TIC-0781.

case-based reasoning (CBR) architecture, to respond to events and monitor Alzheimer patients' health care in execution time. Researches about developing multi-agent systems for context aware services are growing up. For instance: SMAUG [6] is a multi-agent context-aware system that allows tutors and pupils of a university to fully manage their activities. SMAUG offers its users with context-aware information from their environment and also gives them a location service to physically locate every user of the system. Another one is the BerlinTainment [7]; the project has demonstrated a framework for the provision of activity recommendations based on mobile agents. Similar to this one is the AmbieAgents [8], an agent-based infrastructure for context-based information delivery for mobile users. There is also a case study consisting in solving the automation of the internal mail management of a department that is physically distributed in a single floor of a building plant (restricted and well-known test environment) using ARTIS agent architecture [9]. Another recent work using architecture for ubiquitous computing is [10] which design a surveillance system control as a multiagent system applying an agent-orientated methodology.

What we precisely propose in this paper is a distributed solution falling on a multi-agent system architecture and its implementation for the provisioning of context-aware services in an ambient intelligence domain: an airport. We already developed a centralized system [11] using Appear Platform¹ and Aruba Wi-Fi Location System² and we will use the same case study to illustrate the difference between the distributed architecture based on agents we're proposing. This approach includes agents implemented in JADE and LEAP that will make use of the user profiles to customize and recommend different services to other agents taking advantage of the distribution of the context and avoiding an obtrusive participation of a central server. [13]

2 Case Study

In this section we will present the definition of our context-aware problem in an airport domain. We have divided our Madrid-Barajas airport domain into six different zones that are not overlapped each other. Airport Zone (containing the rest of the zones), Customs Zone (customs), Commercial Zone (stores, cafeterias, restaurants), Offices Zone (airline offices), Check-In Desk Zone (check-in desks) and Finger Zone (finger). In our ontology each zone will be represented as "place". This is part of a previous work we have referenced before [11] where we have used a centralized platform to define the behavior of our system. The centralized architecture separates the system responsibilities into: server and client software installed in the wireless devices as shown in Fig 1. The server is the central point of administration and coordination. It is where the contextual information is managed in a centralized and global way. When a wireless device enters the network it immediately establishes the connection and then the position of the user is evaluated. It is important to explain we test our system with the deployment of a Wi-Fi positioning system in order to locate the user's terminals. Particularly, we used an Aruba WLAN deployment with a 2400 Mobility Controller with capacity of 48 access points and 512 users connected at the same time and several AP 60/ AP61 which are single/dual radio wireless access

¹ www.appearnetworks.com

² www.arubanetworks.com

points. Aruba Networks has a location tracking solution which uses an enterprise wide WLAN deployment to provide precise location tracking of any Wi-Fi device in the research facility. So, once the user is located and is in contact with the server a set of applications are pushed to him depending on his context.

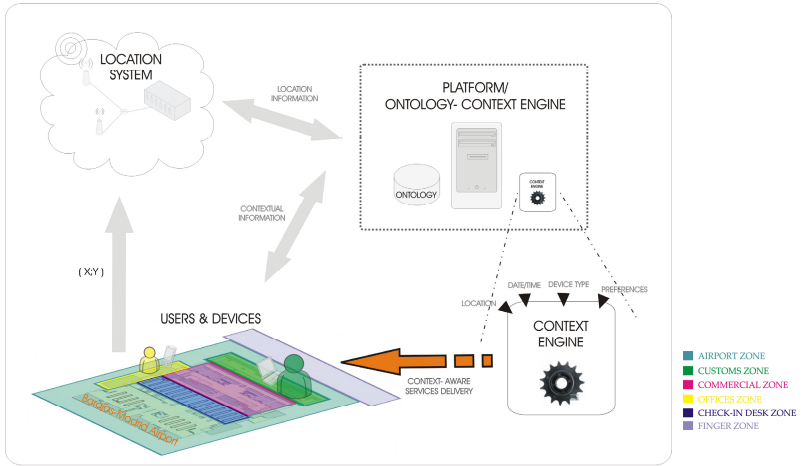


Fig. 1. Centralized architecture of the system

In any case, it is interesting to make a stop in some important context concepts we have defined for the centralized solution. We will use them again; represented in the ontology we’re representing in the distributed solution (next section).

- *Service*: An application or data available to a client
- *Category*: A set of logically grouped services
- *Offering*: A set of available applications for a client
- *Zone Mapping*: Where offerings are mapped to zones and time
- *Context Predicate*: is a set of simple rules or conditions, combined to create a more complex predicate. A predicate can match one or more context parameters available in the system.
- *Roles*: Are used to grant users access to specific categories. It is possible to specify one or more groups that have access to a category.
- *Users*: Users are only used if login is required on the system

The provisioning of the services to users occurs automatically as the right context is found once the context predicates are evaluated. This evaluation in context-aware computing systems occurs though the proper reasoning method in order to become aware of current context. If the method is not affected by the application and the used sensors, an identical context reasoning method can be used in different applications. One of the methods is rule-based systems that compare facts in working memory, which is continually updated, to a stored rule-base, which is represented by *if-then* rules, and infer system response. This is the case we have used in the centralized approach. Context information in the system is used throughout the entire life-cycle of the service: selection based on the context profile, filtering of individual services, and

enhancement of services at boot or runtime and the constant feed of context information to services during execution to allow service adaptation depending on the zone the user is placed and the role in the system users may have.

In order to be clearer, to develop our example we've also defined some generic predicates for Roles and Zones. There is one Predicate for each Role and one for each Zone. This is an example of the rules definition, Fig 2, and its evaluation for providing the SERVICE_PASSENGER CONTROL to a Passenger user who is getting into the Customs Zone. Roles are defined as: Pilot, Flight Technical Officer, Cabin Crew, Airline Staff in check-in counters, Supervisor, Airline Staff in boarding gates, Airline Staff taking care of Incidences and Passenger.

<i>SERVICE_passengers_control</i>		
	<i>p1_passengers_control is true</i>	
	<i>zones_p1_passengers_control is true</i>	
	<i>zone_customs is true</i>	
	OR <i>zone_commercial is true</i>	
	AND <i>roles_p1_passengers_control is true</i>	
	<i>ROLE_Pilot is true</i>	
	OR <i>ROLE_Fto is true</i>	
	OR <i>ROLE_CBC is true</i>	

Fig. 2. Rules definition for a passenger in customs zone

3 Distributed Solution

Our approach consists of a multi-agent system that uses context-aware information to provide customized services to users in closed environments (such as a technological fairground, university, shopping centre, etc.), where customization is based on user location and user profile. First, we identified agent types as: Central, Provider and Client agents. Central agent represents the infrastructure that acquires location information from different sensors; Provider agents act on behalf of the different services in order to distribute the context among them and Client agents represents the users with a wireless device, Fig. 3.

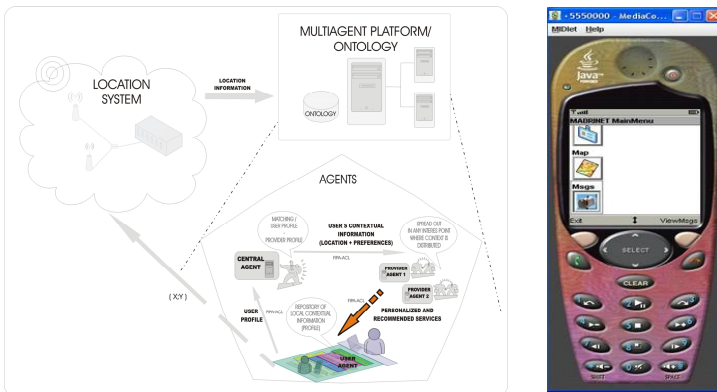


Fig. 3. Distributed architecture and UI of the multi-agent system

Later, we distinguished the roles of agents required in the system as: User Manager (UM), Location Information Manager (LIM), Agent Discover (AD), Service Provider (SP), Recommend Role (RR), Service Provider (SP) and Negotiation Role (NR).

After that we defined three different ad-hoc protocols (Table 1):

- *Receive-registry-profile*: Receive a request of registry of a user which has a shared part of the user profile.
- *Notify-agent*: Send an informative message to the closest provider in order to alert of the presence of a user.
- *Offer-service*: Send offers about services to users, that can be negotiate

Table 1. Protocols and Roles from each agent

Protocols	Roles	Agents
Receive-registry-profile	User Manager (UM) Location Information Manager (LIM)	Central
Notify-agent	Agent Discover (AD) Service Provider (SP) Recommend Role (RR)	Central Provider
Offer-service	Service Provider (SP) Negotiation Role (NR)	Provider Client

So finally, the proposed three types of agents implement the following functionalities:

- Central agent's main tasks relies on detecting users, registering and deregistering users, improving shared user profiles, filtering and notifying closer providers. Central agent manages the main system activities and it is supported on provider agents in order to balance the activity load.
- Provider agent's functionality are closely related with dialogue between client agents and provider agents, since they can reach an agreement with clients and communicate the most suitable services to them according clients preferences and profile. Contextual information is distributed between the providers.
- Client agent's main goals includes negotiation with providers agents, recommend services to other client agents, trust in other agents, and manage and improve their internal profile to receive better services according to it.

Furthermore, an ontology has been implemented for the agent communication instantiating a high level of conceptualization for all possible domains in which system runs (technological fairground, university, shopping centre, etc.). The ontology includes the following concepts: framework, location, spatial region, temporal region, place, profile, service, product, device, predicate, agent action, agent identifier, as it is explained in our previous publications [11, 12, 13]. The relationship between these concepts is shown in Figure 4. We have made a matching between this high level for dynamic environments, and the lower levels for the airport domain:

- Place concept refers to any interest point that can be defined in the airport environment. In this case the main place is the airport itself, and inside of it: Check_in_Desk, Finger, Wifi zone, Smoking zone, Touristic Information Office, Air Companies Offices, Shops (newspapers, souvenirs, etc.) and Customs.

- Services concept includes several subclasses corresponding to defined service categories in the airport problem. And services in each category are defined as a Product in the ontology (Map_Meteo, French News etc.).
- Profile concept represents the clients (one profile per participant) with their personal profiles and preferences: role (pilot, passenger, etc.), flight details, personal identification (name, agent role, passport number and nationality).

4 JADE/LEAP Implementation

We have implemented a multi-agent system that following the architecture of the multi-agent system, and using the particularization of the generic context-aware ontology described above.

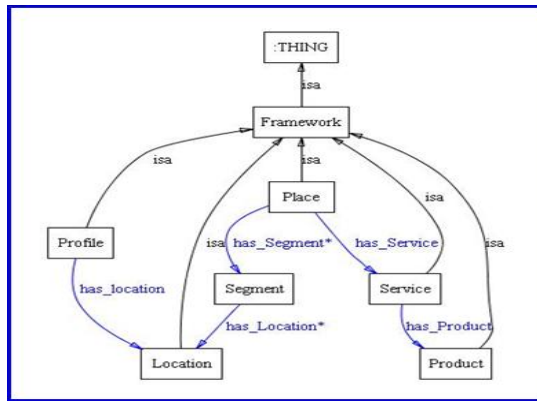


Fig. 4. High level concepts of the generic ontology

Although other alternatives exist, JADE-LEAP agent platform was chosen since it is a java-based and FIPA compliant agent platform, where agents communicate by sending FIPA ACL messages over a TCP/IP connection between different runtime environments on local servers or running on wireless devices. One local server will host the central agent that would rule over all the sensors of the airport (these sensors are just simulated inside the implementation of the central agent) that provide location information of clients, while one or more local servers host provider agents acting on behalf of airport services. Finally each portable client device runs a JADE-LEAP container that hosts one single agent that is used to provide a way to interact with the user (through a GUI), so the user can be informed and interact with the other agents running on different servers. JADE/LEAP platform logically connects the agents on the different servers with each other and with the corresponding client agents running on the users’ devices. First we assume an initial minimal profile known by the client LEAP agent: name, agent role, passport number, nationality and travel info (flight numbers, companies, origin and target). This is what we consider the public profile that Client agent includes in the content of the first message exchanged with the central agent.

Other additional data can be included in the profile as: smoker/nonsmoker, languages spoken, number of times that the passenger visited cities of current travel, professional activity, travel motivation, relationship with other passengers, etc. Some of these would be considered private information therefore this private information would be just internally used inside Client agent to provide customized filter of the services offered by provider agents. The user decides whether he introduces these confidential data or not. Finally our intention is to include reputation information linked to agents, and specific services inside client profiles, but this is not already done. Once client agent in LEAP, provider and central agents in JADE were implemented, we have chosen a specific case study to test such implementation in the airport application domain. We will describe how the system works by following all the messages exchanges when a pilot “Alex” arrives to the airport as we did in centralized approach. When Alex enters into the system’s wireless network, Alex client agent (running the LEAP code on the mobile device) asks for registration to the central agent (running JADE code in a PC acting as server). Registration includes identifying itself, so future location of pilot Alex can be assigned to a particular AID (agent identifier) in advance. Once Receive-registry-profile protocol has concluded, the central agent evaluates its position (through simulated movements) and computes the geographical proximity to the location of different services (platforms/PCs that host a particular JADE provider agent). Then, central agent notifies about the presence of such Alex client agent with ‘Notify agent’ protocol to the corresponding geographically close provider agents. When provider agent has been notified, it will offer its particular services customized according to the context exchanged by the central server (the public profile of the client agent). After that, negotiation included in offer-service protocol takes place in order to reach a possible agreement about services. For instance check-in provider agent would have received information about the flight of Alex, and according to this info the provider agent might offer Alex agent an automated check-in (asking Alex agent about the absence of baggage). Or alternatively check-in provider agent might inform the client about the place to carry out the manual check-in, jointly with other context-specific info such as the expected delay time to check-in, or the number of people in the row, etc. Other example of services such as news/gift shop agents may require richer profiles of clients that should include additional info as the language spoken, first time visiting a city to offer customized products (touristic guides, books and newspapers in mother tongue, etc...).

In any case, the given client agent has to internally filter out the interesting services from the received services based on the private profile of the user. This filter would avoid an overexposition of information to the user and would protect the privacy of the user.

5 Conclusions

We have presented an application of agent technology to an ambient intelligence problem in an airport domain. We adapted the problem we previously defined for this domain and we have already tested in a centralized system into a distributed, agent-based system. We implemented three types of agents: Central, Providers and Clients with JADE and LEAP with the intention to avoid bottle necks and to increase the

possibilities of customization through the use of enriched user profiles. We plan to include the exchange of recommendations from clients and reputation information about client and providers and to test the performance functionalities of both approaches in the next future.

References

1. Shadbolt, N.: Ambient Intelligence. *IEEE Intelligent Systems*, 2–3 (2003)
2. O'Hare, G.M.P., O'Grady, M.J., Kegan, S., O'Kane, D., Tynan, R., Marsh, D.: Intelligent Agile Agents: Active Enablers for Ambient Intelligence. In: ACM's Special Interest Group on Computer-Human Interaction (SIGCHI), Ambient Intelligence for Scientific Discovery (AISD) Workshop, Vienna, April 25 (2004)
3. Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P., Zipf, A.: CRUMPET: Creation of User-friendly Mobile Services Personalized for Tourism. In: Proceedings of the second International Conference on 3G Mobile Communication Technologies, London, UK (2001)
4. Afsarmanesh, H., Guevara Masís, V., Hertzberger, L.O.: Virtual Community Support In Telecare. In: Forth IFIP Working Conference on Virtual Enterprises, PRO-VE 2003, Lugano, Switzerland (October 2003)
5. Corchado, J.M., Bajo, J., Paz, Y., Tapia, D.I.: Intelligent environment for monitoring Alzheimer patients, agent technology for health care. *Decision Support Systems* (2007) doi:10.1016/j.dss.2007.04.008
6. Nieto-Carvajal, I., Botía, J.A., Ruiz, P.M., Gómez-Skarmeta, A.F.: Implementation and Evaluation of a Location-Aware Wireless Multi-agent System. In: Yang, L.T., Guo, M., Gao, G.R., Jha, N.K. (eds.) EUC 2004. LNCS, vol. 3207, pp. 528–537. Springer, Heidelberg (2004)
7. Wohltorf, J., Cissée, R., Rieger, A., Scheunemann, H.: An Agent-based Service-aware Framework for Ubiquitous Context-Aware Services. In: Kudenko, D., Kazakov, D., Alonso, E. (eds.) AAMAS 2004. LNCS (LNAI), vol. 3394. Springer, Heidelberg (2005)
8. Lech, T.C., Wienhofen, L.W.M.: AmbieAgents: A Scalable Infrastructure for Mobile and Context-Aware Information Services. In: The Second European Workshop on Multi-Agent Systems, Barcelona, Spain (2004)
9. Bajo, J., Julian, V., Corchado, J.M., Carrascosa, C., De Paz, Y., Botti, V., De Paz, J.F.: An Execution Time Planner for the ARTIS Agent Architecture. *Journal of Engineering Applications of Artificial Intelligence* 21(8) (January 2008) ISSN: 0925-1976
10. Pavón, J., Gómez Sanz, J.J., Fernández Caballero, A., Valencia, J.J.: Development of intelligent multi-sensor surveillance systems with agents. *Robotics and Autonomous Systems* 55(12), 892–903 (2007)
11. Sánchez-Pi, N., Fuentes, V., Carbo, J., Molina, J.M.: Knowledge-based System to define Context in Commercial Applications. In: 8th Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD), Qingdao (2007)
12. Fuentes, V., Carbo, J., Molina, J.M.: Heterogeneous Domain Ontology for Location Based Information System in a Multi-agent Framework. In: 7th Int. Conf. on Intelligent Data Engineering and Automated Learning, Burgos, Spain (2006)
13. Fuentes, V., Sánchez-Pi, N., Carbó, J., Molina, J.M.: Reputation in User Profiling for a Context-aware Multiagent System. In: 4th European Workshop on Multi-Agent Systems, Lisbon, Portugal (2006)

Design Patterns for Combining Social and Individual Intelligences on Modular-Based Agents

Bianca Innocenti¹, Beatriz López¹, and Joaquim Salvi²

¹ Control Engineering and Intelligent Systems Group

² Computer Vision and Robotics Research Group

Universitat de Girona, Campus Montilivi, edifici P4, 17071 Girona, Spain
{bianca.innocenti,beatriz.lopez,joaquim.salvi}@udg.edu

Abstract. Design patterns have been recently concerned in the multi-agent community for the design of systems with decentralized coordination. In this paper we present a design pattern for dealing with the complexity of developing a decentralized coordination multi-agent system for controlling a single robot. In our pattern, we combine different intelligences: an individual intelligence that enables agents to achieve their own goals, and a social intelligence that makes agents understand and manage with other agents in the community. The design pattern facilitates the implementation of modular-based agents inside the multi-agent architecture and its use helps developers when incorporating new agents in the architecture. The multi-agent architecture is used to control a Pioneer 2DX mobile robot.

Keywords: agent design pattern, multi-agent system, integrated intelligence, mobile robotics.

1 Introduction

A design pattern provides a reusable solution to a recurrent problem in a specific domain [1]. A pattern does not describe an actual design, but an abstract model of the solution using specific entities of the paradigm in use. Patterns make designs more flexible, elegant, and ultimately reusable. They help designers to build new solutions without having to start from scratch [2].

Recently, design patterns have concerned the multi-agent community [1, 3], to which our research have to do with. We have developed ARMADiCo, a multi-agent architecture for a single robot and with a distributed coordination approach to share the system resources [4]. As any other kind of robot architecture, different cognitive abilities are integrated in the multi-agent approach, each requiring different artificial intelligence techniques. However, being a distributed coordination mechanism, the global system behavior emerges from individual agents (micro level behaviors). For such kind of systems, design is still an open issue [5]. One of the current proposals consist in the use of agent patterns designs [6], and we have followed such approach in the design of ARMADiCo. The agents design pattern captures common features of the agents and facilitates the incorporation of agents in the architecture.

Particularly, in a decentralized coordination mechanism, each agent has to deal with, at least, two kinds of intelligences: individual and social. On one hand, individual intelligence enables an agent to achieve its assigned goals (as for example, planning a trajectory for achieving a target point). On the other hand, social intelligence enables an agent to understand and manage with other agents. Consistently, any developer that wants to incorporate a new agent into the architecture has to follow the same recurrent design: define the intelligence methods to deal with individual goals, and define the methods to deal with the global robot behavior.

In this paper we present how design patterns are used in our architecture, ARMADiCo, in order to organize the different intelligences required in the agents. We describe our general agent pattern design and several instantiation corresponding to different behavioral agents. Moreover, we show how the incorporation of new agents is simple by the use of these patterns.

This paper is organized as follows. First, in section 2 the design pattern is described, together with several highlights of the ARMADiCo agents. We continue by giving some details on the behavioral agents according to the pattern. Next, the experimental set up and results are described in sections 4 and 5 correspondingly. Some related work is summarized in section 6 and we end with some conclusions.

2 Design Pattern

The proposed multi-agent architecture, called ARMADiCo –Autonomous Robot Multi-agent Architecture with Distributed Coordination–, can be described according to the main components required in classical Hybrid Deliberative/Reactive Architectures [7]. First, an interface agent is defined to interact with humans or other external agents. Second, to reason about how to achieve high level goals, we propose the mission planning, the task planning, the path planning, the battery charger and the localization agents. Third, to deal with the environment, we implement what we called behavioral agents with the following goals: go to a point, avoid obstacles and go through narrow spaces. Fourth, to deal with the physical world (perception and actuators), an agent is designed for each available sensor (encoder, sonar, battery sensor) and a single actuator agent has been defined (robot), due to limitations of the hardware. Finally, there are also a set of back agents that deal with other functionalities required to give support to the overall multi-agent system (e.g. Directory Facilitator).

In order to design the agents, an agent pattern has been defined. It captures common features of the agents and facilitates the incorporation of agents in the architecture. Each component of the agent pattern is designed as a module. As a consequence, our agents follow a module-based approach inside the multi-agent architecture. The current pattern is shown in Table 1. Note that each agent is different, but the pattern design offers a way to capture the different components that an agent on the architecture must have. Thus, Table 2 shows the main differences among the agents, which corresponds to the particular instantiations of their goal and coordination components (i.e. their individual and social intelligence).

Table 1. Agent pattern design

Internal State:	Mechanism used by the agent in order to know about the progress of its goals, and to update the information of the environment.
Goal:	Goal configuration: Agent goals. Goal methods: Methods that implements agent goals
Competition:	List of possible conflicting agents due to resource sharing, and list of shared resources.
Collaboration:	List of agents from/to exchange messages (request, inform).
Coordination:	Utility Computation: Method (with the required parameters) used to compute the utility value for achieving a coordination agreement Resource exchange: Method used to exchange resources from one agent to another.
Helper methods:	All supporting methods that help the agent in registering in the system, communicating, starting up, etc. They are the same for all the agents.

Table 2. AI techniques for individual and social intelligence in ARMADiCo agents

	agent	individual (goal)	social (coordination)
behavioral	goto	fuzzy	fuzzy
	avoid	pid	fuzzy
	gothrough	pid	fuzzy
deliberative	mission planning	PRS	trajectory merging
	path planning	search	-
	localization	probabilistic MonteCarlo	-
	battery charger	model based	trajectory merging
perception	sonar	probabilistic	-
	encoder	mathematical	-
	battery sensor	model based	-
actuator	robot	-	-

Regarding individual intelligence, that is, the method employed by the agent in order to fulfill its goal is specified in the *goal* slot of the design pattern. Table 2 (column "individual") shows the techniques implemented in the current implementation of ARMADiCo.

On the other hand, social intelligence in an agent is related to the interaction with other agents to resolve the resource usage. When a resource is shared by more than one agent, a conflict can arise. In order to coordinate them, ARMADiCo uses a distributed coordination mechanism, in which the agents in conflict decide which is the winner agent that takes the resource control. No central arbiter decides upon the resource usage. Since robots concerns physically grounded resources, this coordination should take into account possible disruptions in the robot behavior. For this reason the coordination process is split into two different parts: the winner determination method (utility computation slot of the agent pattern) and resource exchange method. Regarding the former, it consists of a process to assign the shared resource to the agent with the highest utility. That is, all of the agents compute an utility function for the actions they require that represents the benefit the system will receive if it carries

Agent Pattern Design		Goto Agent
Internal State		Maintain motion progress information
Goal	Configuration	Drive the robot to the goal position with the desired heading
	Methods	Fuzzy Collaborative Control System
Competition		Avoid, gothrough agents for the robot agent
Collaboration		Encoder, mission planning, battery charger agents
Coordination	Utility Computation	Based on distance to goal position
	Resource Exchange	Fuzzy-based smoothing method
Helper Methods		-

Fig. 1. Pattern Design of Goto Agent

Agent Pattern Design		Gothrough Agent
Internal State		Maintain motion into narrow places progress information
Goal	Configuration	Detect narrow places and drive the robot through them
	Methods	Model based motion
Competition		Avoid and goto agents for the robot agent
Collaboration		Encoder, mission planning, battery charger and sonar agents
Coordination	Utility Computation	Based on distance to side obstacles
	Resource Exchange	Fuzzy-based smoothing method
Helper Methods		-

Fig. 2. Pattern Design of Gothrough Agent

out the proposed action from the point of view of the agent (so their utility functions measures a gain in wealth for the whole society). The utility function is defined in the interval $[0,1]$ for all the agents, being their values comparable. Concerning the latter, Table 2 (column "social") shows the resource exchange methods employed when the agent which wins the resource is different to the agent that has been currently using the resource up to now. Thus, robot behavior disruptions are avoided.

3 Behavioral Agents

In this section we illustrate the pattern design instances for the behavioral agents. As stated in section 2, there are three behavioral agents, the goto, the avoid and the gothrough agents. Fig. 1, 2 and 3 show the instances of our design pattern for the three behavioral agents. All of them have their individual intelligence methods (goal component of the agent pattern). The goto agent uses a fuzzy collaborative control system to move the robot to a target position, the avoid agent a PID control system to avoid obstacles and the gothrough agent a model based system to pass through narrow places.

Regarding coordination, all of them share the robot agent (resource), so conflicts could arise among them. Thus, they all have a utility computation method to determine who obtains the control over the conflicting resource (winner determination method). As stated above, the utility value varies in the interval $[0, 1]$, 1 being the maximum value. Therefore, the agent who is controlling the resource sends to the other conflicting agents its utility value.

Agent Pattern Design		Avoid Agent
Internal State		Maintain dodging obstacles progress information
Goal	Configuration	Avoid obstacles, guaranteeing save motion
	Methods	PID control system
Competition		Goto and gothrough agents for the robot
Collaboration		Sonar,encoder, mission planning, battery charger , robot agents
Coordination	Utility Computation	Based on time to collision
	Resource Exchange	Fuzzy-based smoothing method
Helper Methods		-

Fig. 3. Pattern Design of Avoid Agent

If this value is still the highest one, the agent will continue to control the resource. Otherwise, the agent who wins the resource obtains the opportunity to use the robot agent, but it should proceed on the resource usage taking into account its impact on the physical world in a similar manner than control fusion (resource exchange method). For doing so, we propose a fuzzy method based on the information used in the coordination process that is the same for all the behavioral agents (see [8] for additional details).

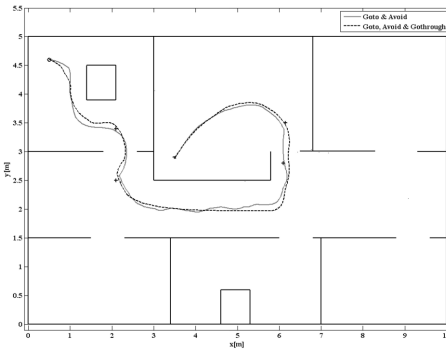
4 Experimental Set-Up

We have implemented ARMADiCo in C++ on Linux. The robot used for experimentation is a Pioneer 2DX of ActivMedia Robotics.

To test the use of the design pattern two ARMADiCo configurations have been set up, according to two different development phases:

- GA (Goto + Avoid): only the goto and the avoid agents are controlling the robot
- GAT (GA + GoThrough): the gothrough agent has been added to test the difficulty to add a new agent in the system as well as to verify that the emergent behavior of the whole system continues to be coherent and the desired one.

Next we present the defined scenario. The robot must go from Room A to Room B avoiding obstacles, as shown in Fig. 4-a).



a) Defined Scenario

Parameters	Scenario	
	GA	GAT
TD	11.64±0.14 m	11.84±0.10 m
FO	2.57±0.60°	1.56 ±0.90°
TT	71.84±6.85 s	60.17 ±3.45 s
P	99.37±0.12 %	99.31±0.21%
TG	68.64±3.14 %	99.31±0.21%

b) Results of the two tested situations

Fig. 4. Scenario to test and results of the emergent behavior of the robot

5 Results

Fig. 4-a) shows the trajectories described by the robot in grey when the GA configuration is used and in dotted when the gothrough is added (GAT). Maybe the most important issue is the fact that introducing the gothrough agent is quite simple since the agent pattern helps to determine the important aspects to consider: following it no modifications on the existing agents must be carried out.

Since our hypothesis is that the addition of the gothrough agent implies should improve the whole robot behavior, we need also to test how this happens. For this purpose, we have considered the following measures: *travelled distance (TD)*, the distance travelled by the robot to reach the goal; *final orientation (FO)*, the heading of the robot at the goal position; *total time (TT)*, the total amount of time the robot needs to achieve the goal; *precision (P)*, how closed to the goal position is the center of mass of the robot; and *time goto (TG)*, the total time the *goto* agent has the robot control.

Fig. 4-b) shows the average and standard deviation of each evaluation measure after five runnings. Comparing the results, we can see that they are very similar, except for the total time (TT) needed to arrive at the destination. With the gothrough agent, this time is decreased, meaning that the average speed is higher than when there is only the goto and the avoid agents.

6 Related Work

The application of multi-agent system to robotics has been mainly concerned to multiple robot system. For example, in [9] several soccer robots coordinate their activities based on case-based retrieval. Regarding the development of multi-agent architectures for a single robot, there are fewer works (see, for example, [10] and [11]). The main difference with our architecture is that we follow a distributed coordination approach, so there is no central arbiter solving conflicts in the use of shared resources. As a consequence, we are following an emergence approach: the global system behavior emerge (macro level) from individual agents (micro level behaviors).

Regarding design issues, centralized coordination approaches use to follow a top-down traditional methodology. For distributed coordination multi-agent development, design is still an open issue. The use of design pattern has recently concerning the multi-agent community to deal with engineering emergence in decentralized autonomic system, like ours. For example, in [3] an agent pattern is proposed to encapsulate a business specific class in an AgentSpace framework and other web-based environments. In [1] the authors propose the use of agent patterns combined with workflows as a methodology for developing such emergence systems. Our design pattern is simpler than the one proposed in [1], but accomplishes the design problems we have: to state clearly the requirements of each agent, that is, an individual intelligence and a social intelligence methods. However, we should contemplate the inclusion of workflows in a future work.

Regarding the use of design patterns in robotics, there are several previous works out of the scope of the multi-agent paradigm. For example, [12] three behavioral-based are proposed to deal with three different ways of dealing with human interaction in robot control: traded, shared and supervised. In traded control, the responsibilities for producing behavior are traded between man and machine; in shared

control, an operator is guiding the robot to target, and in supervisory control the controller performs the entire task. In [13], a single design pattern is proposed to deal with the complexity of developing a robot control, according to three main layers: strategic, tactical and execution. All of these previous approaches focus on the development of a single system for controlling a robot (mainly based on object oriented methodologies) and agent patterns helps in the complex process of dealing either with real-time complexity or with human-robot interaction [14]. However, our proposal focuses on the development of a control architecture composed by a collection of agents (or autonomous systems). So although internally each agent follows a modular-architecture based on object oriented programming as well, the pattern focuses on the cooperation of the different agents in the architecture while maintaining its individual goals. Table 3 shows a comparative view of all the approaches.

Table 3. Different design patterns for robotics

Authors	Focus	Pattern(s)	Number patterns
Graves and Czarnecki [12]	human-robot interaction	traded, shared, supervised	3
Nelson [13]	real-time	Strategic+tactical+ execution	1
This paper	agent cooperation	individual+social	1

7 Conclusions and Discussion

Agents in a multi-agent architecture with distributed coordination are complex; need to deal with, at least, two kinds of intelligence (individual and social). In addition, the design of such agents is a recurrent process in which the same pattern is repeated, considering these two intelligence and design patterns offers a tool to facilitate this process.

In this paper we have described the design pattern we have employed to define the agents of the ARMADiCo robot architecture. This pattern explicitly describes the combination of different kinds of intelligences (and so techniques) at the agent level. Thus, techniques as fuzzy logic, search are used to fulfill the individual intelligence of the agent; while utility computation or fuzzy logic are used to deploy the agent social abilities. The design pattern is then implemented as a module-based architecture that conforms the agent, which in turn interacts to other similar agents in the multi-agent architecture. We have experimentally shown how the use of design patterns facilitates the inclusion of new agents in the architecture, when applying it to control a mobile robot.

As a future work, we are thinking on dealing with other methodological issues, as the ones proposed in [15], in order to deal with the emergence of the overall ARMADiCo architecture.

Acknowledgments. This work was partially supported by the Spanish MEC Project DPI2006-09370 and by the DURSI AGAUR SGR 00296: Automation Engineering and Distributed Systems Group (AEDS).

References

1. Gardelli, L., Viroli, M., Omicini, A.: Design patterns for self-organizing multiagent systems. In: Proceedings of EEDAS (2007)
2. Gamma, E., Helm, R., Johnson, R., Vlissides, J.M.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, Reading
3. Silva, A., Delgado, J.: The agent pattern: A design pattern for dynamic and distributed applications. In: Proceedings of EuroPLoP 1998 (1998)
4. Innocenti, B., López, B., Salvi, J.: A multi-agent architecture with cooperative fuzzy control for a mobile robot. *Robotics and Autonomous Systems* 55, 881–891 (2007)
5. De Wolf, T., Holvoet, T.: Towards a methodology for engineering self-organizing emergent systems. *Self-Organization and Autonomic Informatics (I), Frontiers in Artificial Intelligence and Applications* 135, 52–61 (2007)
6. Tahara, Y., Ohsuga, A., Honiden, S.: Agent system development method based on agent patterns. In: Proceedings of the 21st ICSE 1999, pp. 356–367 (1999)
7. Murphy, R.R.: Introduction to AI Robotics. MIT Press, Cambridge (2000)
8. Innocenti, B., López, B., Salvi, J.: Resource coordination deployment for physical agents. In: *From Agent Theory to Agent Implementation*, 6th Int. Workshop AAMAS (2008)
9. Ros Espinosa, R., Veloso, M.: Executing multi-robot cases through a single coordinator. In: Proceedings of AAMAS 2007 (2007)
10. Neves, M.C., Oliveira, E.: A multi-agent approach for a mobile robot control system. In: Proceedings of MASTA 1997 - EPPIA 1997, pp. 1–14 (1997)
11. Busquets, D., Sierra, López de Mántaras, R.: A multiagent approach to qualitative landmark-based navigation. *Autonomous Robots* 15, 129–154 (2003)
12. Graves, A., Czarnecki, C.: Design patterns for behavior-based robotics. *IEEE Trans. on Systems, Man & Cybernetics, Part A (Systems & Humans)* 30(1), 36–41 (2000)
13. Nelson, M.L.: A design pattern for autonomous vehicle software control architectures. In: Proceedings of 23rd COMPSAC, pp. 172–177 (1999)
14. Zalewski, J.: Real-time software design patterns. In: 9th Conf. on Real-Time Systems, Ulstron, Poland (2002), <http://citeseer.ist.psu.edu/zalewski02realtime.html>
15. De Wolf, T., Holvoet, T.: Using UML 2 activity diagrams to design information flows and feedback-loops in self-organizing emergent systems. In: Proceedings of EEDAS, pp. 52–61 (2007)

Experiments in Multi Agent Learning

Maria Cruz Gaya and J. Ignacio Giraldez

Universidad Europea de Madrid, C/Tajo s/n
28690 Villaviciosa de Odón, Madrid, Spain
{mcruz, ignacio.giraldez}@uem.es

Abstract. Data sources are often dispersed geographically in real life applications. Finding a knowledge model may require to join all the data sources and to run a machine learning algorithm on the joint set. We present an alternative based on a Multi Agent System (MAS): an agent mines one data source in order to extract a local theory (knowledge model) and then merges it with the previous MAS theory using a knowledge fusion technique. This way, we obtain a global theory that summarizes the distributed knowledge without spending resources and time in joining data sources. The results show that, as a result of knowledge fusion, the accuracy of initial theories is improved as well as the accuracy of the monolithic solution.

Keywords: distributed data mining, ensemble techniques, MAS, evolutionary computation.

1 Introduction

There are a lot of real problems where data are dispersed geographically, for example, the client data of different branches of a bank or the medical records from different hospitals. If we want to learn about a behavior pattern of the bank clients or about the symptoms of a disease, we need to take into account all the data sources. One option is to join all the data sources and run a learning algorithm over it. Another option is trying to obtain the knowledge model in a decentralized way. The first option is not always possible, or demands a lot of resources. The second one requires a distributed framework for mining the distributed data sources (in order to obtain local theories). And then, it requires merging the local theories into a single global theory.

Multi Agent Systems (MAS) have been used for distributed data mining successfully. For instance, in [1] MAS are applied to the credit card fraud problem, in [2] MAS are applied to carcinogenesis prediction and, recently, they have also been applied to artificial vision [3].

The merging process that obtains a single global theory from local theories can be considered a meta-learning process. Prodromidis et. al [4] define meta-learning as the technique that computes high level classifiers, called meta-classifiers, which integrate the base classifiers computed over independent and geographically disperse data sources, in some way. There are different well known techniques for meta-learning. [5] presents a study about different Ensembles of Classifiers (EoC) and the reasons for its base classifier accuracy improvement. These methods learn which base

classifier to use for each sample. The training set of the meta-classifier is composed by the base classifiers predictions. The output is a new theory that can provide the final prediction given the base classifiers predictions for a new instance of the problem. In [1,2] the EoC are demonstrated to improve the theory accuracy. Recently, in PAKDD 07 [6] the EoC have been the classifiers better positioned.

Evolutionary computation has been used for combining classifiers. In [11], an evolutionary algorithm is used for combining several classifiers. The members of the population are trees that can decide what initial classifier to apply in each case. In [12], an evolutionary algorithm is used for selecting the base classifier set that will be combined.

In this work, we present a MAS framework with an evolutionary approach to merge theories, in order to obtain in a decentralized way, a global theory more accurate than the initial ones.

In section 2 the framework based on MAS (called Theory Synthesis Evolution: TheSynEv) is described, including how the evolutionary approach has been used to the fusion of theories; in section 4 we present experiments and the results obtained. The paper ends with the conclusions presented in section 5.

2 TheSynEv System

This section describes the TheSynEv System: a MAS that mines several geographically dispersed data sources in order to obtain, in a decentralized way, a global theory that summarizes all the knowledge using an evolutionary approach.

2.1 MAS Framework

Each agent of the MAS can operate in two modes: learning or meta-learning. When an agent is in learning mode it learns from its data source and extracts the local theory that represents its knowledge model of the domain problem (fig. 1). Each agent does the same in a parallel way using a traditional learning algorithm (C4.5 [6]).

When the system operates in meta-learning mode it produces a global theory. Each agent asks for the global theory, waits for it, and executes a fusion process in order to add his local knowledge, resumed in his local theory, to the global system knowledge (global theory). This is done in a sequential mode: only one agent can merge its local knowledge with the global knowledge at a time. The fig.2. outlines TheSynEv in meta-learning mode.

Internally the system can be represented according to fig.3. In each site there is an agent capable of:

1. Learning a local theory from the local training data set. This local theory may be used to classify any problem instance. This local theory also represents the domain map estimated from the evidence of local data. A traditional algorithm C4.5 is used in order to obtain the local theory.
2. Adding its knowledge to the global system knowledge. The agent receives in a message the system global theory and modifies it adding the knowledge extracted from its local data source. The global theory is incrementally synthesized from the individual local theories provided by the remote agents.

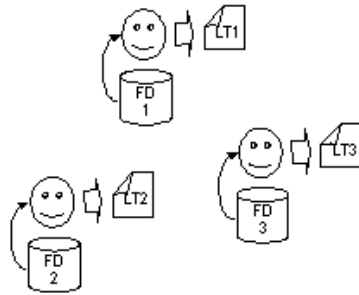


Fig. 1. System agents in learning mode. Each agent extracts his local theory (LT) using its data source (FD) as input.

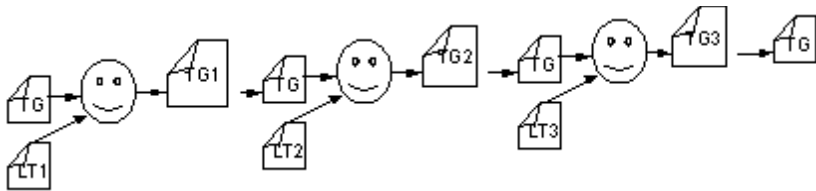


Fig. 2. System agents in meta-learning mode. Each agent fuses the local theory (LT_i) and the global theory (TG) and obtains the new system global theory (TG_i).

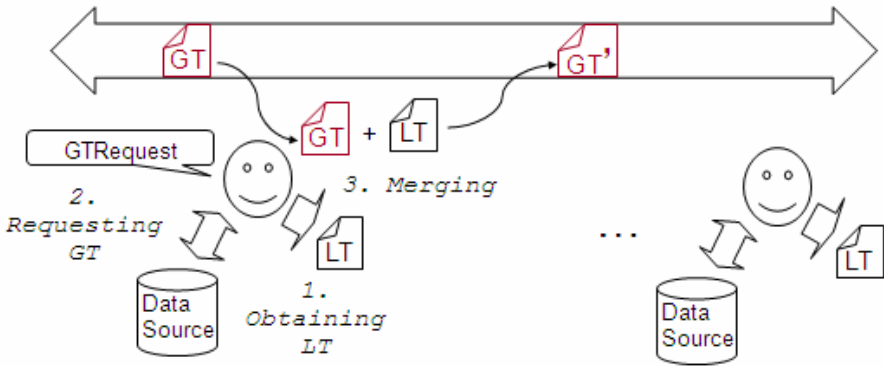


Fig. 3. Each agent extracts the local theory from its data source, asks for the global theory through a GTRequest message, fuses the local theory (LT) and the global theory (TG) modifying this. It sets the modified global Theory (TG') as the new system global theory.

The global theory improves the EoC operation in two aspects: (i) making a prediction without the participation of the base classifiers, (ii) offering an explanation in terms of the domain and (iii) solving knowledge contradictions.

2.2 The Evolutionary Approach

We use an evolutionary algorithm in order to merge theories. Each individual represents an theory. The genetic operators used are: mutation and crossover. The process is divided into two phases: growth and evolution. The first phase has as input the global and local theories, several genetic operators are applied on these theories in order to obtain the initial population to be used in the evolution phase. The evolution phase transforms the initial population into a final population whose best individual will be the resulting global theory.

Each individual is not represented as a bit string, but as several rule lists (one per class label). Each rule represents a region of the domain map.

The mutation operator selects a rule randomly and changes its class label. The crossover operator selects an attribute randomly and computes the domain map for this attribute. The domain map is a partition according to this attribute's values. Figure 4 outlines the operation of the crossover operator for a bi-dimensional problem and an attribute A_k with two values.

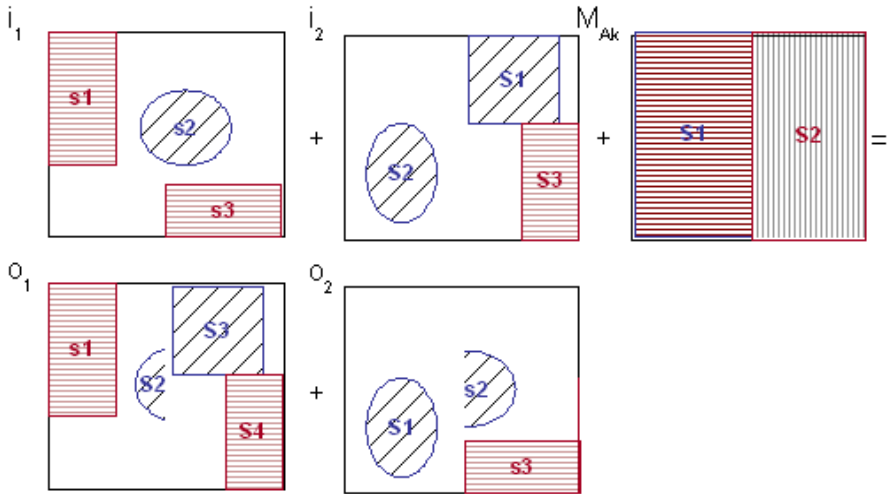


Fig. 4. Representation of the crossover between two individual i_1 and i_2 using the attribute A_k for the partition. The results are the o_1 and o_2 offsprings.

In the growth phase the genetic operators are applied over the global and local theories until the desired population size is reached. The size of the population is a program parameter. Each individual has a weight that represents its importance. The selection operator makes a probabilistic selection (individuals with large weight values are more likely to be selected for the next generation). In the growth phase, the weight to be assigned to the local theory depends on how much the global theory was modified by the local theory.

For the evolution phase, an evolutionary algorithm [7] is used with the following parameters: i) the probability for applying a mutation operator, ii) the probability for applying a crossover operator, iii) the size of the population and iv) the number of

generations to be generated. The output of the fusion process will be the individual of the last generation with highest fitness value, i.e., highest accuracy. The accuracy is measured over a test dataset.

3 Experiments

The experiments have been done over twelve datasets of the UCI Machine Learning Repository [8]. Table 1 shows the characteristics of these problems.

The WEKA application [9] was used to preprocess the datasets by the following way:

1. the unsupervised attribute filter *Discretize* was used in order to discretize the range of numeric attributes in the dataset into nominal attributes using ten binds.
2. the unsupervised instance filter *Randomize* was used in order to randomly shuffle the instances.
3. attributes have been selected using information gain as attribute evaluator and ranker as search method. The ten best attributes have been selected using the attribute unsupervised filter *Remove*.
4. the dataset was partitioned into 3 sets: (i) training set for local theory with 40% of the samples, (ii) training set for the global theory with 40% of the samples and (iii) test set for computing the accuracy with 20% of the samples.

Results are presented in table 2. Several executions have been done for each dataset varying the parameters of the program (mutation probability and crossover probability). All of them use the same values for size population and number of generations (10 and 100 respectively).

Our goal is to determine whether the accuracy of the resulting theory improves the accuracy of the initial theories (local and global) and the accuracy of the monolithic solution. In order to compute the accuracy of the monolithic solution, the instances are joined in the two initial dataset portions for generating the monolithic theory.

Table 1. Characteristics of the problems used in experiments. The columns are the name of the database (*Name*), the number of attributes including the class attribute (*Attributes*) and the number of the instances in the database (*instances*).

Name	Attributes	Instances
Hepatitis	20	155
Balloon	5	76
Echocardiogram	13	132
Cylinder bands	40	512
Agaricus-lepiota	23	8124
1985 Auto Imports.	26	205
BUPA liver disorders	7	345
Respiratory	18	85
Ionosphere	35	351
Sonar	61	208
Weaning	18	302
Haberman	4	306

Table 2. Experiments. The columns are: the name of the dataset (*name*), the mutation probability (*m*), the crossover probability (*C*), the accuracy of the initial local theory (*TL acc*), the initial global theory accuracy (*TG acc*), the accuracy of the new global theory resulting from the fusion process (*TG' acc*) and the accuracy of the monolithic solution (*monolithic acc*). This table shows in bold the experiments in which the accuracy of the global theory resulting from the fusion process improves both, the initial theories accuracy and the monolithic accuracy. The best execution for each dataset is underlined.

Name	m	C	TL acc	TG acc	TG' acc	Monolithic acc
hepatitis10discretize	0.0	0.5	0.0	0.64	0.64	0.84
<u>hepatitis10discretize</u>	<u>0.1</u>	<u>0.4</u>	<u>0.0</u>	<u>0.64</u>	<u>0.84</u>	<u>0.84</u>
<u>hepatitis10discretize</u>	<u>0.2</u>	<u>0.3</u>	<u>0.0</u>	<u>0.64</u>	<u>0.84</u>	<u>0.84</u>
<u>hepatitis10discretize</u>	<u>0.3</u>	<u>0.2</u>	<u>0.0</u>	<u>0.64</u>	<u>0.84</u>	<u>0.84</u>
<u>hepatitis10discretize</u>	<u>0.4</u>	<u>0.1</u>	<u>0.0</u>	<u>0.64</u>	<u>0.84</u>	<u>0.84</u>
hepatitis10discretize	0.5	0.0	0.0	0.64	0.8	0.84
ballons	0.0	0.5	0.7	0.6	0.7	0.7
<u>ballons</u>	<u>0.1</u>	<u>0.4</u>	<u>0.7</u>	<u>0.6</u>	<u>0.8</u>	<u>0.7</u>
<u>ballons</u>	<u>0.2</u>	<u>0.3</u>	<u>0.7</u>	<u>0.6</u>	<u>0.8</u>	<u>0.7</u>
<u>ballons</u>	<u>0.3</u>	<u>0.2</u>	<u>0.7</u>	<u>0.6</u>	<u>0.8</u>	<u>0.7</u>
<u>ballons</u>	<u>0.4</u>	<u>0.1</u>	<u>0.7</u>	<u>0.6</u>	<u>0.8</u>	<u>0.7</u>
ballons	0.5	0.0	0.7	0.6	0.7	0.7
<u>echocardiogramdiscretize</u>	<u>0.0</u>	<u>0.5</u>	<u>0.56</u>	<u>0.76</u>	<u>1.0</u>	<u>0.56</u>
<u>echocardiogramdiscretize</u>	<u>0.1</u>	<u>0.4</u>	<u>0.56</u>	<u>0.76</u>	<u>1.0</u>	<u>0.56</u>
<u>echocardiogramdiscretize</u>	<u>0.2</u>	<u>0.3</u>	<u>0.56</u>	<u>0.76</u>	<u>1.0</u>	<u>0.56</u>
<u>echocardiogramdiscretize</u>	<u>0.3</u>	<u>0.2</u>	<u>0.56</u>	<u>0.76</u>	<u>1.0</u>	<u>0.56</u>
<u>echocardiogramdiscretize</u>	<u>0.4</u>	<u>0.1</u>	<u>0.56</u>	<u>0.76</u>	<u>1.0</u>	<u>0.56</u>
echocardiogramdiscretize	0.5	0.0	0.56	0.76	0.76	0.56
<u>bands10discretize</u>	<u>0.0</u>	<u>0.5</u>	<u>0.766</u>	<u>0.95</u>	<u>1.0</u>	<u>0.883</u>
<u>bands10discretize</u>	<u>0.1</u>	<u>0.4</u>	<u>0.766</u>	<u>0.95</u>	<u>1.0</u>	<u>0.883</u>
<u>bands10discretize</u>	<u>0.2</u>	<u>0.3</u>	<u>0.766</u>	<u>0.95</u>	<u>1.0</u>	<u>0.883</u>
<u>bands10discretize</u>	<u>0.3</u>	<u>0.2</u>	<u>0.766</u>	<u>0.95</u>	<u>1.0</u>	<u>0.883</u>
<u>bands10discretize</u>	<u>0.4</u>	<u>0.1</u>	<u>0.766</u>	<u>0.95</u>	<u>1.0</u>	<u>0.883</u>
<u>bands10discretize</u>	<u>0.5</u>	<u>0.0</u>	<u>0.766</u>	<u>0.95</u>	<u>1.0</u>	<u>0.883</u>
agaricus-lepiota.10.rand	0.0	0.5	0.965	0.990	0.990	0.995
<u>agaricus-lepiota.10.rand</u>	<u>0.1</u>	<u>0.4</u>	<u>0.965</u>	<u>0.990</u>	<u>1.0</u>	<u>0.995</u>
<u>agaricus-lepiota.10.rand</u>	<u>0.2</u>	<u>0.3</u>	<u>0.965</u>	<u>0.990</u>	<u>1.0</u>	<u>0.995</u>
<u>agaricus-lepiota.10.rand</u>	<u>0.3</u>	<u>0.2</u>	<u>0.965</u>	<u>0.990</u>	<u>1.0</u>	<u>0.995</u>
<u>agaricus-lepiota.10.rand</u>	<u>0.4</u>	<u>0.1</u>	<u>0.965</u>	<u>0.990</u>	<u>1.0</u>	<u>0.995</u>
agaricus-lepiota.10.rand	0.5	0.0	0.965	0.990	0.990	0.995
<u>imports</u>	<u>0.0</u>	<u>0.5</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>
<u>imports</u>	<u>0.1</u>	<u>0.4</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>
<u>imports</u>	<u>0.2</u>	<u>0.3</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>
<u>imports</u>	<u>0.3</u>	<u>0.2</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>

Table 2. (continued)

<u>mports</u>	<u>0.4</u>	<u>0.1</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>
<u>imports</u>	<u>0.5</u>	<u>0.0</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>
bupadiscretizerand	0.0	0.5	0.652	0.0	0.652	0.579
bupadiscretizerand	0.1	0.4	0.652	0.0	0.753	0.579
bupadiscretizerand	0.2	0.3	0.652	0.0	0.753	0.579
bupadiscretizerand	0.3	0.2	0.652	0.0	0.753	0.579
<u>bupadiscretizerand</u>	<u>0.4</u>	<u>0.1</u>	<u>0.652</u>	<u>0.0</u>	<u>0.826</u>	<u>0.579</u>
bupadiscretizerand	0.5	0.0	0.652	0.0	0.666	0.579
rdsdiscretizerand10	0.0	0.5	0.882	0.941	0.941	0.941
<u>rdsdiscretizerand10</u>	<u>0.1</u>	<u>0.4</u>	<u>0.882</u>	<u>0.941</u>	<u>1.0</u>	<u>0.941</u>
<u>rdsdiscretizerand10</u>	<u>0.2</u>	<u>0.3</u>	<u>0.882</u>	<u>0.941</u>	<u>1.0</u>	<u>0.941</u>
<u>rdsdiscretizerand10</u>	<u>0.3</u>	<u>0.2</u>	<u>0.882</u>	<u>0.941</u>	<u>1.0</u>	<u>0.941</u>
<u>rdsdiscretizerand10</u>	<u>0.4</u>	<u>0.1</u>	<u>0.882</u>	<u>0.941</u>	<u>1.0</u>	<u>0.941</u>
rdsdiscretizerand10	0.5	0.0	0.882	0.941	0.941	0.941
ionospherediscretizerand10	0.0	0.5	0.9	0.814	0.914	0.928
<u>ionospherediscretizerand10</u>	<u>0.1</u>	<u>0.4</u>	<u>0.9</u>	<u>0.814</u>	<u>0.928</u>	<u>0.928</u>
<u>ionospherediscretizerand10</u>	<u>0.2</u>	<u>0.3</u>	<u>0.9</u>	<u>0.814</u>	<u>0.928</u>	<u>0.928</u>
<u>ionospherediscretizerand10</u>	<u>0.3</u>	<u>0.2</u>	<u>0.9</u>	<u>0.814</u>	<u>0.928</u>	<u>0.928</u>
<u>ionospherediscretizerand10</u>	<u>0.4</u>	<u>0.1</u>	<u>0.9</u>	<u>0.814</u>	<u>0.928</u>	<u>0.928</u>
ionospherediscretizerand10	0.5	0.0	0.9	0.814	0.9	0.928
sonardiscretizerand10	0.0	0.5	0.690	0.738	0.881	0.714
sonardiscretizerand10	0.1	0.4	0.690	0.738	1.0	0.714
sonardiscretizerand10	0.0	0.5	0.690	0.738	0.881	0.714
<u>sonardiscretizerand10</u>	<u>0.1</u>	<u>0.4</u>	<u>0.690</u>	<u>0.738</u>	<u>1.0</u>	<u>0.714</u>
<u>sonardiscretizerand10</u>	<u>0.2</u>	<u>0.3</u>	<u>0.690</u>	<u>0.738</u>	<u>1.0</u>	<u>0.714</u>
sonardiscretizerand10	0.3	0.2	0.690	0.738	0.976	0.714
<u>sonardiscretizerand10</u>	<u>0.4</u>	<u>0.1</u>	<u>0.690</u>	<u>0.738</u>	<u>1.0</u>	<u>0.714</u>
sonardiscretizerand10	0.5	0.0	0.690	0.738	0.857	0.714
weaningdiscretizerand10	0.0	0.5	0.766	0.733	0.783	0.766
weaningdiscretizerand10	0.1	0.4	0.766	0.733	0.8	0.766
weaningdiscretizerand10	0.2	0.3	0.766	0.733	0.783	0.766
weaningdiscretizerand10	0.3	0.2	0.766	0.733	0.8	0.766
<u>weaningdiscretizerand10</u>	<u>0.4</u>	<u>0.1</u>	<u>0.766</u>	<u>0.733</u>	<u>0.9</u>	<u>0.766</u>
weaningdiscretizerand10	0.5	0.0	0.766	0.733	0.733	0.766
habermandiscretize	0.0	0.5	0.766	0.0	0.766	0.766
habermandiscretize	0.1	0.4	0.766	0.0	0.866	0.766
<u>habermandiscretize</u>	<u>0.2</u>	<u>0.3</u>	<u>0.766</u>	<u>0.0</u>	<u>0.9</u>	<u>0.766</u>
habermandiscretize	0.3	0.2	0.766	0.0	0.866	0.766
<u>habermandiscretize</u>	<u>0.4</u>	<u>0.1</u>	<u>0.766</u>	<u>0.0</u>	<u>0.9</u>	<u>0.766</u>
habermandiscretize	0.5	0.0	0.766	0.0	0.8	0.766

The experiments show that:

- In 9 out of 12 datasets the accuracy of the initial theories and of the monolithic solution are improved. The other datasets where this does not happen are hepatitis10discretize and ionosphere, where the accuracy improves that of the initial theories but equals the accuracy of the monolithic solution. In imports dataset the accuracy cannot be improved because it has the highest value. Thus, the improvements to the accuracies of the local theories, and to the accuracy of the monolithic solution, are demonstrated.
- The best values of the accuracy for each domain is reached in 0.0/0.5 execution in 3 times, in 0.1/0.4 execution in 9 times, in 0.2/0.3 in ten times, 0.3/0.2 in 8 times, 0.4/0.1 in 12 times and in 0.5/0.0 in 2 times. So both operators are necessary.

4 Conclusion

This paper shows a method for merging theories using an evolutionary approach. The advantages over other meta-learning approaches are (i) the explanation in terms of the domain attributes, (ii) that base classifiers are not required to classify a new instance of the problem and (iii) the knowledge contradictions are solved.

The accuracies obtained in most datasets improved those of the initial theories (local and global theories) and also the accuracy of the monolithic solution. In the cases in which the improvement isn't achieved, at least the accuracy of the monolithic solution is equaled.

References

1. Giráldez, J.I.: Modelo de toma de decisiones y aprendizaje en sistemas multiagente, Tesis para el grado de doctor en Informática, Universidad Politécnica de Madrid (1999)
2. Stolfo, S., Prodromidis, A.L., Tselepis, S., Lee, W., Fan, W., Chan, P.: JAM: Java Agents for meta-learning over distributed databases. In: Third International Conference in Knowledge Discovery and Data Mining (KDD 1997). Newport Beach, California (1997)
3. Barandela, R., Sánchez, J.S., Valdovinos, R.M.: New applications of ensembles of classifiers. *Pattern Analysis and Applications* 6(3), 245–256 (2003)
4. Prodromidis, A.L., Stolfo, S.J., Chan, P.: *Advances of Distributed Data Mining*. Kargupta, H., Chan, P. (eds.). AAAI press, Menlo Park (2000)
5. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS 1857, pp. 1–15. Springer, Heidelberg (2000)
6. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco (1993)
7. Koza, J.R., et al.: *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers, Dordrecht (2003)
8. <http://www.ics.uci.edu/~mlearn/MLRepository.html>
9. Witten, I.H., Frank, E.: *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers Inc., San Francisco (2000)

Agent-Based Simulation of Business Processes in a Virtual World

Branislav Bošanský¹ and Cyril Brom²

¹ Center of Biomedical Informatics, Institute of Computer Science
Academy of Sciences of the Czech Republic
bosansky@euromise.cz

² Department of Software and Computer Science Education, Faculty of Mathematics and
Physics, Charles University in Prague
brom@ksvi.mff.cuni.cz

Abstract. Business process modeling (BPM) has proven itself as a useful technique for capturing the work practice in companies. In this paper, we focus on its usage in the domain of company simulation and we present a novel approach combining the clarity of BPM with the strength of agent-based simulations. We describe the enhancement of a general process modeling language, the algorithm transforming these enhanced processes into the definition of agents' behavior, and the architecture of the target multi-agent system simulating the modeled company. The example is given as the implemented prototype of all proposed methods leading towards the simulation of a virtual company.

1 Introduction

Business process modeling (BPM) is a widely used technique offering a simple and understandable view on the work practice and it is mainly utilized by managers and executives. However, it has limited usage and can be unsuitable in the domain of company simulation, usually based on a statistical calculation, which cannot sufficiently represent human behavior or social factors. On the other hand, as proven by many existing applications, multi-agent systems and agent-based simulations can at the cost of typically more complicated specification provide more realistic model.

In our research we intend to create an agent-based simulation of the work practice in a company that is defined using a BPM technique. Our goal is to reach following characteristics of a company simulation program:

- A user can define a work assignment for agents (e.g. virtual subordinates) through some process modeling visual tool.
- Employees, represented by agents, should be human-like and they should act in a virtual world in order to accomplish given tasks and predefined goals.
- A user can see the virtual world, he/she can see the visual interpretation of agents representing employees, and can interact with this world.

The motivation behind described goals is to allow an easy modification of an agent-based simulation by a non-specialized user. An example can be given as an educational program for students of economy fields, where they can define what employees

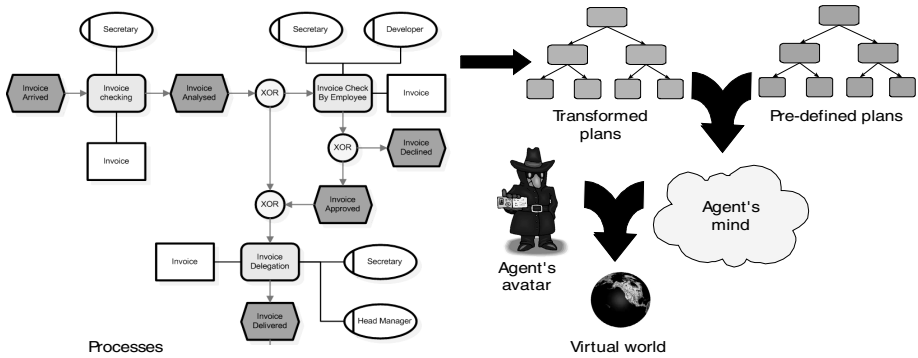


Fig. 1. The ground idea of presented approach

should do and how the employees should cooperate, and the program simulates and shows the progress of the work in the virtual world that represents the company.

In order to accomplish our objectives, we enhance a process language that can be automatically transformed into behavior description of agents that act in a multi-agent system (MAS), which represents the target simulation. The main aim of this paper is to present the algorithm transforming these enhanced processes into the hierarchical reactive plans for agents as well as to describe the architecture of a multi-agent system that can simulate virtual employees in a virtual world. Note that we understand these enhanced processes to be a parameterization for the target MAS and we assume other mandatory aspects for the agent-based simulation, such as the description of a virtual world, *a priori* knowledge of agents and their basic behavior (e.g. picking up an object), to be already captured (see Figure 1).

This paper is organized as follows. In Section 2 we analyze the problem domain and discuss different approaches in the area of company simulation. Section 3 is devoted to the description of a process modeling language enhancement. In Section 4 we describe the functioning of the target multi-agent system together with the algorithm transforming these enhanced processes into hierarchical reactive plans. In Section 5 we briefly present implemented prototype and the case study and we give our conclusions and discuss the possibilities of the future work in Section 6.

2 Business Process Simulation

We understand a business process to be an activity relevant for adding a value to an organization and capturing these activities as the business process modeling (BPM). There are several different languages that can be used for BPM, such as Unified Modeling Language, Business Process Modeling Notation or Event-Driven Process Chains (EPC) [5]. They are all related to some visualization and as they are very similar to each other, we will further talk about a general process language that represents a set of sequences of activities (processes), and that can be split into several flows or joined back together.

Usual process simulation approaches based on BPM methods are based on statistical calculation (e.g. as in [9]). However, several problems can be identified while using this method. As shown in [10], the concept of *work* itself consists of more than a functional transformation of work products by people and there are a lot of influences that cannot be captured using any business process model (e.g. effects of collaboration of employees or their communication). Moreover, employees are considered to be only the resources and it is hard to simulate their specific aspects (e.g. experience level, cultural or social factors) [10]. Finally, this method has only limited capabilities of visual presentation of running simulation and we do not actually see the employees working.

2.1 Agent-Based Company Simulations and Related Work

Agent-based simulations and their usage in a simulation of a company can bring several crucial advantages in the course of the simulation as such [4, 6, 8, 10], and can overcome some of the problems identified in the previous section. Firstly, agents that represent employees are more accordant with people and issues like communication, cooperation or coordination are all the basic characteristics of a multi-agent system. Agents can also be specialized (e.g. in life experience or adaptability in a new environment), agents are able to plan the assigned work or assign work to other agents, and modeling of interruptions or human behavior (e.g. personal characteristics, basic needs) is a lot easier too. Finally, in an agent-based simulation, which is set in a virtual environment, possible non-modeled behaviors can emerge (e.g. an agent carrying paper can be affected by other agents that are blocking the way).

However, the standard specification of an agent-based simulation is usually being done by defining agents' duties separately (e.g. using rules or finite states machines) in a correct way in order to achieve a global goal. Such an approach puts higher demands on a user and it is in contradiction to our requirements.

Hence we need to analyze possibilities of combining the BPM with agents, which is, according to our research, a not well documented approach and it is, to our best knowledge, not covered by any theory that would interpret a relationship between a general process-like language and an agents' architecture from the artificial intelligence point of view. We found several practical solutions in the domain of process modeling [4], in workflow systems [2, 11], or in biomedical informatics [3]. Their common characteristic is the attempt to formalize actions on a very low level (i.e. the atomic actions are rather basic, such as checking specific fields on an invoice) in order to be automatically executed by agents. Also, more advanced issues like the concept of trust, or different perspective of agents on the execution of their joint activity [4, 11] are addressed, which inhibits the development of the general transforming algorithm.

Approach in this paper presents several crucial distinctions. The first evolves in modeling actions on a higher level (i.e. handling an invoice can be the atomic activity). As we intend to apply the system in a simulation, we need to describe the activities and their expected results, but we do not expect the system to actually solve the problems. Moreover, we are integrating the concept of a process modeling technique into an existing agent framework (i.e. we do not model the whole MAS) and we set the simulation into a virtual environment.

3 Agent-Based Simulation Enhancement of a Process Language

In order to successfully simulate processes by means of autonomous agents, we can not use pure BPM language and we need to enhance it with several aspects. As we talk about these enhancements using a general process-like language, we mention the key ideas only and the enhancement of a specific language (EPC) is described in [1] more in detail. Firstly, we discuss inputs, effects, actors, and location of the activity, followed by more advanced concepts.

Inputs are representing the objects that are used during the execution. Moreover, they are also interpreted as a logical input condition that describes what state the attributes of the object should be in, in order to use it in the activity (e.g. we want to handle only new invoices in a process, hence the state of the invoice object should be “new”). *Effects* represent the adding value of a process. Two cases have to be distinguished: when a new object is created, and when an existing object (the one that is already used by the process as an input object) is modified. *Actors* represent a role that agent has to possess in order to execute the activity. We assume that the roles are organized in a tree hierarchy, agents can possess several roles, and a single role can be handled by several agents (e.g. agent Jane is a secretary, but she is also an employee and a human). Finally, *location* points to a place where the activity should be executed in a virtual world.

Beyond these quite straightforward aspects, we identify several more advanced characteristics. Objects and actors can be *optional* in their participation in an action, and they both can be used in more activities *simultaneously*, thus parameters regarding their utilization have to be specified for every activity. Furthermore, we want to utilize the *cooperation* of several agents participating on a single activity, hence we have to allow the assignment of more actors to a process.

The crucial part of the simulation is the course of an activity as such. We describe it using a mathematical function that can be defined for each output effect of a single process separately, meaning we are modeling several courses of changes in time – one for each output effect (e.g. the state of an invoice can during an execution of a single process change from “new” through “verified” to “prepared for payment”, but the amount of money at the same invoice can be increasing linearly during the process due to a fine). Thanks to using general mathematical functions we can determine the precise state of all output effects at any time and we are able to apply partial results into a virtual world when an interruption occurs. Because all of the described functions have only one input variable – discrete time – we can transform the set of functions as a single multidimensional *transition function* of the process. Finally, according to a real-life practice, we expect the real course of the function during the simulation to depend on the actual state of environment and input objects (e.g. if we have some of the optional input objects we need less time to finish). Hence the transition function has to be parameterized by these aspects.

In order to successfully simulate the passage through a process sequence that contains splitting points, we also have to decide which of the continuations shall we follow. We prefer using a probabilistic values at the decision point rather than an attempt to formalize the logic of the decision been made. We argue that gaining the probabilistic values can be easier than the latter approach, as the real-life decisions are usually too hard to formalize.

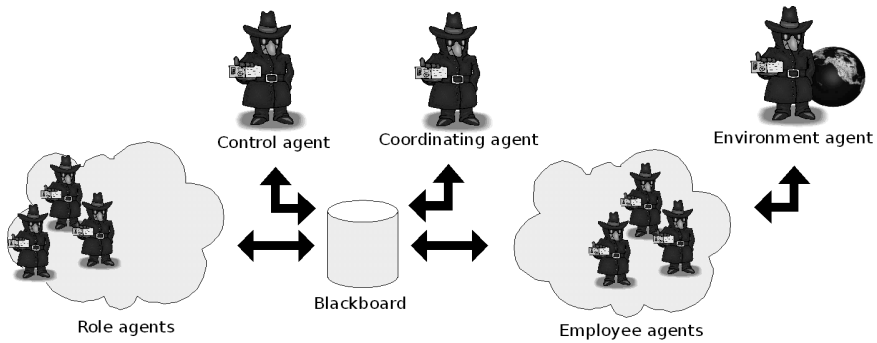


Fig. 2. Architecture of the multi-agent system for a company simulation

4 From Enhanced Processes to an Agent-Based Simulation

In this section we describe the transformation of extended processes into an agent-based simulation together with the architecture of the target MAS that simulates a modeled company. At first we focus on duties of individual agents in MAS followed by the description of the transforming algorithm.

4.1 Architecture of the Multi-Agent System

The architecture of the MAS is shown in Figure 2. We differentiate several types of agents, but there are three main groups. The first is the *environment agent* representing the virtual world. Secondly, there are *employees agents* that correspond with the virtual employees and they have their virtual body in the virtual world. Finally, we identify three types of auxiliary agents (*control*, *coordinating* and *role agents*) which help to organize *employee agents* in case of more complicated scenarios. In our approach we are using the blackboard architecture, where every agent is able to read and write at the common blackboard. The reason for this choice results from an easier realization of coordination of agents as well as an easier implementation. For *employee agents* we use hierarchical reactive plans as the inner control architecture. We based this decision on the fact, that hierarchical rules can already define a complex behavior and they further can be enhanced with planning or learning.

Let us present the simplest scenario of the simulation of a single activity: An *employee agent* reads from the blackboard the set of currently allowed actions (based entirely on process chains, the input conditions of actions are not considered here), it autonomously chooses one of them on the basis of its internal rules and the ability to satisfy the input conditions, and commits itself to execute it. It asks the transition function of the process, what is the expected finish time of this instance of the activity (as it can depend on the actual values of input objects attributes), and after the specified time it applies the target values of the effects of the activity as provided by the transition function and marks the activity as finished at the blackboard. However, during the execution of the activity, the agent can suspend its work (e.g. because it needs to accomplish a task with higher priority). At the time of the occurrence of this

suspension, the agent asks the transition function for actual values of all effects and reflects the partial changes in the environment.

Now, we focus on the functioning of the whole system. The *control agent* is the one who controls the correct order of the process execution according to the process chains and sets the set of currently allowed activities. In the case of cooperation of several agents in a process execution, the *coordinating agent* takes responsibility for notifying the correct subordinate agents and monitors their progress. Note that in this case, only one *employee agent* gets to actually execute the activity as explained above (so called *master agent*) and the other ones are considered to be only passive observers. However, the *coordinating agent* is necessary again in the case of an interruption, where it chooses one of the other participating agents to be the master. Finally, we describe the duties of the *role agents*. As we have stated in Section 3, we are using the concept of roles. Hence the *role agent* reads the set of currently active processes for the given role (set by *control* or *coordinating agent*) and activates them for selected *employee agent* that will execute them. Furthermore, when an interruption occurs and the suspended agent should be replaced by another one, *role agent* is responsible for notifying another *employee agent* possessing the same role.

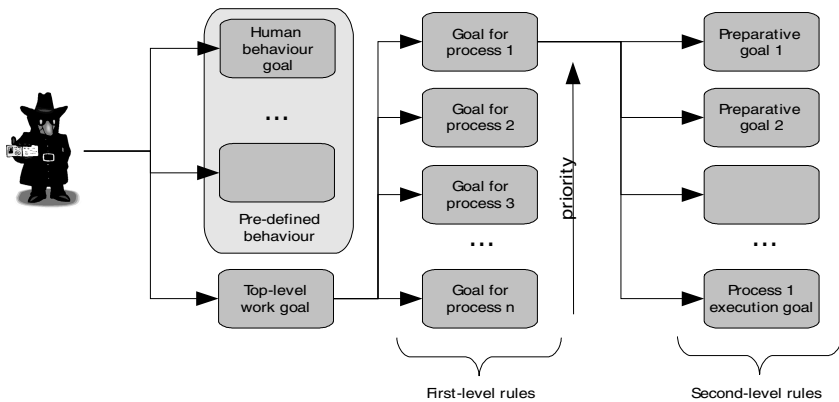


Fig. 3. Plan visualization for an employee agent

4.2 Transforming Algorithm

Let us now describe how the algorithm for automatic hierarchical reactive plans generation works (see Figure 3). As we have already stated, we are using the reactive architecture in our implementation of the algorithm, hence each goal of the plan is represented by a fuzzy if-then rule. For each process the *employee agent* can participate in, one rule is automatically generated. These rules are for each agent ordered by the descending priority of the activities and they form the first level of hierarchical architecture of the agent. The second layer is created by several sets of rules, where each set is collocated with one first-level rule. This second-level set of rules then represent several partial activities that are necessary to accomplish according to the conventions in the virtual world (e.g. transporting movable objects to the place of the execution of the process), and one rule for executing the simulation of the activity as

such (modeled by a *transition function* as described in Section 3). Except the last one, the nature of these rules depends on the conventions that hold in the virtual world and therefore cannot be generalized.

The condition of a first-level rule is created as a conjunction of all constraints related to properties of input objects and participants (i.e. correct values of their utilization (whether they can execute this activity) and possibly other attributes, such as the state of an invoice etc.), and activation of an appropriate process. Moreover, if an input object or a participant is not mandatory, related conditions do not need to hold in order to fire the rule.

5 Prototype and a Case Study

In order to verify proposed process language enhancement, transforming algorithm together with the multi-agent system, we implemented a working prototype of a company simulation. For brevity, we highlight only the key parts and the details can be found in [1].

From several possible BPM languages, we chose and enhanced EPC. The observations defined in Section 3 are mostly semantic and we use standard EPC objects and their properties to capture them (e.g. we use orientation of “relation edges” connecting processes with data objects to differentiate input and output objects, we specify properties to capture their utilization in the process, obligation, etc.). Furthermore, we implemented these process language enhancements as an extension of existing EPC formal notation language EPML [7], and we created a new language which we call A-EPML. As a formalization of the transition function we used a Java class that can be highly configured through XML parameters stored in an A-EPML file. Using the new element of the A-EPML language, we can connect the Java class with the activity – e.g. a general linear function for each output effect is implemented in Java, but the parameters (e.g. different slopes of linear functions for separate effects) are passed through the XML configuration. This way we satisfied all demands on the transition function defined in Section 3 (we can easily implement a configurable multidimensional mathematical function using Java) together with preserving the possibility of easy visual modeling (user can set these properties using a visual tool).

As the next step we selected a proper multi-agent simulation framework – Intelligent Virtual Environment¹ – and implemented the transforming algorithm that creates sets of hierarchical rules from A-EPML files for each employee agent as described in Section 4.2. Finally, we created a case-study: a simulation of a small software development team. We designed the virtual world representing several offices with computers, desks and six employees (see Figure 4), and implemented pre-defined behavior of auxiliary agents and employee agents (e.g. walking, transporting objects). We modeled four EPC process chains on the basis of a real company. We selected the ones that, on the one hand, cover some cognate areas, but they as well cover most of the possibilities in enhanced EPC modeling – we chose repairing a broken computer, fixing a bug in software, implementing and testing a new feature in a software product and handling an invoice. We successfully applied our algorithm to transform them into rules for agents in IVE, where they acted according to the process model.

¹ <http://urtax.ms.mff.cuni.cz/ive>



Fig. 4. Screenshot of the virtual world of implemented case study

6 Conclusion and Future Work

In this paper we have presented a novel approach combining the process language with the description of agents' behavior creating that way an easy modifiable agent-based simulation of the work practice in a company. We identified necessary additional characteristics that we need to enhance a process language with, designed the algorithm transforming these processes into reactive hierarchical plans for agents, and defined the functioning of a multi-agent system simulating the modeled company. To prove the functionality of our proposal we implemented described enhancement into the EPC language, created an appropriate A-EPML notation to store these processes, and implemented the algorithm transforming them into a multi-agent framework resulting into an agent-based simulation.

Our approach has several advantages. It opens the possibility of the behavior description of agents in the simulation using a simple graphical tool in the form of processes. Furthermore, these agents can easily be visualized in the virtual world representing a company, giving that way a good overview at the course of the simulation.

We argue that the presented method can also have other applications, than the one shown in this paper. Because the process modeling method is quite natural it can be applied in various areas (e.g. formalized medical guidelines). However, there are some limitations in the architecture of the employee agents and dependence on the rules of the virtual world during the transforming algorithm. Hence the future work will be mainly focused on overcoming these disadvantages and applying proposed approach in other areas as well.

Acknowledgments

This research was partially supported by the project of the Institute of Computer Science of Academy of Sciences AV0Z10300504, the projects of the Ministry of

Education of the Czech Republic No. 1M06014 and MSM0021620838, and the project "Information Society" under project 1ET100300517. We would like to thank Lenka Lhotská and Roman Neruda for valuable advices, and we also thank Petr Kocáb from SoftDec company for lending us the ISO protocols for the prototype implementation.

References

1. Bosansky, B.: A Virtual Company Simulation by Means of Autonomous Agents. Master's thesis, Charles University in Prague (2007)
2. Buhler, P.A., Vidal, J.M.: Towards adaptive workflow enactment using multiagent systems. *Inf. Technology and Management* 6, 61–87 (2005)
3. Corradini, F., Merelli, E.: Hermes: Agent-Based Middleware for Mobile Computing. In: Bernardo, M., Bogliolo, A. (eds.) *SFM-Moby 2005*. LNCS, vol. 3465, pp. 234–270. Springer, Heidelberg (2005)
4. De Snoo, C.: Modelling planning processes with TALMOD. Master's thesis, University of Groningen (2005)
5. Finkelstein, A., Kramer, J., Nuseibeh, B., Finkelstein, L., Goedicke, M.: Viewpoints: A framework for integrating multiple perspectives in system development. *Int. Journal of Software Eng. and Knowledge Engineering* 2, 31–57 (1992)
6. Jennings, N.R., Faratin, P., Norman, T.J., O'Brien, P., Odgers, B.: Autonomous agents for business process management. *Int. Journal of Applied Artificial Intelligence* 14, 145–189 (2000)
7. Mendling, J., Nuttgens, M.: XML-based reference modelling: Foundations of an EPC markup language. In: Becker, J., Delfmann, P. (eds.) *Proc. of the 8th GI Workshop Referenzmodellierung 2004 at MKWI 2004*, pp. 51–72 (2004)
8. Moreno, A., Valls, A., Marin, M.: Multi-agent simulation of work teams. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) *CEEMAS 2003*. LNCS (LNAI), vol. 2691, p. 281. Springer, Heidelberg (2003)
9. Scheer, A.W., Nuttgens, M.: ARIS architecture and reference models for business process management, Bus. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) *Business Process Management*. LNCS, vol. 1806, pp. 376–389. Springer, Heidelberg (2000)
10. Sierhuis, M.: Modeling and Simulating Work Practice. PhD thesis, University of Amsterdam (2001)
11. Singh, M.P., Huhns, M.N.: Multiagent systems for workflow. *Int. Journal of Intelligent Syst. in Accounting, Finance and Management* 8, 105–117 (1999)

Temporal-Bounded CBR for the Management of Commitments in RT-Agents*

Marti Navarro, Stella Heras, Vicente Botti, and Vicente Julián

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n. 46022 Valencia (Spain)
Tel.: (+34) 96 387 73 50
{mnavarro,sheras,vbotti,vinglada}@dsic.upv.es

Abstract. In MAS, agents sometimes require services that other agents provide. In addition, agents could ask for the performance of the requested services in a bounded time. The late fulfilment of some services could reduce the quality of the response offered by a multi-agent system. Therefore, before committing itself to performing a service, an agent must analyse if it is able to complete the service within a specific time. This article proposes the use of the case-based reasoning methodology to carry out this analysis. A framework featuring a Commitment Manager that takes commitment decisions on the basis of previous experiences is presented in the paper.

1 Introduction

Real-Time Systems and, more specifically, Real-Time Artificial Intelligence Systems (RTAIS) have emerged as useful techniques for solving complex problems, which require intelligence and real-time response times. In RTAIS, flexible, adaptive and intelligent behaviours are of great importance. Nowadays, the Multi-Agent Systems (MAS) paradigm has appeared to be especially appropriate for developing these kinds of systems [2]. However, the main problem concerning real-time MAS (RT-MAS) and real-time agents (RT-agents) is the merging of intelligent deliberative techniques with real-time actions in complex and distributed environments.

In MAS, agents are grouped together to form communities or organisations that cooperate to achieve individual or collective goals. In most of these systems, the community members have different (but related) problem solving expertise that has to be coordinated for solving problems. Such interactions are necessary due to the dependencies between the actions of the agents, the need for meeting global goals or the agent's lack of sufficient capabilities, resources or information to solve the entire problem. To develop better models of coordination and improve the efficiency of MAS, it is necessary to gain a deeper understanding of the fundamental concepts

* This work was partially supported by CONSOLIDER-INGENIO 2010 under grant CSD2007-00022 and by the Spanish government and FEDER funds under TIN2005-03395 and TIN2006-14630-C0301 projects.

underpinning agent interaction [6]. A way of characterising MAS is to employ the notion of commitments. Commitments are viewed as responsibilities acquired by an agent for the fulfilment of an action under certain conditions concerning other agents [10]. This provides agents with a degree of predictability, taking the activities of others into account when dealing with inter-agent dependencies, global constraints or resource utilisation conflicts. If we apply the notion of commitments in real-time systems, the responsibility acquired by an agent for the accomplishment of an action under (Possibly hard) temporal conditions increases the complexity of these kinds of systems. If one agent delegates a task to another with a determined deadline, the agent which commits itself to developing this task must not fail to fulfil this commitment. Otherwise, a deadline violation in real-time systems may cause serious or catastrophic effects in the system or dramatically reduce the quality of the response.

This work proposes a commitment-based framework for real-time agents based on the Case-Based Reasoning (CBR) methodology. The framework provides an agent with the necessary tools to decide, in a temporal bounded way, if it has the resources and the necessary time to commit itself to the execution of a certain service. This management includes the strict control of the available resources (especially the time available without affecting critical responsibilities). The main component of the framework is a Commitment Manager, which is capable of determining if the agent has the necessary resources and the required time for the successful execution of the commitment. Therefore, this manager will have two key functionalities: to determine the availability of resources, and to estimate the time involved in the execution of the related tasks. The latter functionality employs bounded CBR techniques providing temporal estimations based on previous experiences. This temporal bounded CBR allows a feasibility analysis to be carried out, checking if the agent has enough time to fulfil the commitment, while guaranteeing the real-time constraints of the agent.

The rest of the paper is structured as follows: section 2 shows the Commitment Manager features; section 3 presents the proposed CBR technique; section 4 shows an application example based on mobile robots; and finally, some conclusions are explained in section 5.

2 Real-Time Commitment Management

The Commitment Manager (CM) is a module of a real-time agent aimed at improving the agent's behaviour when it offers services in a real-time environment. This module performs two main functions: (i) to analyse whether or not the agent can satisfy the service requests of other agents. Once a service request is accepted, the agent is committed to its performance for the MAS where the agent is located; (ii) to manage the different commitments that the agent has acquired in order to avoid possible violations. As a last resort, if the agent's integrity is in danger due to an unexpected error, the manager can cancel a commitment. The CM is composed by:

- **The Resources Manager (RM):** with this module the agent can check if it has the necessary resources to execute the related tasks to achieve the goal associated to the service when its request arrives. Otherwise, the module can determine when the agent will have the necessary resources. This analysis calculates when the agent should start the task execution with all of the available resources.

- **The Temporal Constraints Manager (TCM):** before the agent commits to performing a service, it is necessary to verify whether it can complete the service before its deadline. The TCM module performs this verification by using real-time dynamic scheduling techniques to determine whether the task execution is feasible.

The Commitment Manager module works as follows:

1. When a service request arrives, the agent that offers it uses the Commitment Manager module to extract the tasks that can achieve the goal fired by the service.
2. For each task (or set of tasks) that can achieve the goal, the Commitment Manager must: (i) check if the agent has the necessary resources to perform the task (using the RM); and (ii) analyse whether the task can be before the specified deadline (using the Temporal Constraint module).
3. Once the Commitment manager has verified that the agent has all of the necessary resources related to the requested service and, therefore, that the service can be done in time, the agent commits to performing it.
4. The Commitment Manager is in charge of ensuring that the acquired commitments are fulfilled. In cases where a commitment cannot be fulfilled, the agent should renegotiate the commitment or cancel it, reducing the potential negative effects.

One of the main difficulties concerning this manager is obtaining the temporal cost for specific services that include unbounded tasks. To deal with this problem, our approach employs a bounded CBR technique, explained in the following section.

3 RT-CBR

The TCM is in charge of deciding if an agent can commit itself to perform a service without exceeding the maximum time assigned to complete that service. Our approach is to make use of previous experiences when making such a commitment. We assume that, unless unpredictable circumstances arise, an agent can commit itself to performing a service within a specific time if it has been able to do so in a similar situation in the past. This is the main assumption of CBR systems, which hold that similar problems have similar solutions [1]. By using this methodology, agents can *retrieve* previous experiences from a *case-base*, *reuse* the knowledge acquired, *revise* such knowledge to fit the current purposes and finally, *retain* the new knowledge gained in this reasoning process. Therefore, to decide if an agent can commit itself to execute a service within a specific time (following a soft Real Time approach) a Case-Based Reasoning module has been included in the framework. This RT-CBR module is used by the TCM to analyse whether the execution of a task can be finished before the specified deadline in view of similar previous cases stored in a case-base that the manager can accede. The cases of the RT-CBR module must somehow keep information about the time spent on performing specific services in the past. Therefore, the general structure of the cases will be the following:

$$C = \langle S, \{T\} \rangle \quad (1)$$

where S is a structured representation of the problem (i.e. one task of a service) that the case originated from and T is the time that the execution of that task took in the past. Note that T can also be a series of values, since the time spent performing tasks

can change in different contexts. In this case, the estimation obtained by the RT-CBR module will be a function $t: T \rightarrow f(T)$ which, in view of past temporal values, provides an estimation of the necessary time to perform a task. By using this estimation, the expected time to perform a service T_s (consisting of a set of tasks) is the aggregation of the estimated time for each of its tasks:

$$T_s = \sum_{i=1}^I t_i \quad (2)$$

In addition, if a *success probability* $P(T_s)$ can also be estimated, agents could use this probability to take strategic decisions about the commitments that they are asked to fulfil. Furthermore, if a *confidence factor* (CF) which sets a minimum threshold for the success probability could be specified, agents would commit themselves to fulfilling a service if:

$$\exists T_s / P(T_s) \geq CF \wedge T_s \leq \text{deadline} \quad (3)$$

In this case, greater confidence factors give rise to less risky decision strategies.

3.1 Real-Time Case-Based Commitment Management

Due to the soft real-time constraints of the framework, the retrieve and reuse phases of the CBR cycle must be executed in a bounded time. On the one hand, the RT-CBR case-base must be structured in a way that eases the search and retrieval of its cases by using algorithms with a bounded temporal cost. Indexing the case-base as a *Hash Structure*, where the search has a worst case temporal cost $O(n)$, n being the number of cases stored, is a possible solution. However, this choice forces the RT-CBR case-base to keep a maximum number of cases, since the temporal cost of the retrieval depends on the case-base size. In addition, constant maintenance is also required to ensure that the case-base stores useful and up-to-date information.

On the other hand, during the reuse phase the retrieved cases are adapted to be useful in the current situation. Such adaptation must also be performed in a bounded time, since the TCM needs to take a quick decision about whether or not to make a particular commitment, and it has a limited time to such reasoning. Otherwise, a late response from the RT-CBR module will be useless.

Finally, the revise and retain phases of the CBR cycle can be executed off-line. It has been assumed that the revision of the final result achieved with the execution of a task does not affect the decision to commit the agent to performing that task. Thus, the revision of the utility of the advice that the RT-CBR module has provided to the TCM can be performed when the agent has some available time. Similarly, the retention of new information in the case-base can also be performed off-line. This paper is focused on the explanation of the retrieval phase and, indirectly, on the reuse phase, which is coupled with the retrieval in the current RT-CBR design. As explained above, these phases must observe temporal constraints. The rest of the phases of the CBR cycle are omitted due to space restrictions.

4 Application Example

A prototype of a mail robot example has been developed for our proposal. The problem to solve consists of the automated management of the internal and external mail (physical mail, non-electronic) on a department floor. The system created by this automation must be able to request the shipment of mails from an office on one floor to another on the same floor, as well as the reception of external mail at a collection point for later distribution. Once this service has been requested, a set of mobile robots must gather the shipment and direct it to the destination. It is important to note that each mail distribution task must be completed before a maximum time, specified in the shipment request. Thus, the use of real-time agents is necessary in this example. This example is a complex problem and is clearly distributed, which makes the MAS paradigm suitable for its solution. Summarising, we can distinguish three types of agents in the system:

- **Interface Agent:** this agent is in charge of gathering user requests. A is transmitted by the interface agent to the mail service offered by the floor agent. The user can employ a mobile device to communicate with an interface agent.
- **Floor Agent:** the mission of this agent is to gather the delivery/reception of mail sent by the interface agent and to distribute the work among the available robot agents around the plant. The floor agent interacts with the robot agents by means of the invocation of their mail delivery service.
- **Robot Agent:** the robot agent is in charge of controlling a physical robot and managing the mail list that this robot should deliver. In the proposed example, three robot agents are employed. Each robot agent controls a *Pioneer 2* mobile robot. The agent periodically sends information about its position and state to the floor agent. This information is used by the floor agent to decide, at each moment, the most appropriate agent to send a new delivery request.

One of the services offered by the robot agent is mail delivery, which involves its movement from an initial position to a final one. In order for an agent to commit itself to this delivery in a bounded time, a temporal estimation, as accurate as possible, is a must. In this system, the Commitment Manager makes use of the CBR methodology to deal with this requirement.

4.1 CBR Integration

In this example the RT-CBR module has been integrated in the TCM of the robot agent. The CBR methodology has already been applied to control multi-agent robot navigation [7][9], plan individual moves and team strategies in the RoboCup league [8] and develop a hybrid planner for a real-world mobile robot [4]. However, these approaches do not observe a specific deadline for the performance of tasks or services, but they assume that agents have an unbounded time to fulfil them. The purpose of this module is to decide if the agent will be able to perform a requested service without exceeding a specified deadline and, this being the case, commit itself to carrying it out. Hence, the module case-base stores the information about previous shipment experiences which will be used to support the agent in the undertaking or not of a new commitment. The cases are tuples of the form:

$$C = \langle I, F, N_t, N_s, T \rangle \quad (4)$$

where I and F represent the coordinates of the initial and final positions of a path along which the robot travelled in the past, N_t is the total number of times that the robot travelled down this path, N_s is the number of times that the robot successfully completed the path within the case-based estimated time and T is the real time (or the series of time values) that the robot spent travelling on this route. Note that no complete routes (i.e. from the robot original position to the final position that it wants to reach) are stored in the case-base. The paths that it travelled straight down whose composition forms the complete route, are stored as cases though. This structure eases the reuse of cases and allows the time that the agent would spend on travelling down a new route to be inferred by trying to compose the path with several cases.

Before the agent starts to travel the path to perform a shipment service, it must be sure that it can reach the position of the delivery point within the time assigned to this service. Therefore, the agent uses the TCM to decide if it can travel the entire route without exceeding the deadline of the service. So, it uses the RM to estimate if it will have enough power resources to travel the path. The former starts a new reasoning cycle in the RT-CBR module. In each cycle, the module tries to retrieve from the case-base cases that compose the entire route that the agent has to travel. In the best case, the entire route can be composed by using several cases from the case-base or even better, the case-base has a case representing exactly the same route that the agent wants to travel. If that is possible, the estimated time that the agent will spend to complete the shipment can be computed by using the time that it spent in the past to travel each sub-path that composes the route. Therefore, the RT-CBR module could provide the agent with a probability of being able to travel the route on time. Otherwise, at least the module could provide the agent with a probability estimation.

Due to the temporal constraints that the CBR process has to keep, we have followed an *anytime* approach [3] in the design of the retrieval-reuse phase algorithm. Therefore, the reasoning process is performed in an iterative way, providing the agent with a solution at every step (the probability of being able to travel the route). This solution is improved in subsequent steps of the algorithm, while the maximum time assigned to the RT-CBR module for making its estimation has not been exceeded.

Firstly, the algorithm retrieves the cases from the case-base of the module. Then, for each case it uses a *confidence function* to compute the probability of a robot travelling from an initial point to a final one in an area without diverting its direction. In the ideal case whereby the agent has already travelled the path a case representing it is stored in the case-base and the following confidence function will be used:

$$f_{trust}(i, j) = 1 - \frac{dist_{ij}}{\max Dist} * \left(\frac{N_s}{N_t} \right) \text{ where } dist_{ij} \leq \max Dist \quad (5)$$

where $dist_{ij}$ is the distance travelled between the points $\langle i, j \rangle$, N_t represents the total number of times that the robot has travelled the path, N_s represents the number of times that the robot has travelled the path within the case-based estimated time and $\max Dist$ is a threshold that specifies the maximum distance above which the agent is unlikely to reach its objective (the points are too far apart). However, if the agent has

still not travelled the path and the case-base does not have the case, a confidence function that takes into account the distance that separates both points will be used:

$$f_{trust}(i, j) = \begin{cases} 1 - \frac{dist_{ij}}{const_1} & \text{if } 0 \leq dist_{ij} \leq d_1 \\ 1 - const_2 * dist_{ij} & \text{if } d_1 < dist_{ij} \leq d_2 \\ \frac{dist_{ij}}{dist_{ij}^2} & \text{if } d_2 < dist_{ij} \end{cases} \quad (6)$$

where $const_1$ and $const_2$ are normalization parameters defined by the user, $dist_{ij}$ is the Euclidean distance between the points $\langle i, j \rangle$ and d_1 and d_2 are distance bounds. This function computes a smoothed probability that the robot can travel the path straight ahead. Note that if the final point is close to the initial point (below d_1) the confidence function should produce high values. However, as the distance increases, this probability quickly decreases. After a distance d_2 , the probability approaches 0.

Once the probability of the robot reaching its objective is computed for each case, the path with the maximum probability of success must be selected. To perform this selection, the complete path from the initial position to the final one is composed by following an A^* heuristic search approach [5]. The selection function $F(n)$ (7) consists of two functions: $g(n)$ (8) which computes the case-based confidence of travelling from the initial point to a certain point n and $h(n)$ (9) which computes an estimated confidence of travelling from the point n to the final point. This estimation is always greater than the actual confidence.

$$F(n) = g(n) * h(n) \quad (7)$$

$$g(n) = g(m) * f_{trust}(m, n) \quad (8) \quad h(n) = 1 - \frac{dist_{nf}}{\max Dist} \quad dist_{nf} \leq \max Dist \quad (9)$$

$$T(n) = time(n) + E(n) \quad (10)$$

$$time(n) = time(m) + \frac{dist_{mn}}{V_{robot}} + \frac{const_{trust}}{f_{trust}(m, n)} \quad (11) \quad E(n) = \frac{dist_{nf}}{V_{Robot}} \quad (12)$$

Finally, the function $T(n)$ (10) determines if the robot agent has enough time to travel the path from the initial point to the final one. If the time estimated by this function exceeds the maximum time needed to travel the path, the algorithm prunes this specific route, since the agent does not have enough time to fulfil its commitment. This function consists of two functions: $time(n)$ (11) which computes the case-based time of travelling from the initial point to a certain point n and $E(n)$ (12) which computes an estimated time of travelling from the point n to the final point. In (11) $dist_{nf}$ represents the distance between the point n to the final point, V_{robot} is the speed of the robot, $f_{trust}(m, n)$ corresponds to (5) or (6) (depending on the existence of the path in the case-base) and the constant $const_{trust} \in [0, 10]$ represents the degree of caution of the robot agent. Greater values of the constant increase the value of $time(n)$ and hence, the agent will be more cautious. If the RT-CBR algorithm is able to compute the entire path from the initial point to the final point, it returns the case-based

probability of fulfilling the commitment. In cases where the algorithm did not find a solution, it returns the product of the accumulated probability until that moment and a pessimistic probability of travelling from the last point visited to the point that it wants to reach. Finally, if all solutions exceed the time assigned to fulfil the commitment, the algorithm returns a null probability.

5 Conclusions

The agent's commitment management needs deliberative processes in order to decide if the agent has the resources and can commit itself to doing a specific task. In the case of real-time agents, which are situated in real-time environments, a commitment can be temporal bounded. So, the agent must have the necessary mechanisms to study the temporal cost associated with its available tasks. The work presented in this paper introduces a CBR approach for deciding if an agent can commit itself to perform a service without exceeding the deadline assigned for performing that service. The approach provides the necessary tools allowing the agent to decide, in a temporal bounded way, if it has enough time to commit itself to the execution of a specific service. For now, we have focused our work on the design and implementation of the different phases of the CBR cycle for a mobile robot problem. This work is under progress and, at the current time, we are testing the proposal in a simulated scenario.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning; Foundational issues, methodological variations, and system approaches. *AI Communications* 1(7), 39–59 (1994)
2. Botti, V., Carrascosa, C., Julian, V., Soler, J.: Modelling Agents in Hard Real-Time Environments. In: Garijo, F.J., Boman, M. (eds.) *MAAMAW 1999*. LNCS, vol. 1647, pp. 63–76. Springer, Heidelberg (1999)
3. Dean, T., Boddy, M.: An analysis of time-dependent planning. In: *Proceedings of the seventh National Conference on Artificial Intelligence*, pp. 49–54 (1988)
4. Fox, S.E.: Behavior Retrieval for Robot Control in a Unified CBR Hybrid Planner. In: *Proceedings of FLAIRS 2001*, pp. 98–102 (2001)
5. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on SSC* 4, 100–107 (1968)
6. Jennings, N.R.: Commitments and conventions: The foundation of coordination in multi-agent systems. *Knowledge Engineering Review* 8(3), 223–250 (1993)
7. Likhachev, M., Kaess, M., Kira, Z., Arkin, R.C.: *Spatio-Temporal Case-Based Reasoning for Efficient Reactive Robot Navigation*. Mobile Robot Laboratory, College of computing, Georgia Institute of Technology (2005)
8. Marling, C., Tomko, M., Gillen, M., Alexander, D., Chelberg, D.: Case-Based Reasoning for Planning and World Modeling in the RoboCup Small Size League. In: *IJCAI 2003 Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments* (2003)
9. Ros, R., López de Mántaras, R., Sierra, C., Arcos, J.L.: A CBR System for Autonomous Robot Navigation. In: *Proceedings of CCIA 2005, Frontiers in Artificial Intelligence and Applications*, vol. 131, pp. 299–306 (2005)
10. Winikoff, M.: Designing commitment-based agent. In: *IAT 2006*, pp. 18–22 (2006)

A Constrained Dynamic Evolutionary Algorithm with Adaptive Penalty Coefficient

Bo Xiao, Danpin Yu, Lei Zhang, Xin Tian, Song Gao, and Sanyou Zeng

Dept. of Computer Science, Technology Research Center for Space Science& Technology
and The State Key Laboratory of Geological Processes and Mineral Resources,
China University of Geosciences, Wuhan, 430074, China
sanyou-zeng@263.net

Abstract. This paper proposes a new evolutionary algorithm with adaptive penalty coefficient. Firstly, the crossover operator of the new algorithm searches a lower-dimensional neighbor of the parent points, so that the algorithm converges fast, especially for high-dimensional problems. Secondly, the violation values of all constraint functions and the value of the objective function are normalized, and therefore, only one penalty coefficient is needed in the scheme. The penalty coefficient is selected adaptively. It is not too big, so as the algorithm can converge fast, and it is not too small so as the algorithm can avoid local optimal as much as possible. Thirdly, the standard deviation of violation values of the constraint functions is added to the violation item in the penalty function, and therefore, the individuals in the population can evenly approach the feasible region from the infeasible space. We have used the 24 constrained benchmark problems to test the new algorithm. The experimental results show it works better than or competitive to a known effective algorithm [7]

1 Introduction

EAs for constrained optimization can be classified into several categories by the way the constraints are treated:

- 1) Constraints are only used to see whether a search point is feasible or not [1], [2].
- 2) The constraint violation and the objective function are used separately and are optimized separately [3]–[8]. These methods adopt a lexicographic order, in which the constraint violation precedes the objective function, with relaxation of the constraints. The methods can effectively optimize problems with equality constraints through the relaxation of the constraints. Deb [3] proposed a method using an extended objective function that realizes the lexicographic ordering. Runarsson and Yao [4] proposed a stochastic ranking method based on ES and using a stochastic lexicographic order that ignores constraint violations with some probability. These methods have been successfully applied to various problems.
- 3) The constraints and the objective function are optimized by multi-objective optimization methods [9]– [12].

- 4) The constraint violation, which is the sum of the violation of all constraint functions, is combined with the objective function. Approaches in this category are called penalty function methods.

This paper proposed a new evolutionary algorithm and tried to overcome the difficulty in the category 4. The crossover operator of the new algorithm searches a lower-dimensional neighbor of the parent points where the neighbor center is the barycenter of the parents, and therefore the new algorithm converges fast. A new scheme is proposed to control the penalty coefficient dynamically. The violation values of all constraint functions and the value of the objective function are normalized first, and then only one penalty coefficient is needed in the scheme. The normalization for all individuals in the population is executed in each generation. Therefore, each generation of evolution has a determined penalty coefficient and the coefficient in a generation is probably different from the one in the next generation.

2 The Used EA

2.1 The Framework of the Used EA

Real-code is used in the algorithm for individual $\vec{x} = (x_1, x_2, \dots, x_n)$, where x_j is the j th gene. Denote l_j, u_j be the lower boundary and the upper boundary of $x_j, j=1, 2, \dots, n$. For each individual (as father) in the population, the new algorithm generates an offspring to compete its father. If the offspring is better than the father, then the father is replaced by the new one.

The framework of the used EA is as followed:

Algorithm 1 the framework of the used EA

*Step1 Randomly create a population $P(0)$ of size N . Set the generation counter $t = 0$
REPEAT*

Step2 For individual \vec{x}^i , Execute offspring generation operator with output \vec{s}^i

(Algorithm 2), $i=1, 2, \dots, N$, let $S(t) = \bigcup_{i=1}^N \vec{s}^i$

Step3 For individual \vec{x}^i , Compare \vec{s}^i with \vec{x}^i . If \vec{s}^i is not worse than \vec{x}^i , then

\vec{x}^i is replaced by $\vec{s}^i, i=1, 2, \dots, N$, let $P(t+1) = \bigcup_{i=1}^N \vec{x}^i$

Step4 Generation $t = t+1$

UNTIL evolution ends

2.2 The Offspring Generation Operator

The offspring generation operator for father \vec{x}^i includes three operators: crossover, copy and mutation which are all based on gene. The details of the operator are as followed:

Algorithm 2 offspring generation operator for father $\vec{x}^i = (x_1^i, x_2^i, \dots, x_n^i)$

Step1. Randomly choose M different individuals $\vec{x}^{\rightarrow p_1}, \vec{x}^{\rightarrow p_2}, \dots, \vec{x}^{\rightarrow p_M}$ from population

$P(t)$. They are all different from \vec{x}^i .

Step2. Crossover operator: Randomly create a_1, a_2, \dots, a_M , with $a_1 + a_2 + \dots + a_M = 1$ in the range $-1 \leq a_1, a_2, \dots, a_M \leq M$

$$\text{Let } \vec{s} \leftarrow a_1 \vec{x}^{\rightarrow p_1} + a_2 \vec{x}^{\rightarrow p_2} + \dots + a_M \vec{x}^{\rightarrow p_M}, \quad (1)$$

denote $\vec{s} = (s_1', s_2', \dots, s_n')$

Step3. For each gene position j ($1 \leq j \leq n$)

Step 3.1. Let $s_j'' \leftarrow s_j'$ with crossover probability p_c and $s_j'' \leftarrow x_j$ with copy probability $1 - p_c$

Step3.2. Let $s_j \leftarrow \text{random}(l_j, u_j)$ with mutation probability p_m and $s_j \leftarrow s_j''$ with probability $1 - p_m$

Step4. Output $\vec{s} = (s_1, s_2, \dots, s_n)$

Like evolution strategies (ES)[13], evolutionary programming (EP)[14], differential evolution (DE)[15] and particle swarm optimization (PSO)[16], the new algorithm in this paper is suitable for solving real-parameter problems. We shall use it to solve real-parameter optimization problems with constraints. However, for dealing with the constraints, some more work has to do for the algorithm, which will be given in the following section.

3 The Constrained EA with Adaptive Penalty Coefficient

3.1 General Formulation of Constrained Optimization Problem for Minimization

A general constraints optimization problems is described as in Equation 2 :

$$\begin{aligned} Q: & \text{Minimize } f(\vec{x}) \\ & \text{Subject to} \\ & g_i(\vec{x}) \leq 0 \quad i = 1, \dots, q \\ & h_i(\vec{x}) = 0 \quad i = q + 1, \dots, m \\ & X = \{ \vec{x} = (x_1, x_2, \dots, x_n) \mid l_i \leq x_i \leq u_i \} \\ & \vec{l} = (l_1, l_2, \dots, l_n) \\ & \vec{u} = (u_1, u_2, \dots, u_n) \end{aligned} \quad (2)$$

where \vec{x} is the decision vector, X denotes the decision space, \vec{l} and \vec{u} are the upper bounds and lower bounds of the decision space.

Usually equality constraints are transformed into inequalities of the form

$$|h_i(\vec{x})| - \delta \leq 0, i = q + 1, \dots, m$$

A solution \vec{x} is regarded as feasible if $g_i(\vec{x}) \leq 0, i=1, \dots, q$, and $|h_i(\vec{x})| - \delta \leq 0, i=q+1, \dots, m$. In this paper δ is set to 0.0001

3.2 Normalization of the Decision Space

In this paper, the decision space X is normalized by the following linear transformation:

$$x_i \leftarrow \frac{(u_i - x_i)}{(u_i - l_i)}, i = 1, \dots, n \tag{3}$$

After the transformation, we have $x_i \in [0, 1]$.

3.3 Dynamic Format of the Constraint Optimization Problem

Literature [17] proposed a ϵ -Constrained method to deal with equality constraints. Here we interpret the method in a dynamic way. The constrained optimization problem is transformed into the following dynamic optimization problem:

$$\begin{aligned}
 Q(t): \text{Minimize } & f(\vec{x}) \\
 \text{Subject to} & \\
 & g_i(\vec{x}) \leq 0 \quad i = 1, \dots, q \\
 & |h_i(\vec{x})| \leq \epsilon_{hi}(t) \quad i = q + 1, \dots, m
 \end{aligned} \tag{4}$$

Where $\lim_{t \rightarrow \infty} \epsilon_{hi}(t) = 0$. With $t \rightarrow \infty$, the problem $Q(t)$ in Equation 4 will approach the problem Q in Equation 2. Let denote $\lim_{t \rightarrow \infty} Q(t) = Q$. Suppose t denote the generation of an evolutionary algorithm and let the evolutionary algorithm solve problem $Q(t)$ at generation t , then with $t \rightarrow \infty$, the evolutionary algorithm will solve the problem Q finally. Based on this idea, we use evolutionary algorithm to solve problem Q in Equation 2.

3.4 Calculating Penalty Coefficient for the Problem $Q(t)$

Denote the violation of constraint functions

$$G_i(\vec{x}) = \max\{0, g_i(\vec{x})\}, i = 1, \dots, q \tag{5}$$

$$H_i(\vec{x}, t) = \max\{\epsilon_{hi}(t), |h_i(\vec{x})|\}, i = q + 1, \dots, m \tag{6}$$

The penalty function in the usually penalty function methods is as follow

$$F(\vec{x}, t) = f(\vec{x}) + \sum_{i=1}^q \alpha_i G_i(\vec{x})^{\beta_i} + \sum_{i=q+1}^m \alpha_i H_i(\vec{x}, t)^{\beta_i} \tag{7}$$

It is difficult to select appropriate values for the penalty coefficients α_i, β_i in Equation 7, that adjusts the strength of the penalty. The algorithm may converge very slow with big α_i, β_i , while it may trap in local optimal with too small α_i, β_i . This paper tries to overcome the difficulty by using two skills.

The first skill is to use one coefficient only by normalizing the value of objective and the violation values of constraints.

The normalization of the objective value and the violation values of constraints are as follows:

$$\bar{f}(\vec{x}, t) = \frac{f(\vec{x}) - \min_{\vec{x} \in P(t) \cup S(t)} f(\vec{x})}{\max_{\vec{x} \in P(t) \cup S(t)} f(\vec{x}) - \min_{\vec{x} \in P(t) \cup S(t)} f(\vec{x})} \tag{8}$$

$$\bar{G}_i(\vec{x}, t) = \frac{G_i(\vec{x})}{\max_{\vec{x} \in P(t) \cup S(t)} G_i(\vec{x})}, i=1, \dots, q \tag{9}$$

$$\bar{H}_i(\vec{x}, t) = \frac{H_i(\vec{x}, t)}{\max_{\vec{x} \in P(t) \cup S(t)} H_i(\vec{x}, t)}, i=q+1, \dots, m \tag{10}$$

where $P(t)$ is the population and $S(t)$ is the off-springs generated in generation t .

With the normalization, only one coefficient is needed for the penalty function. The Equation 7 is adjusted as follow:

$$\bar{F}(\vec{x}, t) = \bar{f}(\vec{x}, t) + \alpha(t) \left(\sum_{i=1}^q \bar{G}_i(\vec{x}, t) + \sum_{i=q+1}^m \bar{H}_i(\vec{x}, t) \right) \tag{11}$$

For a problem with many constraints, even only one constraint is violated while all others are satisfying, the solution is infeasible yet. Therefore, we expect all constraints approach feasible region simultaneously. The standard deviation of violation values of the constraint functions can be used to measure whether all constraints approach feasible region simultaneously.

$$\sigma(t) = \sqrt{\frac{1}{m} \left(\sum_{i=1}^q (\bar{G}_i(\vec{x}, t) - M(t))^2 + \sum_{i=q+1}^m (\bar{H}_i(\vec{x}, t) - M(t))^2 \right)} \tag{12}$$

$$M(t) = \frac{1}{m} \left(\sum_{i=1}^q \bar{G}_i(\vec{x}, t) + \sum_{i=q+1}^m \bar{H}_i(\vec{x}, t) \right)$$

The smaller the deviation, the more simultaneously the constraints approach feasible region. Adding $\sigma(t)$ to the violation item in Equation 11, we have

$$\bar{F}(\vec{x}, t) = \bar{f}(\vec{x}, t) + \alpha(t) \left(\sum_{i=1}^q \bar{G}_i(\vec{x}, t) + \sum_{i=q+1}^m \bar{H}_i(\vec{x}, t) + \sigma(t) \right) \tag{13}$$

To be sure that the algorithm does not trap in local optimal, $\overline{F}(\vec{x}, t)$ must be larger than global optimal. Suppose the global optimal is f_{opt} , the following inequality must be satisfied:

$$f_{opt} < \overline{f}(\vec{x}, t) + \alpha(t) \left(\sum_{i=1}^q \overline{G}_i(\vec{x}, t) + \sum_{i=q+1}^m \overline{H}_i(\vec{x}, t) + \sigma(t) \right)$$

That is

$$\alpha(t) > \frac{(f_{opt} - \overline{f}(\vec{x}, t))}{\left(\sum_{i=1}^q \overline{G}_i(\vec{x}, t) + \sum_{i=q+1}^m \overline{H}_i(\vec{x}, t) + \sigma(t) \right)} \quad (14)$$

for \vec{x} in infeasible space. There are two difficulties for the algorithm to select penalty coefficient $\alpha(t)$ appropriately: one is the f_{opt} is unknown, the other is that it is impossible to search all \vec{x} in infeasible space.

The second skill is to select $\alpha(t)$ appropriately. Three cases are discussed.

(1) If all \vec{x} in the population are infeasible, then $\alpha(t)$ chooses infinite value. The value in this paper is $\alpha(t) = e^{30}$.

(2) If some individuals are feasible and some infeasible, then best feasible solution (denote $f_{bestfeasible}$) may replace f_{opt} in Equation 14, while the infeasible solutions in the population are regarded as a sample of the search space. We have

$$\alpha(t) = \max_{\vec{x} \in P(t) \cup S(t)} \frac{(f_{bestfeasible} - \overline{f}(\vec{x}, t))}{\left(\sum_{i=1}^q \overline{G}_i(\vec{x}, t) + \sum_{i=q+1}^m \overline{H}_i(\vec{x}, t) + \sigma(t) \right)} \quad (15)$$

A problem is that infeasible solution \vec{x} may be close to feasible. That is ,the denominator in Equation 15 is close to zero. To avoid such singular cases, we delete the infeasible solution with violation value less than a regular parameter η from the sample. And for the limited representation of the infeasible space of the sample, the $\alpha(t)$ may be too small. We use 1 to replace $f_{bestfeasible}$. Therefore, the Equation 15 is adjusted as follow:

$$\alpha(t) = \max_{\vec{x} \in P(t) \cup S(t) \setminus \left\{ \left(\sum_{i=1}^q \overline{G}_i(\vec{x}, t) + \sum_{i=q+1}^m \overline{H}_i(\vec{x}, t) + \sigma(t) \right) < \eta \right\}} \frac{1 - \overline{f}(\vec{x}, t)}{\sum_{i=1}^q \overline{G}_i(\vec{x}, t) + \sum_{i=q+1}^m \overline{H}_i(\vec{x}, t) + \sigma(t)} \quad (16)$$

(3) If all individuals in the population are feasible, then $\alpha(t)$ uses the penalty coefficient in the last generation, that is $\alpha(t) = \alpha(t-1)$

3.5 Framework of Constrained EA with Adaptive Penalty Coefficient

Adding a step of calculating penalty coefficient dynamically into the EA (see the algorithm 1), we get the new constrained EA with adaptive penalty coefficient. The details are as follows.

Algorithm 3 the constrained EA with adaptive penalty coefficient

Step1. Randomly create population $P(0)$ with size N . Set the generation counter $t = 0$

REPEAT

Step2. For individual \vec{x}^i , Execute offspring generation operator with output \vec{s}^i
 (Algorithm 2), $i=1,2,\dots,N$, let $S(t) = \bigcup_{i=1}^N \vec{s}^i$

Step3. Calculate penalty coefficient $\alpha(t)$ according to Equation 16

Step4. For individual \vec{x}^i , Compare \vec{s}^i with \vec{x}^i . If penalty value of \vec{s}^i is less than that of \vec{x}^i , then \vec{x}^i is replaced by \vec{s}^i , $i=1,2,\dots,N$, let $P(t+1) = \bigcup_{i=1}^N \vec{x}^i$

Step5. Generation $t = t+1$

UNTIL evolution ends

4 Numerical Experimental Results

Twenty four benchmark problems and format for reporting results are specified in “Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization”[18].

4.1 Parameters Setting

(1) Population size $N=50$

(2) Number of Parents $M=5$

In the case of big M , the crossover operator would search a large space and the algorithm would converge slowly, while in the case of small M , the algorithm would be easy to lose diversity of the population and converge immature. This paper sets $M=5$.

(3) Crossover probability $p_c=0.95$ in Algorithm 2.

(4) Mutation probability $p_m=0.05$ in Algorithm 2.

(5) The regular parameter η in the Equation 16: $\eta = 0.00001$.

(6) Max function evaluations: 50000

(7) Constraints relaxation controlling parameter $C=1.02$

$\mathcal{E}_{hi}(t)$ is set by the technique proposed in [17]. In this paper,

$\mathcal{E}_{hi}(t) = \max_{x \in P(0) \cup S(0)} H_i(\vec{x}, 0) / C^t$, The constant $C (> 1)$ is used to control the

relaxation of the constraints.

The parameter C is related to the max evolutionary generation. The evolutionary generation of the new constrained EA is calculated as follow:

Max evolutionary generation = (Max function evaluations)/(Population size) = 50000/50 = 1000

Table 1. Comparison of ϵ -constrained DE(ϵ -CDE) [7] with constrained EA in this paper on 24 benchmark problems where both ϵ -CDE and the new constrained EA runs 25 times for each problem. Each run takes 50000 function evaluations.

EA		g01	g02	g03	g04	g05	g06
	Dimensions	13	20	10	5	4	2
ϵ -C DE	Optimal	-15*	-0.803619104*	-1.000500100*	-3.066553867*	5126.496714007*	6961.813875580*
	Best	-14.999610100	-0.7791391041	-0.9865031000	-30665.538671783	5126.4975271371	-6961.8138755801
	Median	-14.9992216700	-0.7323421041	-0.9487781000	-30665.538671783	5126.5084690071	-6961.8138755801
	Worst	-14.9985624000	-0.6810691041	-0.8868701000	-30665.538671783	5126.6729240071	-6961.8138755801
	Mean	-14.9991557400	-0.7342581041	-0.9461121000	-30665.538671783	5126.5188800071	-6961.8138755801
	Std	0.00025185	0.027681	0.022945	0.000000000	0.034215	0.000000000
New Constrained EA	Best	-14.999999280	-0.789430783	-1.000499022	-30665.538671783	5126.496714060	6961.813875580
	Median	-14.999996163	-0.776563625	-1.000478637	-30665.538671783	5126.496714342	6961.813875580
	Worst	-14.999982913	-0.707152867	-1.000008104	-30665.538671783	5126.496715628	6961.813875580
	Mean	-14.999994546	-0.768555616	-1.000434766	-30665.538671783	5126.496714483	6961.813875580
	Std	0.00000467579	0.0216180984	0.0001137678	0.000000000	0.000000350919	0.000000000
		g07	g08	g09	g10	g11	g12
	Dimensions	10	2	7	8	2	3
ϵ -C DE	Optimal	24.306209068*	-0.095825041*	680.630057374*	7049.248020528*	0.74999*	-1*
	Best	24.30808776818	-0.09582504142	680.63005737441	7049.43558052867	0.749900853777	-1.00000000000
	Median	24.30970816818	-0.09582504142	680.63005737443	7050.65972052867	0.74990441920	-1.00000000000
	Worst	24.31489686818	-0.09582504142	680.63005737445	7054.13472052867	0.74991800100	-1.00000000000
	Mean	24.31009116818	-0.09582504142	680.63005737443	7050.92002052867	0.74990566160	-1.00000000000
	Std	0.0015955	0.000000000	0.000000000	1.1794	0.000043725	0.000000000
New Constrained EA	Best	24.306380619	-0.09582504142	680.630057374	7049.430381773	0.749900000	-1.00000000000
	Median	24.308525694	-0.09582504142	680.630057374	7049.796694924	0.749900000	-1.00000000000
	Worst	24.316516366	-0.09582504142	680.630057375	7050.507236615	0.749900000	-1.00000000000
	Mean	24.308879279	-0.09582504142	680.630057374	7049.802123904	0.749900000	-1.00000000000
	Std	0.0024888949	0.000000000	0.0000000002	0.300609611970	0.0000000000	0.000000000
		g13	g14	g15	g16	g17	g18
	Dimensions	5	10	3	5	6	9
ϵ -C DE	Optimal	0.053941514*	-47.764888459*	961.715022289*	-1.905155258*	8853.539674806*	-0.866025403*
	Best	0.05395117494	-46.74698845949	961.71515014996	-1.905155258	8856.73067480648	-0.86592058378
	Median	0.05397121404	-44.14838845949	961.99646228996	-1.905155258	8882.39067480648	-0.86568604378
	Worst	0.05418376404	-38.75208845949	970.62562228996	-1.905155258	8949.53967480648	-0.86503534378
	Mean	0.05401564404	-43.84818845949	963.04192228996	-1.905155258	8897.21967480648	-0.86562711378
	Std	0.000069894	1.6546	2.2109	0.000000000	36.996	0.00022200
New Constrained EA	Best	0.053941514	-47.760539272	961.715022289	-1.905155258	8853.539674817	-0.866022891
	Median	0.053941514	-47.644873631	961.715022289	-1.905155258	8853.539674891	-0.866010081
	Worst	0.999999998	-46.992665005	961.783391797	-1.905155247	8853.539676014	-0.674688300
	Mean	0.397026119	-47.583584390	961.717757070	-1.905155257	8853.539674963	-0.858347759
	Std	0.415914202	0.18785658112	0.01339763264	0.000000002	0.000000250301	0.0374893387
		g19	g20	g21	g22	g23	g24
	Dimensions	15	24	7	22	9	2
ϵ -C DE	Optimal	32.655592950*	--	193.724510070*	--	-400.05509999*	-5.508013271*
	Best	39.821792950246	--	195.520810070035	--	-208.595100000	-5.508013271
	Median	44.259592950246	--	214.021510070035	--	-27.0751000000	-5.508013271
	Worst	51.9575929502	--	421.3945100700	--	372.094900000	-5.508013271
	Mean	44.41659295025	--	224.16651007003	--	-12.565100000	-5.508013271
	Std	2.9576	--	42.519	--	144.70	0.000000000
New Constrained EA	Best	32.701074248	--	193.733203765	--	-399.981135649	-5.508013271
	Median	32.847013115	--	237.144639646	--	-376.118345593	-5.508013271
	Worst	34.300095848	--	315.440394343	--	-46.8889302658	-5.508013271
	Mean	32.948676684	--	239.595682740	--	-346.041308876	-5.508013271
	Std	0.3155177139	--	43.4666468369	--	77.20020336002	0.000000000

For preserving $\mathcal{E}_{hi}(t) \rightarrow 0$, we set parameter $C=1.02$ when t is the evolutionary generation.

4.2 Results Comparison and Discussion

The new constrained EA is compared with the ϵ -constrained Differential Evolution [7] proposed in 2006, which is the modified version of DE [13]. The ϵ -constrained DE

algorithm uses Gradient-based mutation, while the algorithm in this paper uses no gradient-based mutation. Therefore, the new algorithm does not demand the differentiability of the problems.

Comparison of the computational results of the two algorithms. Both the new algorithm in this paper and the ϵ -CDE algorithm solve the 24 benchmark problems proposed in [18] independently 25 runs where each run evaluates function 50000 times. Table 1 shows the best, the worst, the mean, and the standard deviation of the difference between the best value found f^{best} and the optimal value f^* .

As Table I shows, the performance of the new algorithm is better than or competitive to that of the ϵ -CDE algorithm. The new algorithm is better than the ϵ -CDE algorithm for solving problem 1, 2, 3, 5, 7, 10, 11, 14, 15, 17, 19, 23; the computational results are same for the problem 4, 6, 8, 12, 16, 24; the new algorithm is a little bit worse than the ϵ -CDE algorithm for the problem 9, 13, 18, 21. For the seven higher-dimensional problems with space dimensions over 10, computational, the results of the new algorithm are better than that of ϵ -CDE.

Problem 20 and 22 are over-constrained, No results are listed.

Limit of the new algorithm. The diversity of the population is not preserved well.

5 Conclusions

This article introduced a constrained EA with adaptive penalty coefficient. The efficiency of the new constrained EA is derived mainly from the following points:

1. The crossover operator of the new algorithm search a lower dimensional space no matter how many dimensions the decision space of the optimization problem is. Therefore, the new algorithm converges fast especially for optimization problems with higher dimensions.

2. A new scheme is proposed to control the penalty coefficient dynamically. The violation values of all constraint functions and the value of the objective function are normalized, and therefore, only one penalty coefficient is needed in the scheme. The penalty coefficient is selected adaptively. It is not too big, so as the algorithm can converge fast, and it is not too small so as the algorithm can avoid local optimal as much as possible

3. The standard deviation of violation values of the constraint functions is added to the violation item in the penalty function, and therefore, the individuals in the population can evenly approach the feasible region from the infeasible space.

Acknowledgments. This work was supported by The National Natural Science Foundation of China (Nos: 60473037, 60473081, 40275034, 60204001, 60133010), and by the open research program of the Geological Processes and Mineral Resources (GPMR), China University of Geosciences (No. GPMR200618).

References

1. Michalewicz, Z., Nazhiyath, G.: Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints. In: Proceedings of the Second IEEE International Conference on Evolutionary Computation, pp. 647–651. IEEE Press, Los Alamitos (1995)

2. El-Gallad, A.I., El-Hawary, M.E., Sallam, A.A.: Swarming of intelligent particles for solving the nonlinear constrained optimization problem. *Engineering Intelligent Systems for Electrical Engineering and Communications* 9(3), 155–163 (2001)
3. Deb, K.: An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* 186(2/4), 311–338 (2000)
4. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 4(3), 284–294 (2000)
5. Mezura-Montes, E., Coello, C.A.C.: A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Trans. on Evolutionary Computation* 9(1), 1–17 (2005)
6. Venkatraman, S., Yen, G.G.: A generic framework for constrained optimization using genetic algorithms. *IEEE Trans. on Evolutionary Computation* 9(4), 424–435 (2005)
7. Takahama, T., Sakai, S.: Constrained Optimization by the ϵ -Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites. In: *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, Canada, pp. 308–315 (2006)
8. Takahama, T., Sakai, S.: Learning fuzzy control rules by a constrained powell's method. In: *Proceedings of 1999 IEEE International Fuzzy Systems Conference*, Seoul, Korea, vol. 2, pp. 650–655 (August 1999)
9. Camponogara, E., Talukdar, S.N.: A genetic algorithm for constrained and multiobjective optimization. In: *Alander, J.T. (ed.) 3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA)*, pp. 49–62. University of Vaasa, Vaasa (August 1997)
10. Surry, P.D., Radcliffe, N.J.: The COMOGA method: Constrained optimisation by multiobjective genetic algorithms. *Control and Cybernetics* 26(3), 391–412 (1997)
11. Runarsson, T.P., Yao, X.: Evolutionary search and constraint violations. In: *Proceedings of the 2003 Congress on Evolutionary Computation*, vol. 2, pp. 1414–1419. IEEE Service Center, Piscataway (2003)
12. Aguirre, A.H., Rionda, S.B., Coello, C.A.C., Lizárraga, G.L., Montes, E.M.: Handling constraints using multiobjective optimization concepts. *International Journal for Numerical Methods in Engineering* 59(15), 1989–2017 (2004)
13. Schwefel, H.P.: *Evolution and Optimum Seeking*. John Wiley & Sons, New York (1995)
14. Fogel, L.J., Owens, A.J., Walsh, M.J.: *Artificial Intelligence through Simulated Evolution*. John Wiley, New York (1966)
15. 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)
16. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *Proc. IEEE Int. Conf. Neural Networks*, pp. 1942–1948. IEEE Press, Piscataway (1995)
17. Hamida, S.B., Schoenauer, M.: ASCHEA: New results using adaptive segregational constraint handling. In: *Proc. Congr. Evolutionary Computation*, vol. 1, pp. 884–889 (2002)
18. Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello, C.A.C., Deb, K.: Problem definitions and evaluation criteria for the CEC2006 special session on constrained real-parameter optimization (2006)

Enhanced Cooperative Co-evolution Genetic Algorithm for Rule-Based Pattern Classification

Fangming Zhu¹ and Sheng-Uei Guan²

¹ Institute of Systems Science
National University of Singapore
25 Heng Mui Keng Terrace, Singapore 119615

² School of Engineering and Design
Brunel University
Uxbridge, Middlesex, UB8 3PH, UK

Abstract. Genetic algorithms (GAs) have been widely used as soft computing techniques in various application domains, while cooperative co-evolution algorithms were proposed in the literature to improve the performance of basic GAs. In this paper, an enhanced cooperative co-evolution genetic algorithm (ECCGA) is proposed for rule-based pattern classification. Concurrent local and global evolution and conclusive global evolution are proposed to improve further the classification performance. Different approaches of ECCGA are evaluated on benchmark classification data sets, and the results show that ECCGA can achieve better performance than the cooperative co-evolution genetic algorithm and normal GA.

Keywords: genetic algorithms, cooperative co-evolution, classifiers.

1 Introduction

As typical algorithms in evolutionary computation, Genetic algorithms (GAs) have attracted much attention and become one of the most popular techniques for pattern classification [1][2][3]. Among these systems, rule-based solution is widely used for pattern classification problems, either through supervised or unsupervised learning [4].

In the literature, various models and approaches have been proposed to address difficulties in mapping the domain solutions into GA models, while avoiding the possibility of being trapped into local optima. For example, Holland [5] indicated that crossover induces a linkage phenomenon. It has been shown that GAs work well only if the building blocks are tightly linked on the chromosome [1]. In order to tackle the linkage-learning problem, some algorithms have been proposed to include linkage design into problem representation and recombination operator or use some probabilistic-based models [6]. For multi-objective optimization problems, incremental multi-objective GAs have been employed to search for the Pareto-optimal set more accurately and efficiently [7]. In another aspect, complex systems can be decomposed and evolved in the form of interacting co-evolution systems. Classifier systems

evolves interacting rule whose individual fitness are determined by their interaction with other rules, and concept of niches and species is employed [8].

Cooperative co-evolution has attracted more research interests as an effective approach to decompose complex structure and achieve better performance. The idea of cooperative co-evolution mainly derives from the species in the nature. As a normal practice, a complex problem may be decomposed into several sub-problems. The partial solutions in the subpopulations evolve separately, but with a common objective. Cooperative co-evolution, together with incremental learning, has been applied with many soft computing techniques such as neural networks [9]. The introductive work on cooperative co-evolution in the GA domain was conducted by Dejong and Potter [10]. In their work, the cooperative co-evolution genetic algorithm (CCGA) was initially designed for function optimization, and later the general architecture was proposed for cooperative co-evolution with co-adapted subcomponents.

In this paper, the cooperative co-evolution scheme is revisited with a rule-based GA system for pattern classification [11] [12]. In order to improve further the classification performance, an enhanced cooperative co-evolution genetic algorithm (ECCGA) approach is proposed. As a hybrid artificial intelligence system, it integrates rule-based inference and genetic algorithms. The concurrent global and local evolution and conclusive global evolution are integrated into the ECCGA. Different approaches with ECCGA are evaluated on benchmark data sets. The experimental results show that ECCGA performs better than the CCGA and normal GA.

The integration of the local fitness element is a major feature of ECCGA. We postulate that, apart from the global fitness as exercised normally in CCGA, the local fitness element is also a suitable indicator and facilitator for the whole evolution. Therefore, both local and global fitness are employed to guide the evolution. Our experimental results have supported such postulation. It is found that the additional evolution pressure from the local fitness element may produce more opportunities to escape from traps in local optima and advance the evolution further.

2 Design of GA and CCGA

In normal GA, an initial population is created randomly. Based on fitness evaluation, some chromosomes are selected by a selection mechanism. Crossover and mutation will then be applied to these selected chromosomes and the child population is thus generated. Certain percentage of the parent population will be preserved and the rest will be replaced by the child population. The evolution process will continue until it satisfies the stopping criteria [1] [12].

Let us assume a classification problem has c classes in the n -dimensional pattern space, and p vectors $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $i = 1, 2, \dots, p$, $p \gg c$, are given as training patterns. The task of classification is to assign instances to one out of a set of pre-defined classes, by discovering certain relationship among attributes. Then, the discovered rules can be evaluated by classification accuracy or error rate either on the training data or test data.

```

Decompose the problem into  $n$  species;
gen=0;
for each species  $s$ 
    { randomly initialize population  $p(gen)$ ;
      evaluate fitness of each individual;
    }
while (not termination condition)
    { gen++;
      for each species  $s$ 
        { select  $p(gen)$  from  $p(gen-1)$  based on fitness;
          apply genetic operators to  $p(gen)$ ;
          evaluate fitness of each individual in  $p(gen)$ ;
        }
    }
    
```

Fig. 1. Pseudocodes of CCGA

Figure 1 shows the algorithm of CCGA. As the first step, the original problem should be decomposed into n sub-problems, and each of which is handled by one species. Then, each species evolves in a round robin fashion using the same procedure as the normal GA, with the exception of fitness evaluation. When an individual in one species is evaluated, it will be combined with other individuals from the other species and the fitness of the resulting chromosome is evaluated and returned.

In our rule-based GA system, we use the non-fuzzy IF-THEN rules with continuous attributes for classifiers. A rule set consisting of a certain number of rules is a solution candidate for a classification problem. The detailed designs are discussed in the following subsections and can be found further in [11].

2.1 Encoding Mechanism

An IF-THEN rule is represented as follows:

$$R_i : \text{IF } (V_{1\min} \leq x_1 \leq V_{1\max}) \wedge (V_{2\min} \leq x_2 \leq V_{2\max}) \dots \wedge (V_{n\min} \leq x_n \leq V_{n\max}) \text{ THEN } y = C$$

Where R_i is a rule label, n is the number of attributes, (x_1, x_2, \dots, x_n) is the input attribute set, and y is the output class category assigned with a value of C . $V_{j\min}$ and $V_{j\max}$ are the minimum and maximum bounds of the j th attribute x_j respectively. We encode rule R_i according to the diagram shown in Figure 2.

Antecedent Element 1		Antecedent Element n			Consequence Element	
Act_j	$V_{j\min}$	$V_{j\max}$	Act_n	$V_{n\min}$	$V_{n\max}$	C

Notes:

1. Act_j denotes whether condition j is active or inactive, which is encoded as 1 or 0 respectively;
2. If $V_{j\min}$ is larger than $V_{j\max}$ at any time, this element will be regarded as an invalid element.

Fig. 2. Encoding mechanism

Each antecedent element represents an attribute, and the consequence element stands for a class. Each chromosome CR_j consists of a set of classification rules R_i ($i=1,2,\dots,m$) by concatenation:

$$CR_j = \bigcup_{i=1,m} R_i \quad j = 1,2,\dots,s \quad (1)$$

where m is the maximum number of rules allowed for each chromosome, s is the population size. Therefore, one chromosome will represent one rule set. Since we know the discrete value range for each attribute and class *a priori*, V_{jmin} , V_{jmax} , and C can be encoded each as a character by finding their positions in the ranges. Thus, the final chromosome can be encoded as a string.

2.2 Genetic Operators

One-point crossover is used. It can take place anywhere in a chromosome. Referring to the encoding mechanism, the crossover of two chromosomes will not cause inconsistency as all chromosomes have the same structure. On the contrary, the mutation operator has some constraints. Different mutation is available for different elements. For example, if an activeness element is selected for mutation, it will just be toggled. Otherwise when a boundary-value element is selected, the algorithm will select randomly a substitute in the range of that attribute. The rates for mutation and crossover are selected as 0.01 and 1.0.

We set the survival rate or generation gap as 50% (SurvivorsPercent=50%), which means half of the parent chromosomes with higher fitness will survive into the new generation, while the other half will be replaced by the newly created children resulting from crossover and/or mutation. Roulette wheel selection is used as the selection mechanism. In this investigation, the probability that a chromosome will be selected for mating is given by the chromosome's fitness divided by the total fitness of all the chromosomes.

2.3 Fitness Function

As each chromosome in our approach comprises an entire rule set, the fitness function actually measures the collective behavior of the rule set. The fitness function simply measures the percentage of instances that can be correctly classified by the chromosome's rule set.

Since there is more than one rule in a chromosome, it is possible that multiple rules matching the conditions for all the attributes but predicting different classes. We use a voting mechanism to help resolve any conflict. That is, each rule casts a vote for the class predicted by itself, and finally the class with the highest votes is regarded as the conclusive result. If any classes tie on one instance, it means that this instance cannot be classified correctly by this rule set.

2.4 Stopping Criteria

There are three factors in the stopping criteria. The evolution process stops after a preset generation limit, or when the best chromosome's fitness reaches a preset threshold (which is set at 1.0 through this paper), or when the best chromosome's

fitness has shown no improvement over a specified number of generations -- *stagnationLimit*. The detailed settings are reported along with the corresponding results.

3 Design of ECCGA

Figure 3 illustrates the design of normal GA and ECCGA. As shown in Figure 3(a), a normal GA maps attributes to classes directly in a batch manner, which means all the attributes, classes, and training data are used together to train a group of GA chromosomes. ECCGA is significantly different. As shown in Figure 3(b), there are n species (SP), each of which evolves the sub-solution for one attribute in classification. The normal GA is employed in each species to advance the local evolution.

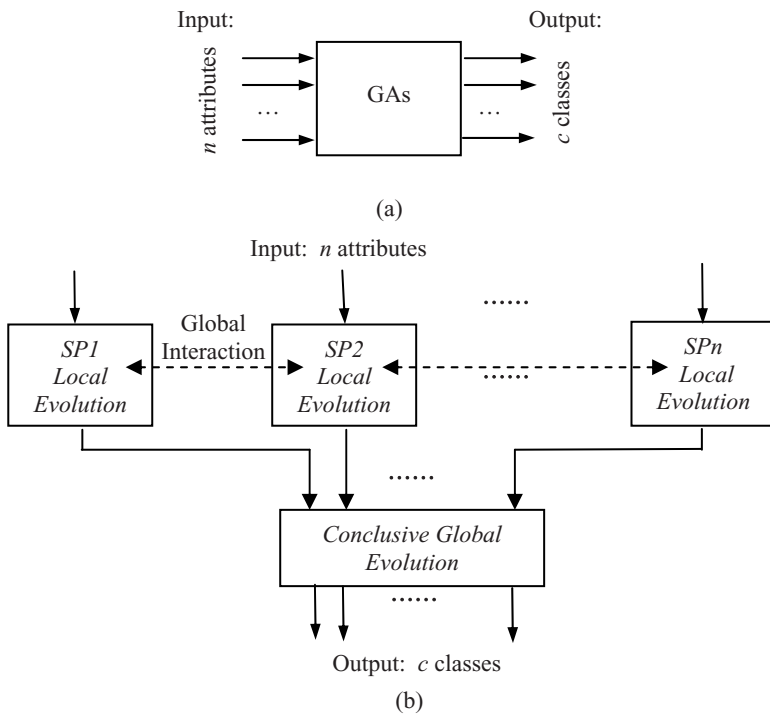


Fig. 3. Illustrations of (a) normal GA and (b) ECCGA

The global interaction in ECCGA can take place in predefined intervals. During such interaction, the global fitness of individuals in species will be assessed. Another enhancement is the introduction of the conclusive global evolution (CGE), which follows the completion of all sub-evolution in species. The objective of CGE is to escape from possible traps by the local optima in the local evolution of species and evolve further to the final solution.

Following the notations presented above, we denote the evolution in each species of ECCGA as:

$$f_i : (X_i) \rightarrow C \quad i = 1, 2, \dots, n \quad (2)$$

where, f_i is a sub-solution for the sub-problem based on the i -th attribute. X_i is the vector of training patterns with the i -th attribute, and C is the set of output classes.

Figure 4 shows the pseudocode of ECCGA. Compared to the CCGA shown in Figure 1, ECCGA has been innovated with new improvement. First, the fitness function is revised. The fitness in CCGA only involves the global fitness, while that of ECCGA consists of two elements, i.e. global fitness and local fitness. The global fitness is obtained with the same method as for CCGA. The local fitness is evaluated on the single attribute in each species. That is, the individual in each species classifies the partially masked training data, where only the training data portion matched with the targeted attribute function will be applied while the portion for the other attributes are treated as non-contributing. The resulting classification rate is recorded as its local fitness. The global and local fitness are then averaged as the representative fitness. In order to save training time, the global fitness is not computed in each generation. Instead, an option for a predefined interval (*genInterval*) is provided. When the evolution in each species advances with a certain number of generations and reaches the preset *genInterval*, the global fitness of each individual will be assessed. Otherwise, the average global fitness in the last generation will be inherited and used as the global fitness element. As normal evolution process advances steadily, the average global fitness will not change abruptly. Therefore, inheriting the average global fitness in the previous generation is an acceptable and reasonable choice. As the evaluation of global fitness involves the chromosome combination process which is time-consuming, it is aimed to save training time to conduct such evaluation in predefined intervals. After all the species reach the termination condition, the evolution will continue with a CGE. An initial population is constructed by combining the randomly selected individuals from the chromosomes in all species, with a definite inclusion of the best global solution recorded during the earlier evolution and the combination of the best chromosomes in all species. A normal GA is then continued until it reaches the stopping criteria. The best chromosome with the highest CR is recorded as the final solution.

Figure 5 shows how the global fitness is obtained during the global interaction. When the preset *genInterval* is met, the chromosomes from all species are combined, and the resulting chromosome is evaluated and the fitness value will be returned as the global fitness component. In order to add more choices and increase robustness, we adopt the method used in [13]. That is, there are two ways to combine chromosomes coming from species. Either the evaluated individual is combined with the current best individual in each other species, or it is combined with individuals selected randomly from each other species. The two resulting chromosomes are then evaluated and the better of them is returned as the individual's global fitness. This method has been fully explored in [13], and it is found that it performs well in function optimization and it successfully escapes from the local optima resulting from interacting variables.

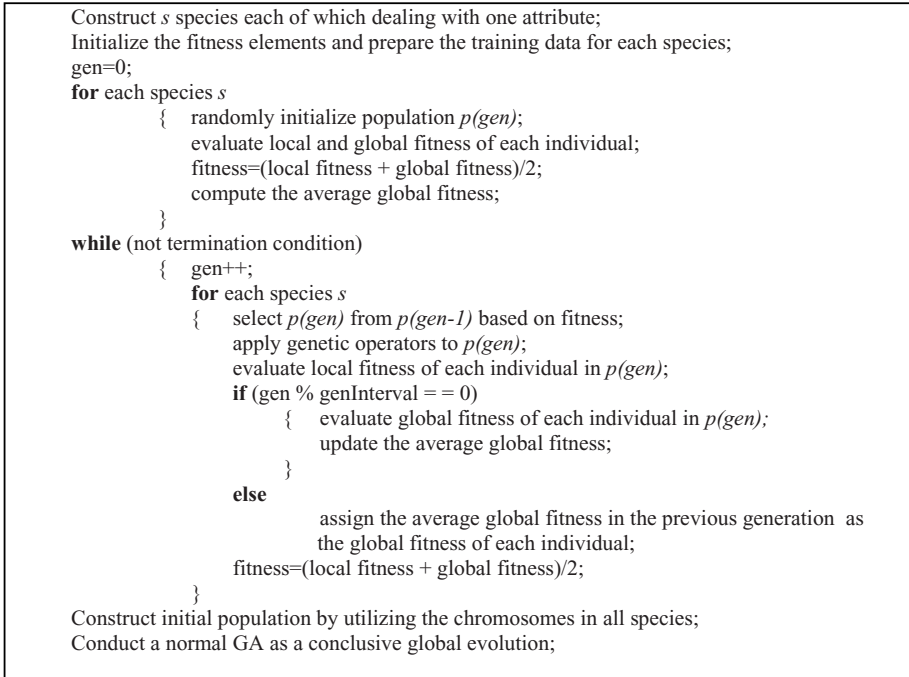


Fig. 4. Pseudocode of ECCGA

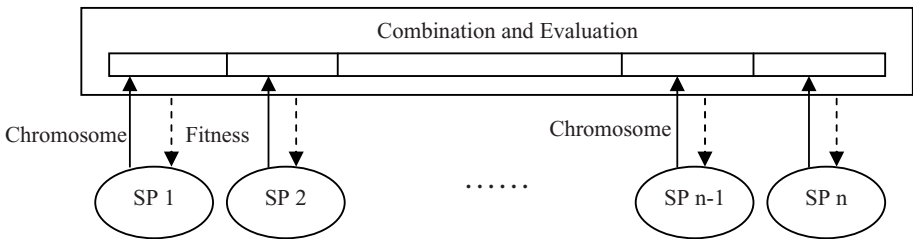


Fig. 5. Chromosome combination and fitness evaluation

4 Experiments and Analysis

We have implemented several classifiers running on three benchmark data sets, i.e., the yeast data, glass data, and housing data. They all are real-world problems and available in the UCI machine learning repository [14]. Table 1 lists the number of instance, attributes, and classes in each data set. The yeast problem predicts the protein localization sites in cells. The glass data set contains data of different glass types. The results of chemical analysis of glass splinters (the percentage of eight different constituent elements) plus the refractive index are used to classify a sample to

Table 1. Datasets used for the experiments

Data Set	No. of Instances	No. of Attributes	No. of Classes
Yeast	1484	8	10
Glass	214	9	6
Housing	506	13	3

be either float processed or non-float processed building windows, vehicle windows, containers, tableware, or head lamps. The housing data concern housing values in suburbs of Boston. Each data set is equally partitioned into two parts. One half is for training, while the other half is for testing.

All experiments are completed on Pentium IV 1.4GHz PCs with 256MB memory. The results reported are all averaged over 10 independent runs. The parameters, such as mutation rate, crossover rate, generation limit, stagnation limit etc., are given under the results. We record the evolution process by noting down some indicative results, which include initial classification rate (CR), generation cost, training time, training CR, and test CR. (Their exact meanings can be found in the notes under Table 2.)

Table 2 shows the performance comparison on the yeast data among ECCGA, CCGA, and GA. The improvement percentage compared to the normal GA is also computed. For ECCGA, three different values for *genInterval* have been tried, i.e. *genInterval*=1, 5, and 10. As ECCGAs involve the additional CGE, both the generations and training time comprise two elements. The latter element indicates the additional generations and training time incurred by CGE. Here, the training time for CCGA and the first training time element for ECCGA are the summary of the time cost of all species in a serial implementation. If the evolution in species is implemented in parallel, or run in a multi-processor system with each computing element running for one species evolution, the training time can be tremendously reduced.

It is found from Table 2 that all ECCGAs and CCGA outperform the normal GA in terms of training CR and test CR, with a significant improvement around 20% - 40%. ECCGAs also outperform CCGA, which means the enhancement really helps achieve better performance. As for the comparison among the three ECCGAs, we find that ECCGA with *genInterval*=1 achieves the best performance and the performance degrades with the increase of the *genInterval*, which means the more frequent global interaction, the better the final results.

It is also found that training time becomes longer in CCGA and ECCGAs, compared to the normal GA. It is because that the chromosome combination and global fitness evaluation takes more time. Considering the larger improvement on the CRs, this is affordable. Furthermore, we can also find that with a larger value of *genInterval*, the training time can be largely reduced with a small degradation in performance. Therefore, we can choose to set a larger value for *genInterval* when the time cost is an issue.

Table 2. Performance comparison on the yeast data – ECCGA, CCGA and GA

Summary	GA	CCGA	ECCGA (<i>genInterval</i> =1)	ECCGA (<i>genInterval</i> =5)	ECCGA (<i>genInterval</i> =10)
Initial CR	0.2356	0.2961	0.2996	0.2911	0.2947
Generations	139.6	91.3	152+104.6	131.5+121.5	106.5+110.5
CPU Time (ms)	592.5	7912.9	10082+454.1	3964.3+486.6	2024.4+469.0
Ending CR	0.3406	0.4147 (21.8%)	0.4771 (40.1%)	0.4493 (31.9%)	0.4156 (22.0%)
Test CR	0.3257	0.3908 (20.0%)	0.4348 (33.5%)	0.4147 (27.3%)	0.3936 (20.8%)

Notes:

1. *mutationRate*=0.01, *crossoverRate*=1, *survivorsPercent*=50%, *ruleNumber*=30, *popSize*=100, *stagnationLimit*=30, *generationLimit*=200;
2. “Initial CR” means the best classification rate achieved by the initial population;
“Generations” means the number of generations needed to reach the stopping criteria;
“CPU time (ms)” means the CPU time cost, and its unit is millisecond;
“Ending CR” means the best classification rate achieved by the resulting population on the training data;
“Test CR” means the classification rate achieved on the test data;
3. The percentage shown for CCGA and ECCGA is the percent improvement over the normal GA.
4. The other tables follow the same notations as this table.

Table 3. Performance comparison on the glass data – ECCGA, CCGA and GA

Summary	GA	CCGA	ECCGA (<i>genInterval</i> =1)	ECCGA (<i>genInterval</i> =5)	ECCGA (<i>genInterval</i> =10)
Initial CR	0.3271	0.3710	0.3673	0.3869	0.3673
Generations	98.6	92.1	104.4+88.2	115.1+85.7	96.5+112.6
CPU Time (ms)	77.5	3162.1	1534.6+41.4	1309.9+66.5	622.6 +88.0
Ending CR	0.5262	0.6327 (20.2%)	0.7374 (40.1%)	0.7159 (36.1%)	0.6972 (32.5%)
Test CR	0.3841	0.4112 (7.1%)	0.4701 (22.4%)	0.4654 (21.2%)	0.4374 (13.9%)

Notes:

1. *mutationRate*=0.01, *crossoverRate*=1, *survivorsPercent*=50%, *ruleNumber*=30, *popSize*=100, *stagnationLimit*=30, *generationLimit*=200.

The experimental results on the glass and housing data sets are reported in Tables 3 and 4 respectively. Similar results as the yeast data are obtained. That is, ECCGAs outperform CCGA, and CCGA outperforms the normal GA, although the improvement percentage over the normal GA is different for each data set. The ECCGA with *genInterval*=1 still achieves the best performance among the three ECCGAs on both data sets.

We have conducted some analysis on the inner mechanisms of the ECCGA through investigating the contribution of local fitness in the fitness function. Figure 6 shows a typical run of the ECCGA (*genInterval*=1) and CCGA in one species on the

yeast data, excluding the CGE stage for the ECCGA. Both ECCGA and CCGA perform similarly in the earlier stage of evolution. However, it is found that CCGA is trapped later and stagnates around the 110-th generation, while ECCGA advances steadily and reaches a higher CR finally. If we recall the design of ECCGA and CCGA, the only difference here is that CCGA only employs global fitness, while ECCGA involves the local fitness in addition. That means the local fitness of ECCGA will give another opportunity to escape from traps in local optima and advance the evolution further.

Table 4. Performance comparison on the housing data – ECCGA, CCGA and GA

Summary	GA	CCGA	ECCGA (<i>genInterval</i> =1)	ECCGA (<i>genInterval</i> =5)	ECCGA (<i>genInterval</i> =10)
Initial CR	0.4553	0.5372	0.5301	0.5364	0.5344
Generations	106.8	80.8	137.8+151.6	88.1+105.1	69.2+93.9
CPU Time (ms)	98.3	7291.6	6168.4+155.4	1950.8+110.9	972.4+93.3
Ending CR	0.6526	0.6941 (6.4%)	0.8506 (30.3%)	0.7700 (18.0%)	0.7364 (12.8%)
Test CR	0.4368	0.4593 (5.2%)	0.5747 (31.6%)	0.4791 (9.7%)	0.4672 (7.0%)

Notes:

1. *mutationRate*=0.01, *crossoverRate*=1, *survivorsPercent*=50%, *ruleNumber*=30, *popSize*=100, *stagnationLimit*=30, *generationLimit*=200.

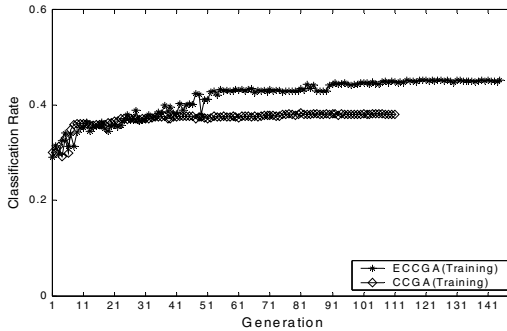


Fig. 6. A typical run of ECCGA and CCGA in species on the yeast data

5 Conclusions

In this paper, the cooperative co-evolution scheme is revisited with a rule-based GA system for pattern classification. It comes with the feature of hybrid artificial intelligence systems, which integrates rule-based inference and genetic algorithms. An innovative approach of ECCGA is proposed to improve further the classification performance. In this approach, the original problem is decomposed into several species, each dealing with one attribute only. The evolution in each species advances together based on the evaluation on both local fitness and global fitness. After all species reach convergence, CGE is used for further global evolution.

The simulation results on three benchmark data sets showed that ECCGA outperforms CCGA and the normal GA. The introduction of local fitness and CGE is helpful to the classification performance. The future work includes further exploration on ECCGA and its performance on tackling newly arriving attributes.

References

1. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Massachusetts (1989)
2. Fidelis, M.V., Lopes, H.S., Freitas, A.A.: Discovering comprehensible classification rules with a genetic algorithm. In: *Proc. of Congress on Evolutionary Computation*, vol. 1, pp. 805–810 (2000)
3. Merelo, J.J., Prieto, A., Moran, F.: Optimization of classifiers using genetic algorithms. In: Patel, M., Honavar, V., Balakrishnan, K. (eds.) *Advances in the Evolutionary Synthesis of Intelligent Agents*, pp. 91–108. MIT press, Mass (2001)
4. Lanzi, P.L., Stolzmann, W., Wilson, S.W.: *Learning Classifier Systems: from Foundations to Applications*. In: Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) *IWLCS 1999. LNCS (LNAI)*, vol. 1813. Springer, Heidelberg (2000)
5. Holland, J.H.: *Adaptation in Nature and Artificial Systems*. Univ. of Michigan Press (1975)
6. Harik, G.R., Goldberg, D.E.: Learning linkage through probabilistic expression. *Computer Methods in Applied Mechanics and Engineering* 186, 295–310 (2000)
7. Chen, Q., Guan, S.-U.: Incremental multiple objective genetic algorithms. *IEEE Trans. on Systems, Man, and Cybernetics, Part B* 34(3), 1325–1334 (2004)
8. DeJong, K.A., Potter, M.A.: Evolving complex structure via cooperative coevolution. In: *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, CA (1995)
9. García-Pedrajas, N., Hervás-Martínez, C., Muñoz-Pérez, J.: Multi-objective cooperative coevolution of artificial neural networks. *Neural Networks* 15(10), 1259–1278 (2002)
10. Potter, M.A., DeJong, K.A.: Cooperative coevolution: an architecture for evolving co-adapted subcomponents. *Evolutionary Computation* 8(1), 1–29 (2000)
11. Guan, S.-U., Zhu, F.: An incremental approach to genetic-algorithms-based classification. *IEEE Trans. on Systems, Man and Cybernetics, Part B* 35(2), 227–239 (2005)
12. Zhu, F., Guan, S.-U.: Ordered incremental training with genetic algorithms. *International Journal of Intelligent Systems* 19(12), 1239–1256 (2004)
13. Potter, M.A., DeJong, K.A.: A cooperative coevolutionary approach to function optimization. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) *PPSN 1994. LNCS*, vol. 866, pp. 249–257. Springer, Heidelberg (1994)
14. Blake, C.L., Merz, C.J.: *UCI Repository of machine learning databases*. Department of Information and Computer Science. University of California, Irvine (1998), <http://www.ics.uci.edu/~mllearn/MLRepository.html>

Learning User Profile with Genetic Algorithm in AmI Applications

Verónica Venturini, Javier Carbó, and José M. Molina

Group of Applied Artificial Intelligence (GIAA)
Computer Science Department
Carlos III University of Madrid
Colmenarejo – Spain

{veronica.venturini, javier.carbo, josemanuel.molina}@uc3m.es

Abstract. In this paper, the algorithm for prediction of user preferences is described through making use of the Heuristic Genetic Algorithm. A Multi-Agent System was used to evaluate our algorithm with the goal of obtaining the user private information to determine what the user requests, in order to offer appropriate services. Additionally, a list of prioritized tasks is generated accordingly, in order to assist the user in making decisions.

1 Introduction

AmI (Ambient Intelligence) applications (for example ALZ-MAS [9], E-Tourism [10], KORE [11]) fit perfectly with the agent paradigm [1]. Multi-Agents Systems support complex interactions between different entities. Therefore, our goal is to develop a Multi-Agent System that will comprise several agents running in real time to give useful information to corresponding users [12]. The methodology used to design the agent system plays a fundamental role because it helps to define ontology, agent's services and roles. In [15] some meta methodologies were analyzed: GAIA, MESSAGES, INGENIAS, ADELFE, PASSI-MAS, RICA, AUML. Ontology definition is one of the steps in any methodology in MAS design. The *ontology* represents the connection between physical information of the real world and conceptual information of the digital world [8]. It allows sharing knowledge between agents in a particular domain. Universe of ontology allocates the reasoning context to agents; a clearly example of ontology is decrypted in OCASS [2]. In the case of an AmI application, agents should reason about knowledge in different possible business domains (conferences, shopping, foreground, etc). Taking into account existing ontologies, our proposed generic ontology described in [8] included the following concepts: location, spatial and temporal region, place, participants, service, product and device to represent the knowledge of several domains.

We are focused in any domain of smart environment (home, offices or public spaces) where people would like to obtain help from agents in order to run their tasks according to their preferences. It is highlighted in this work, that, given the assumption of a closed environment and the possibility of getting user preferences, it is possible to provide suggestions about user activities that would satisfy his interests.

Although, the users may be concerned about privacy issues, since they would not want their private tastes disclosed to others [13]. Thus obtaining explicit explanations about the preferences involved in the immediate past activities, may be very difficult. Deducing information about individual preferences just by observing the immediate past activities is a very challenging task and this is our final goal: the acquisition of a user profile from the observation of decisions taken in past activities. In this way, we intend to determine the relevance of each past decision of the user in order to deduce his implicit preferences.

Approaches to provide a solution to the problem above include the machine learning techniques, such as neural networks [16] and evolutionary computation [7]. These techniques allow the automatic learning of preferences. We have chosen GA (Genetic Algorithm) since they build up gradually the control system by exploring the variations in the interactions between the environment and the agent itself. GAs have undoubted advantages, especially the ability to approximate a real problem by finding the most promising regions of the entire search space. GAs can be more powerful when they fit into a traditional heuristic method (such as a local search) [3]. GA does not only optimize the system accurately and in real time, also it also finds the interests of users according to the previous information. Using GA, personal interests are modelled and then encoded in a chromosome as described in [6]. Cross-operation allows the discovery of new interests by the search of the best parents' interests [3]. The system uses a priori heuristics rules to edit information sequences in which the user has interest. And afterwards analysis of the findings, the system finds user interests and it detects new interests using this information. Given the before explanation, we can say that the goal of using Heuristics of Genetic Algorithms is to get the preferences of the user (since it will be user private information, to obtain what he want and to offer services of his interest), as well as generate a list of prioritized tasks according to their interests to help in making decisions. In order to apply GA, we need to evaluation possible solutions, so we have developed an evaluation method that minimizes the evaluation costs because a complete generation has to be evaluated with the user election.

We combine GA techniques to predict in a similar way to [3] and with the final goal of obtaining immediate interests in an Aml environment as [4] pursued. These can be considered the closest precedents to the innovative approach presented in this paper.

The paper is organised as follows: section 2 about profile managing, section 3 describe the learning feature-based user profile with GA, section 4 provides experimental results and analysis, finally, section 5 concludes.

2 Profile Managing

People have interests and these interests can be in long, middle or short time [4]. We have to work with short time user interests, and therefore is necessary to have two kind of knowledge: one about the domain, and the other, about users interests as in [4]. In this way, the system can provide suggestions to the user about the task that the user could run, according to his immediate past interests. Therefore the Genetic Algorithms uses a tasks list decided from hidden the user preferences that we want to obtain.

There are several ways to acquire such knowledge. A possibility is to ask the user about products and services that are interesting for the user in the current moment. Other way is by using questionnaires to obtain indirectly user preferences. This allows learning about user characteristics (personality, customs, preferences, etc.) and your relation with user decisions in some circumstances. This is the so called knowledge based user profiling, which is static, intrusive and time consuming. While behaviour based user profiling uses the user behaviour itself to build and improve the profile dynamically [14].

Normally the user wants to obtain a quick response from the system, and he thinks that if he has to complete a form, he is losing time. Therefore we will focus on the implicit (behaviour-based) profiling.

3 Learning Feature-Based User Profile with GA

Usually a user has a set of tasks to run at any time. The system sends a list of task to the user’s agent according the user interests. We represent such a task list as:

$$\{T_i\}_{i=1}^n = T_1, T_2, \dots, T_n, \tag{1}$$

where each task is defined as a set of features c_i :

$$T = \{c_i\}_{i=1}^n. \tag{2}$$

Here, each feature represents quality, time, cost or any characteristic that a person takes into account to launch an activity. Normally, users assign to these tasks a priority, possibly based on evaluation of the characteristics described above. We define user interest as a set of preferences

$$P = \{w_i\}_{i=1}^n, \tag{3}$$

where w is the user’s weight over each characteristic (c_1, c_2, \dots, c_n).

The preferences can be evaluated over each task with the importance that the user assigned. Since the preferences are unknown, they have to be obtained by GA. Therefore, the preferences of each task are defined as follows:

$$P(T_i) = \sum_{i=1}^n w_i c_i. \tag{4}$$

We assign characteristic values of the n system tasks and give a value for each one to represent the problem. Then, with the use of Genetic Algorithm individuals are found with characteristic values similar to user characteristics values. We give a priority to each task in the list based on preferences obtained. Using GA user preferences are obtained. Based on this information, a tasks list is built in order to be presented to the user and to assist the user in making decisions. On the other hand, if the user preferences are known (since it will be user private information) we can offer services of his interest.

In the proposed algorithm, we first initialized the values for each characteristic for the user A and the values of each characteristic per task defined at the system. Then the user preferences for each task in the system are calculated and inserted into a list, which is ordered from highest to lowest, according to the preferences of the user. The system generates a random initial population of individuals. We repeat the same process for each individual, i.e. we calculate preferences for the individual for each task and order the task list obtained for the individual. Subsequently, through the evaluation function, GA compares the position of each task in the list user to the list of the individual. The optimal value of the evaluation function is '0', but the algorithm will stop when it has reached to a number of generations previously defined, which, in each iteration, it will undertake the reproduction of individuals, also according to configured parameters (type of reproduction and selection, the probability of mutation, etc.). We have an individual, Equation (5), with a chromosome with as many genes as features to be assessed. Each gene represented w value for each characteristic c_i .

$$Individual \rightarrow \bar{w} = \{w_i\}, \quad (5)$$

where each w_i is encoding with 10 bits in range [0-9].

Before the presentation of a task list for the user, user profiles that define the preferences over each task are built. The task recommendation process can start once the profiles are acquired, with Active user A given, a set of individuals with similar profiles to A may be found. We use GA to obtain evolutionarily the weights that each user assigned to his preferences, to acquire a priority task list. For each individual, we make a prioritization of tasks and then we use the "distance" to select which of the individuals selected is the closest to a user's profile A. We choose a Phenotypic Distance to measure what looks like the solutions generated by each of the individuals to compare. Comparing the task list generated by each individual and the activities selected by the user can then be carried out by a modified Euclidean distance, which takes into account various features c_i . We assume that the application of heuristic GA has evolved values of features for the preferences of tasks, and it has obtained the following list of tasks, thanks to that individual. Therefore, Euclidean Function (A, I), showed in Equation (6), is the distance between user A and individual I, in which the summary from $i=0$ to n can be made, since the system has n tasks. Fitness function is calculated in relation at task position in the list.

$$Euclidean - Distance(A, I) = \sum_{i=1}^n \sqrt{Diff(jAi, jIi)}^2, \quad (6)$$

where

A is the task list that the user chose,

I is a task list given by selection process, also $I \neq A$

i is a task, where there *Profile(A,i)* and *Profile(I,i)*

n is the quantity of tasks that the user can do

j indicates task position in the priority task list

diff means the differences in the i -th task position between user A and individual I

The evaluation function, computes the distance between each individual in the population (every task list proposed) and the user list of tasks . The distance between

the two task lists is “0” when the optimal solution is found, namely a list of tasks identical to the user’s choice.

For example, we suppose that are going to evaluate the following features: quality, cost, time and distance. The valuation of each feature to the User A is a vector with the follow values: $\bar{w} = \{9, 0, 8, 5\}$ these values are quality, cost, time and distance respectively. For each task a value of the characteristic is introduced in the system (for example, Task 1 with $q=2, c=9, t=9$ and $d=7$). Once these parameters are presented to the GA as 'pre-process', the individuals who belong to the profile are generated, by calculating the difference between the preferences of the user tasks (user A) and the preferences of tasks of each individual. Subsequently, with the preferences obtained by each task, GA generates a list of tasks for the individual that was ordered in accordance with preferences values for each task, and it finally applies the fitness, to evaluate how different is the list of the user to a list obtained from the best individual in the population. The difference approaches to a value "0" for similar lists. Generated task lists are shown in Table 1, according to individuals generated randomly. Note that for each task we have obtained a preference value, and with it we have been able determine priorities.

Table 1. Skills learned by individuals I1, I2 and I3 in relation to the user A

User A	I1	I2	I3
T ₁	T ₁	T ₁	T ₁
T ₃	T ₂	T ₂	T ₃
T ₂	T ₃	T ₃	T ₂
T ₄	T ₅	T ₄	T ₄
T ₅	T ₄	T ₅	T ₅
Difference between A/I	4	2	0

Since we want to minimize the difference between the task lists it can be concluded based on Table 1 that the optimal result corresponds to the individual I3 (the task list identical to that of the user A), followed by suboptimal result corresponding to the individual I2, and the worst result obtained for the individual I1.

4 Experiments

In order to define a problem with GA it is necessary to configure a set of parameters, represented by variables x_1, x_2, \dots, x_n of objective function and constraints, as well as its data type and the minimum and maximum values allowed; the optimized target function $f(x)$; a set of restrictions $g(x)$; and finally, the roles of pre-processing and post-processing, which are functions that are evaluated only at the beginning or end of the implementation of the algorithm, respectively.

The following parameters are required by the GA: Population length, Generation number, Kind of reproduction, Probability of Reproduction, Selection, Probability of Mutation, Elitism (Individual to preserve), Refresh period f the population, and Percentage of population cooled.

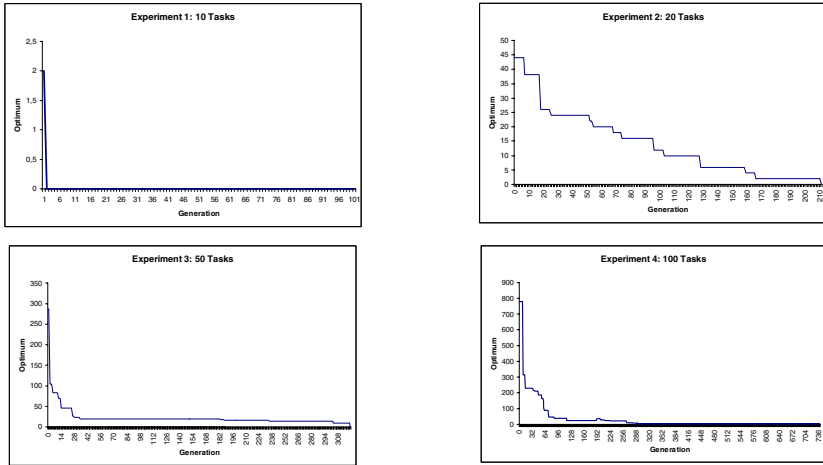


Fig. 1. Evolution of GA with 10, 20, 50 and 100 tasks

We worked on an initial population of 6 individuals. An elitist GA has been chosen to be developed that only preserves 1 individual in the last population and with a crossing point. The selection was made by the selection training, the probability of mutation was 0.005, and the reproduction and selection are 0.89 and 0.85 respectively. The convergence concept is related with the progression to uniformity: generation had converged when at least the 95% of the individuals in the population share the same value for this gene [5]. We assumed that the population is converging when all the genes had converged. The population had evolved to the global maximum.

There have been tests where the task list system was ranging from 5, 10, 20, 50 to 100 tasks (Fig. 1). For each case we ran the program with 1000 generations and we analyzed the convergence of GA. The values obtained are shown in Fig. 2. Of the total number of tests performed, in 99% of cases the trial leads to the optimal solution, 0.1% were unsuccessful and the remainder (0.9%) solutions were sub-optimal.

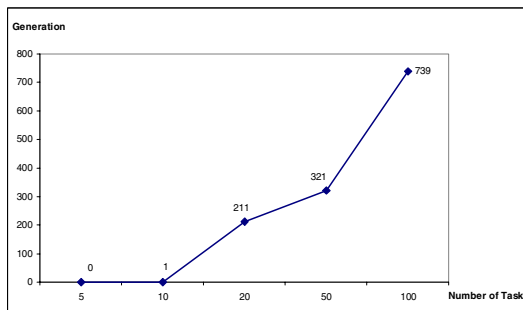


Fig. 2. Convergence of GA in relation to number of task

5 Conclusions

The interest of researchers in Ambient Intelligence has been increasing in the last years. It seems to be one of the new main challenges of Computer Science applications and particularly of Artificial Intelligence, but it still has to solve some key problems such as predicting user's preferences in an unobtrusive way.

In this paper, we used a Heuristic Genetic Algorithm to know what the user really wants. Specifically, we generated a list of prioritized tasks according to the user assumed preferences. If the context-aware multi-agent system that has to predict user preferences has a small set of tasks, GA may then be not justified because of the computational cost, but if we assumed enough number of tasks and of characteristics, then GA application would be justified.

Application of Genetic Algorithms allows us to get the implicit user preferences on some features or characteristics of tasks from an ordered list of these tasks. We have an individual with a chromosome with so many genes as features are to be assessed. For each new individual generated, we compute the corresponding prioritization of tasks and then, through the evaluation function, GA compares the position of each task in the list user to the list of the individual. Therefore, the optimal value of the evaluation function that compares them is then zero. GOAL software was used to run the GA¹ that obtains evolutionarily the weights that each user assigns to his preferences from this comparison of priority task list to select which of the individuals is the closest one to the user preferences.

Finally, we analyzed different executions of this GA with Goal software to specify in which number of generation the algorithm converges when the number of tasks increases. And we concluded that this method is more effective when the number of tasks to be considered is comprised between 10 and 100.

References

1. Fuentes, V., Sánchez, N., Carbó, J., Molina, J.M.: Generic Context-Aware BDI Multi-Agent Framework with GAIA methodology. In: International Workshop on Agent-Based Ubiquitous Computing, Int Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2007), Hawaii, USA, May 14-18 (2007)
2. Kim, Y., Uhm, Y., Hwang, Z., Lee, M., Kim, G., Song, O., Park, S.: A Context-Aware Multi-agent Service System for Assistive Home Applications. School of Electrical and Electronics Engineering, Chung-Ang University, 221, Heukseok-dong, Dongjak-gu, Seoul 156-756, Korea, pp. 732–745
3. Chen, X., Cheng, L., Huo, J., Huo, Y., Wang, Y.: Combining Prediction through Heuristic Genetic Algorithm on Intelligence Service System, Networking, Sensing and Control. In: 2004 IEEE International Conference, March 21-23, vol. 1, pp. 407–411 (2004)
4. Nijholt, A.: Capturing Immediate Interests in Ambient Intelligence Environments. In: IADIS International Conference Intelligent Systems and Agents 2007, Lisbon, Portugal, July 3-5 (2007)
5. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, Part I. Genetic Algorithms, 3rd edn., pp. 11–88. Springer, London (1996)

¹ <http://www.geocities.com/geneticoptimization/GOAL.html>

6. Ujjin, S., Bentley, P.J.: Learning user preferences using evolution, University College London. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL 2002), Singapore (2002)
7. Berlanga, A., Isasi, P., Segovia, J.: Interactive evolutionary computation with small population to generate gestures in avatars. In: Spector, L., Goodman, E.D., Wu, A., Langdon, W., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp. 823–828. Morgan Kaufmann, San Francisco (2001)
8. Fuentes, V., Carbó, J., Molina, J.M.: Heterogeneous Domain Ontology for Location based Information System in a Multi-Agent Framework. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 1199–1206. Springer, Heidelberg (2006)
9. 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* 44(2), 382–396 (2008)
10. Paganelli, F., Bianchi, G., Giuli, D.: A Context Model for Context-Aware System Design Towards the Ambient Intelligence Vision: Experiences in the e-Tourism Domain. In: Proc. of 9th ERCIM Workshop User Interfaces For All, Florence, Italy. Electronics and Telecommunications Department, University of Florence (2006)
11. Bombara, M., Calí, D., Santero, C.: KORE: A Multi-Agent System to Assist Museum Visitors. In: Workshop on Objects and Agents (WOA 2003), Villasimius, CA, Italy, pp. 175–178 (2003)
12. Bajo, J., Julian, V., Corchado, J.M., Carrascosa, C., De Paz, Y., Botti, V., De Paz, J.F.: An Execution Time Planner for the ARTIS Agent Architecture. *Engineering Applications of Artificial Intelligence* 21(8) (2008)
13. Masthoff, J., Vasconcelos, W., Aitken, C., Correa da Silva, F.: Agent-Based Group Modelling for Ambient Intelligence. In: AISB Symposium on Affective Smart Environments, Newcastle, UK (2007)
14. Fuentes, V., Sanchez, N., Carbó, J., Molina, J.M.: Reputation in user profiling for a Context-Aware Multiagent System. In: Fourth European Workshop on Multi-Agent Systems EUMAS 2006, Lisboa, Portugal, December 14-15. GIAA, Computer Science Dept., Carlos III University of Madrid (December 2006)
15. Bernon, C., Cossentino, M., Pavon, J.: Agent Oriented Software Engineering. *Knowledge Engineering Review* 20(02), 99–116 (2005)
16. Morariu, N., Vlad, S.: Using Pattern Classification and Recognition Techniques for Diagnostic and Prediction. *Advances in Electrical and Computer Engineering* 7(1) (2007)

Unsupervised Genetic Algorithm Deployed for Intrusion Detection

Zorana Banković¹, Slobodan Bojanić¹, Octavio Nieto¹, and Atta Badii²

¹ ETSI Telecomunicación, Universidad Politécnica de Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain

{zorana,slobodan,nieto}@die.upm.es

² Intelligent Media Systems & Services Research Centre School of Systems Engineering, White Knights campus, University of Reading, Reading, RG6 6AY, United Kingdom
atta.badii@reading.ac.uk

Abstract. This paper represents the first step in an on-going work for designing an unsupervised method based on genetic algorithm for intrusion detection. Its main role in a broader system is to notify of an unusual traffic and in that way provide the possibility of detecting unknown attacks. Most of the machine-learning techniques deployed for intrusion detection are supervised as these techniques are generally more accurate, but this implies the need of labeling the data for training and testing which is time-consuming and error-prone. Hence, our goal is to devise an anomaly detector which would be unsupervised, but at the same time robust and accurate. Genetic algorithms are robust and able to avoid getting stuck in local optima, unlike the rest of clustering techniques. The model is verified on KDD99 benchmark dataset, generating a solution competitive with the solutions of the state-of-the-art which demonstrates high possibilities of the proposed method.

Keywords: intrusion detection, genetic algorithm, unsupervised.

1 Introduction

Software applications (both commercial and research ones) that deploy a machine learning technique are considered to be among the emerging technologies that have demonstrated compelling value in enhancing security and other related data analysis. Machine learning-based applications use complex mathematical algorithms that scour vast amounts of data and categorize them in much the same fashion as a human would, or at least according to the categorization rules set by a human. Yet, they are able to examine far more data in less time and more comprehensively than a human can, highlighting those events that appear suspicious enough to warrant human or automated attention.

Genetic Algorithms (GA) [1], in particular, offer certain advantages over other machine learning techniques, namely:

- GAs are intrinsically parallel, since they have multiple offspring, they can explore the solution space in multiple directions at once. If one path turns out to be a dead end, they can easily eliminate it and continue work on more promising avenues.

- Due to the parallelism that allows them to implicitly evaluate many schemas at once, genetic algorithms are particularly well-suited to solving problems where the space of all potential solutions is truly huge - too vast to search exhaustively in any reasonable amount of time, as network data is.
- Working with populations of candidate solutions rather than a single solution, and employing stochastic operators, to guide the search process permit GAs to cope well with attribute interactions and to avoid getting stuck in local maxima, which together make them very suitable for dealing with classifying rare classes, as intrusions are.
- A system based on GA can easily be re-trained. This property provides the adaptability of a GA-based system, which is an imperative quality of an intrusion detection system bearing in mind the high rate of new attacks emerging.

Our GA forms clusters of similar data and the principal idea is to allow the security expert to be the final arbiter of what is and is not an actual threat. In addition, after forming the initial clusters and assigning them the corresponding labels (also performed by the security expert), when classifying new events that do not correspond to any of the existing clusters, the network administrator is the one to decide whether the event corresponds to an existing cluster or whether a new cluster should be established. Our GA tool assists him by providing the level of proximity between the new event and the existing clusters. In this way, the system continuously updates itself, at the same time learning more about its environment. Thus, the main objective of this tool is to couple expert knowledge and GA data analysis technologies. An important point that should be emphasized is that our algorithm is unsupervised, i.e. it does not require training data to be labelled. In this way the extensive engineering and error-prone job of labelling network data is avoided.

In the recent past, there has been a great deal of criticism towards applying machine learning techniques in network security. The critics refer mainly to two issues:

1. Machine learning techniques do not exhibit broad applicability, i.e. they are not able to detect attacks for which they were not trained to detect
2. Most of machine learning is focused on understanding the **common** cases, but what is wanted is to find the **outliers** (that are generally not so common)

We have tried to mitigate these issues by coupling the GA with expert knowledge. In this way the GA process of learning is “revised” and moves towards the desired direction. This would not be possible without the intrinsic properties of GA, namely a high level of adaptability to the environment changes. Moreover, it is demonstrated that GAs cope well with classifying rare or **uncommon** cases. Finally, the algorithm is only one part within the broader intrusion detection [2] framework, while the critics mostly concern cases where a machine learning technique is deployed without any additional support.

The rest of the work is organized as follows. Section 2 presents related work. Section 3 details the implementation of the system. Section 4 gives the results of initial testing, while Section 5 presents novelty and advantages of the approach, draws conclusions and suggests future strategies.

2 Related Work

Most of the machine-learning techniques deployed for intrusion detection are supervised, as these techniques exhibit higher level of accuracy than the unsupervised ones. However, they have an important deficiency because they operate on labelled data. Labelling network data is time-consuming and error-prone process whose possible errors may affect negatively on the level of accuracy of deployed technique.

On the other hand, unsupervised techniques as K-means clustering, although do not suffer from the problem of data labelling, exhibit other deficiencies. The number of clusters in K-means clustering has to be determined a priori and cannot be changed during the process. This is an important deficiency, as the optimal number of clusters in an environment that is constantly changing, as network environment is, is hard to determine. Moreover, this technique is known to get stuck at sub-optimal solutions depending on the choice of the initial cluster centres. k NN algorithm suffers from similar problem, that consists in determining the optimal value of k a priori without the possibility of changing it on the fly. Again, this does not provide the level of flexibility that is necessary in dynamic environments. Furthermore, its accuracy can be severely degraded by the presence of noisy or irrelevant features, meaning that it is not robust enough.

On the other hand, GAs avoid getting stuck in local optima due to working with populations of candidate solutions rather than a single solution, and employing stochastic operators. For the same reason, they are very robust which helps them in dealing with noisy network data. Furthermore, our design does not assume that the number of clusters has to be known a priori which provides higher level of flexibility. Thus, our approach offers true prospects for gaining higher performances when dealing with network data.

3 System Implementation

The process of producing security intelligence is comprised of the following five steps depicted in Fig.1 below:

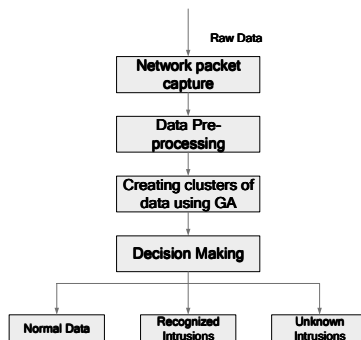


Fig. 1. Detection process flow

The algorithm whose software implementation is presented in the following uses the GALib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology [3].

3.1 Designate Data

The input to the algorithm is raw data. The data is further captured by the Wireshark Network Protocol Analyzer [4] which provides the capture data in .csv format as output. Captured data completely defines particular connection, as it contains information about source and destination IP address, ports and payload data.

3.2 Data Pre-processing

In this step, each attribute, i.e. each feature of the pre-determined model of network connections, gets its value based upon the raw data obtained in the previous step. Thus, the output of this step consists of multi-dimensional event vectors that represent the model of network traffic. In this way every network connection is represented as a vector.

3.3 Model Training

This is the process of forming clusters of similar data. It is assumed that the number of clusters is not known *a priori*. This provides a higher level of flexibility as an unknown attack does not necessarily have to be joined to an existing cluster, yet it can form a new cluster. Initial clusters are formed in the process of training by evolving a genetic algorithm. The genetic algorithm type deployed is Incremental GA, i.e. it uses overlapping populations, but with very little overlap (only one or two individuals get replaced in each generation). The replacement scheme used is WORST, i.e. the worst-performing individuals are replaced with the new ones in each iteration [3].

The pseudocode of the training process is given in Fig.2 below. The process of training is constantly repeated at certain moments of time. Each time different dataset is used for training.

```
Set the number of individuals, the number of generations, and the
number of individuals to be changed in each generation, mutation and
crossover rate values;
Initialize the population;
For each individual from the population
    For each connection from the dataset
        Calculate fitness function according to the formula;
        Breed the population using mutation and crossover;
        Substitute the worst individuals with the new generated
ones;
    Repeat the previous steps for the specified number of
generations;
```

Fig. 2. Pseudocode of the Training Process

A further explanation of each step follows:

Individual Representation and Population Initialization. Genomes of the GA are two-dimensional matrices, where each row represents a cluster centre. GALib provides resizable two-dimensional genomes making in this way the idea of unknown number of clusters feasible [3]. Each element of a cluster centre corresponds to a feature of the pre-determined model of network traffic. Thus, the size of a row is equal to the pre-determined number of network features that describe the network model deployed.

At the beginning of the first execution of the training process, each element of each genome of the population is initialized to a random float number. The initial population of every next training process will be the output of the previous one. The size of the population and the number of generations are set by the user.

Performance Measurement (Fitness Function). For each chromosome, the centres encoded in it are first extracted, and then a partition is obtained by assigning the points to a cluster corresponding to the closest centre. The distance between the points is computed as the Euclidian distance [5]:

$$d(X, C) = \sqrt{\sum_{i=1}^n (x_i - c_i)^2} \quad (1)$$

where c represents the vector of cluster centre whose dimensions are $1 \times n$, n is the number of predetermined features used to represent the model of network traffic, and x represents the current packet converted to an event vector mentioned above. If all the distances to the corresponding centres are greater than the predefined threshold, then a new centre is established. The cluster centres encoded in a chromosome are then replaced by the centroids of the corresponding clusters. The cluster centroid is the average vector of all the vectors that belong to a cluster.

Given the above partition, and the number of clusters, the value of the cluster validity index is computed. The fitness of a chromosome is then defined as a function of the corresponding cluster validity index. The Davies-Bouldin (*DB*) index [6] is selected because of the following advantages over other measures:

1. Stability of results: this index is less sensitive to the position of a small group of data set members (so called outliers) than other measures, such as for example, the Dunn's index.
2. In the case of more than 2 clusters and the need to rank them, some measures (for example the Silhouette index) behave unpredictably, whereas the expected behaviour of the Davies-Bouldin index in these cases is good.

DB index is a function of the ratio of the sum of *within-cluster scatter* to *between-cluster separation*. The scatter within the i th cluster is computed as:

$$S_i = \sqrt{\frac{1}{|C_i|} \sum_{x \in C_i} \|x - c_i\|^2} \quad (2)$$

where C_i and $|C_i|$ represent the i th cluster number of the elements that belong to the i th cluster respectively. $\|x - c_i\|$ is the distance between the centre and an element from the same cluster. It is calculated as the Euclidean distance given in the previous

formula. The distance d_{ij} between two clusters is considered to be the Euclidean distance between the corresponding cluster centres. Then, we compute:

$$R_i = \max_{j, j \neq i} \left\{ \frac{S_i + S_j}{d_{ij}} \right\} \quad (3)$$

The Davies-Bouldin DB index is then computed as:

$$DB = \frac{1}{K} \sum_{i=1}^K R_i \quad (4)$$

where K is the number of different clusters. The objective is to minimize the DB index for achieving proper clustering. Therefore, the fitness of chromosome j is defined as $(1/DB_j)$, where DB_j is the Davies-Bouldin index computed for this chromosome. The maximization of the fitness function will ensure minimization of the DB index.

Selection. Standard roulette-wheel selection is deployed. The possibility of selecting an individual is directly proportional to its fitness value [1].

Crossover. Standard one-point crossover is deployed. The possibility of crossover can be set by the user [1].

Mutation. A swap-mutator is deployed to carry out the process of mutation. The possibility of mutation can be set by the user [1].

The deployed selection, mutation and crossover operator are built-in the GALib package [3].

3.4 Cluster Labelling and Decision Making

After the process of establishing the initial clusters, the next step is to classify network events. Before performing the process of classification, the clusters are labelled by the security expert. According to his previous knowledge, security expert may label the clusters as intrusive, normal or previously unseen that need further investigation. We have opted for this approach as it is the most appropriate solution for a system that is to be employed in a real-world application.

During the process of classification, the Euclidean distances between the event and the existing clusters are calculated. The event is assigned to belong to the closest cluster. However, if all the calculated distances surpass the predetermined threshold value (that determines the maximum possible distance between a network event and its corresponding cluster centre, meaning that the packet does not correspond to any of the existing clusters), the security expert is called upon to be the final arbiter. Our GA tool assists him by providing the level of proximity between the new packet and the existing clusters. The security expert renders the final decision as to whether the event corresponds to an existing cluster or a new cluster should be established, meaning that the event has been unknown up to this moment. In this way, the system continuously updates itself, at the same time learning more about its environment.

4 Initial Testing and Results

Testing of the implemented algorithm was carried out on the benchmark KDD99 dataset [7]. As KDD set has its own pre-defined features [7], the step of data-preprocessing explained above is skipped. GA parameters of initial testing are the following ones: initial population contains 100 individuals, 50 generations are evolved during the process of evolution, with the probabilities of crossover and mutation operator 0.9 and 0.1 respectively. For the purpose of initial testing, the process of labeling is performed using the existing labels of the dataset: a cluster gets the label of the group (normal or one of four types of attacks) whose number of individuals that pertain to the group is higher than the rest.

The KDD dataset has been found to have a number of drawbacks [8], [9]. Thus the testing results do not reflect the behaviour of the algorithm in a real-world environment. Moreover, current testing is not performed in a manner the system is going to be utilized, i.e. all the tests were carried out on the initial clusters without performing any re-clustering by a security expert. The process of cluster labeling is also performed in a different way. Testing results, however, reflect a high capability that the algorithm exhibits in distinguishing intrusive connections from the non-intrusive ones. The overall detection rate of 91% with a false-positive rate of 0.5% ranks this system as one of the best-performing unsupervised methods in the current state-of-the-art. Due to the skewed distribution of different attack groups (number of DoS attacks highly surpasses the number of the rest), deployed labeling results in assigning DoS label to all of the attacks groups. Furthermore, this leads to not having alteration of the final result (expressed in the terms of detection and false-positive rate) while changing different parameters of the proposed algorithm, such as population size or possibilities of the operators, or different fitness functions (Dunn index [6] or I -index proposed in [10]).

The observed drawback consists in assigning some of the attacks into the clusters that represent different attacks groups, or declare them as normal. The latter is often the case with U2R (User-to-root) and R2L (Remote-to-Local) types of attacks for the dataset. This occurs partially due to the known drawback of the dataset, i.e. the dataset is proven to possess connections with absolutely the same or very similar feature values, but with different labels [8]. However, all the drawbacks are expected to be mitigated through the influence of the security expert.

5 Conclusions and Future Strategies

The described GA design represents a novelty in the network security field. Most of the existing applications of GAs for intrusion detection assume supervised learning where the training dataset needs to be labelled. Our design of the algorithm is unsupervised, so we have avoided the process of labelling the data which can be time-consuming and error-prone. On the other hand, various clustering algorithms have been deployed for classifying network events, but none of them was based on genetic algorithms. Hence, the algorithm as deployed presents a novelty in the network security field. Moreover, our system can cooperate with a security expert providing in that way higher accuracy and adaptability to environmental changes.

Initial testing reflects a high capability that the algorithm exhibits in distinguishing intrusive connections from the non-intrusive ones. In the near future, the algorithm will be tested in a real-world environment as a part of the final product of the Intrusion Detection Project [2].

Strategies for the continuation and the improvement of the work presented here are numerous and concern various aspects of the algorithm. The question of data pre-processing still remains unresolved. A promising solution could be to deploy the idea presented in PAYL [11]. More flexibility could be given to the algorithm by providing different options for performance measurement, selection, crossover and mutation. Moreover, there is a great possibility of using different standard genetic operators for selection (rank, tournament, deterministic sampling) or crossover (two-point, even-odd) and mutation (random flip) or even developing custom operators [3], [1].

Acknowledgements. This work has been partially funded by the Spanish Ministry of Education and Science under the project TEC2006-13067-C03-03 and by the European Commission under the FastMatch project FP6 IST 27095.

References

1. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Longman (1989)
2. <http://www.fastmatch.org>
3. GALib A C++ Library of Genetic Algorithm Components, <http://lancet.mit.edu/ga/>
4. <http://www.wireshark.org> (accessed, 2007)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley InterScience, Chichester (2000)
6. Bolshakova, N., Azuaje, F.: Cluster Validation Techniques for Genome Expression Data. Signal Processing 83, 825–833 (2003)
7. KDD Cup 1999 data (October 1999) (accessed, 2006/2007), <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
8. Bouzida, Y., Cuppens, F.: Detecting Novel and Known Intrusions, IFIP/SEC 2006. In: 21st IFIP TC-11 International Information Security Conference Karlstad University, Karlstad, Sweden (May 2006)
9. McHugh, J.: Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Library. ACM Transactions on Information and System Security 3(4), 262–294 (2000)
10. Bandyopadhyay, S., Maulik, U.: Nonparametric genetic clustering: comparison of validity indices. IEEE Transactions on Systems, Man, Cybernetics, Part C (2001)
11. Wang, K., Stolfo, S.J.: Anomalous Payload-based Network Intrusion Detection. In: Jons-son, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 203–222. Springer, Heidelberg (2004)

Automatic Neural Net Design by Means of a Symbiotic Co-evolutionary Algorithm

Elisabet Parras-Gutierrez, Víctor M. Rivas, and Maria Jose del Jesus

Department of Computer Sciences, University of Jaen
{eparrasg,vrivas}@vrivas.es, mjjesus@ujaen.es

Abstract. One of the most important issues that must be taken in mind to optimize the design and the generalization abilities of trained artificial neural networks (ANN) is the architecture of the net. In this paper Symbiotic_RBF is proposed, a method to do automatically the process to design models for classification using symbiosis. For it, there are two populations who evolve together by means of coevolution. One of the populations is the method EvRBF, which provides the design of radial basis function neural nets by means of evolutionary algorithms. The second population evolves sets of parameters for the method EvRBF, being every individual of the population a configuration of parameters for the method. Thus, the main goal of Symbiotic_RBF is to find a suitable configuration of parameters necessary for the method EvRBF, which is adapted automatically to every problem.

Keywords: neural networks, evolutionary algorithms, data sets, coevolution.

1 Introduction

In this paper, a method to automatically design models for classification using symbiosis is proposed. In order to do this, the so called Symbiotic_RBF uses two populations that evolve together by means of coevolution.

From a biological point of view, coevolution is defined as "the evolution of two or more species that do not cross among them but that possess a narrow ecological relation, across reciprocal selection pressures, the evolution of one of the species depends on the evolution of other one" [19]. There are three main kinds of coevolution: Competitive (Parasitic), Cooperative (Symbiotic), and Exploitation.

In competitive evolution, the organisms, belonging to the same species or not, compete for the same resources. On the other hand, exploitation takes place when the presence of a species stimulates the development of another, while the presence of this one disables the development of the first one. And finally, the interactions among two species are cooperative when both (of them) benefit from this interaction. Symbionts organisms are extreme cases in which two species are so intimately integrated that they act as an alone organism.

In this paper symbiotic coevolution is used, thus, every population contributes to the evolution of the other and vice versa.

First population is composed of many instances of the method EvRBF, which provides the design of radial basis function neural nets by means of evolutionary

algorithms. The second population evolves sets of parameters for the method EvRBF, being every individual of the above mentioned population a configuration of parameters for it.

Thus, for every given problem, Symbiotic_RBF searches automatically the optimal configuration of parameters that the method EvRBF should use. Symbiosis shows then its ability to increase the usability of traditional methods, one of the key issues on future trends in data mining.

2 Related Work

The design of optimal artificial neural networks (ANNs) is a key issue in the study of ANNs [21] and both evolutionary and non-evolutionary algorithms have been developed to face this task. Most non-evolutionary algorithms can be classified into constructive and destructive methods. Constructive algorithm start with a minimal network (network with minimal number of hidden layers, nodes, and connections) and add new layers, nodes, and connections during training; destructive algorithms do the opposite, i.e., start with the maximal network and delete unnecessary layers, nodes, and connections during training.

Harpham et al. reviewed in [6] some of the most well known methods that apply evolutionary algorithms to RBFNN design. They concluded that, in general, methods tend to concentrate in only one aspect when designing RBFNN. Nevertheless, there also exist methods intended to optimize the whole net, such as [9] for Learning Vector Quantization (LVQ) nets or [3] for multilayer perceptrons.

Chronologically, the first papers [4, 20] used binary codification and were constricted by the number of hidden neurons, that had to be set a priori. While Carse [4] works with a population of nets, Whitehead [20] works with a population of neurons (thus, only one RBFNN was build and evaluated) that compete and cooperate to find the optimal net.

Subsequent papers [2] presented algorithms based on real number representation. Again, the imposition of a limit to the number of neurons, optimization of only the centers for RBF, and a badly defined penalization term for the number of times an individual can reproduce, are the main drawbacks of this algorithm.

Recent applications try to overcome the disadvantages of the preceding methods. Rivera [16] made many neurons compete, modifying them by means of *fuzzy evolution*, i.e., using a fuzzy rules table that specifies the operator that must be applied to a neuron in order to improve its behavior.

Most recent methods still face single aspects when configuring a net. Thus, Ros et al.'s method [17] automatically initializes RBFNN, so that finds a good set of initial neurons, but relies in some other method to improve the net and to set its widths.

Different kinds of coevolution have also been used to design ANN, as can be found in literature. Paredis [10] competitively coevolved the weights of ANNs of fixed topology for a synthetic classification task. After this, Paredis [11] proposed a general framework for the use of coevolution to boost the performance of genetic search, combining coevolution with life-time fitness evaluation.

Potter and De Jong [13] proposed Cooperative Coevolutionary Algorithms (CCGAs) for the evolution of ANNs of cascade network topology. Coevolution occurs by decomposing the problem into interacting subtasks.

Barbosa [1] employs a competitive coevolutionary genetic algorithm for structural optimization problems using a game-theoretic approach. The first player, the designer of a mechanical structure, tries to minimize the Compliance (a measure of the overall deformability under a specific load). The second player, nature, challenges the designer's constructions by maximizing the compliance.

Schwaiger and Mayer [18] employed a genetic algorithm for the parallel selection of appropriate input patterns for the training datasets (TDs). After this, Mayer [7], in order to generate coadapted ANNs and TDS without human intervention, investigated the use of symbiotic (cooperative) coevolution. Independent populations of ANNs and TDSs were evolved by a genetic algorithm, where the fitness of an ANN was equally credited to the TDS it had been trained with.

3 System Overview

The optimal design of a net involves the establishment of its structure as well as the weights of the links. In this sense, RBFNN represent a special kind of NN since, once the structure has been fixed, the optimal set of weights can be computed. Symbiotic_RBF is based on EvRBF, an evolution method developed to automatically design asymmetric RBF.

The main goal of Symbiotic_RBF is to find a suitable configuration of parameters necessary for the method EvRBF, which is adapted automatically to every problem. In the section 3.1 and 3.2 both methods are described detailed.

3.1 The EvRBF Algorithm

EvRBF [14, 15] is a steady state evolutionary algorithm that includes elitism. It follows Pittsburgh scheme, in which each individual is a full RBFNN whose size can vary, while population size remains equal.

EvRBF codifies in a straightforward way the parameters that define each RBFNN, using an object-oriented paradigm; for this reason, it includes two operators for recombination and four for mutation that directly deal with the neurons, centers and widths themselves. Recombination operators (X_FIX and X_MULTI) interchange information between individuals, trying to find the building blocks of the solution. In the other hand, mutation operators (centers and width modification: C_RANDOM and R_RANDOM) use randomness to increase diversity generating new individuals so that local minima can be avoided. Furthermore, EvRBF tries to determine the correct size of the hidden layer using the operators ADDER and DELETER to create and remove neurons, respectively.

The exact number of neurons affected by these operators (except ADDER) is determined by their internal application probabilities.

EvRBF incorporates tournament selection for reproduction. The *fitness function* measures the generalization capability of each individual as the percentage of samples it correctly classifies. When comparing two individuals, if and only if both two individuals have exactly the same error rate, the one with less neurons is said to be better. The LMS algorithm is used in order to train individuals, and to obtain the weights of links.

3.2 Description of Symbiotic_RBF

Every individual of the algorithm Symbiotic_RBF is a string composed of 0 and 1 representing a set of 8 parameters for the method EvRBF, as the size of the population, the size of the tournament for the selection of the individuals or the maximum number of generations. This sequence of bits is organized as follows:

- 7 bits for the size of the population (a value between 2 and 100)
- 2 bits for the size of tournament size (between 2 and 5)
- 6 bits for the number of generations (between 0 and 50)
- 10 bits for the percentage of the initial limit of neurons (between 0.1 and 1.0.)
- 10 bits to represent the percentage of training patterns used for validation (between 0.1 and 0.5)
- 10 bits for the rate of replacement (between 0.1 and 1.0.)
- 10 bits for the rate of the cross over (between 0.0 and 1.0)
- 10 bits that will represent the rate of mutation (between 0.0 and 1.0.)

In order to set the fitness of an individual, its chromosome is decoded into the set of parameters it represents. Then, these parameters are used to perform a complete execution of EvRBF, and the percentage of validation patterns correctly classified by the best net found is used as fitness for the individual.

Finally, the parameters that Symbiotic_RBF needs to perform its own evolving are: size of the population, size of tournament, rate of replacement, rate of mutation and number of executions.

4 Experimental Results

The system Symbiotic_RBF has been evaluated across three data sets from UCI data set repository¹: ionosphere (351 instances, 34 features, 2 classes), Sonar (208 instances, 60 features, 2 classes) and Wdbc (570 instances, 30 features, 2 classes). For every database, 7 different sets of data have been generated using various feature selection methods, so that Symbiotic_RBF has been tested over 21 different classification databases.

Feature selection algorithms used in this work included both filter and wrapper methods. Briefly, FS-LVF and FS-Focus are non-evolutionary filters; FS-Forward and FS-Backward algorithms are two classic, non-evolutionary wrapper greedy methods; FS-GGA generational genetic algorithm filter; FS-SSGA is a steady state genetic algorithm wrapper. Finally, the seventh dataset for every problem is composed of the original data. Table 1 shows the number of attributes present in every dataset.

In order to test the generalization capabilities of Symbiotic_RBF, a 10-crossfold validation method has been used, so that every dataset has been divided into 10 different sets of training-test patterns; for every training-set couple, the algorithm has been executed three times, and the results show the average of these 30 executions.

The performance of the method has been compared with two other evolutionary algorithms found in the literature: EvRBF [12, 14, 15], and CoCoRBFN [12]². Both of

¹ <http://www.ics.uci.edu/~mllearn/MLRepository.html>

² Results yielded by EvRBF and CoCoRBFN have been borrowed from [12].

them are designed to automatically build RBF neural networks. The first one by means of a Pittsburg genetic algorithm, and the second using a cooperative-competitive evolutionary algorithm, i.e., another kind of co-evolutionary approach.

The experiments carried out have been executed along 30 generations, with a population of 50 individuals; tournament selection used 5 individuals, and 60% of individuals were replaced in every generation.

Table 1. Number of attributes for every problem with respect to the different feature selection methods

Preprocessing	Ionosphere	Sonar	WDBC
None	34	60	30
FS-GGA	7	14	7
FS-LFV	9	17	8
FS-SSGA	9	17	8
FS-BackwardLVO	29	39	27
FS-Focus	3	6	10
FS-ForwardLVO	4	4	5

Tables 2 to 4 show the results obtained by Symbiotic_RBF, EvRBF and CoCoRBF in the above cited classification problems. Results allow to observe that the percentages of classification yielded by Symbiotic_RBF are similar or better to those obtained by EvRBF and CoCoRBF. Thus, the overfitting problem does not affect to this new algorithm, despite the fact that the training dataset is used more times in Symbiotic_RBF than in the rest of algorithms.

With respect to the number of neurons, as in EvRBF, but unlike CoCoRBF, Symbiotic_RBF does not fix neither imposes an upper limit to the size of the net. Although this could turn into over-specified models, Symbiotic_RBF builds nets whose size is similar to those designed with EvRBF. The number of neurons is then controlled by means of the genetic operators used by EvRBF, and the probabilities used by these operators are set by the symbiotic method.

Table 2. Experimental Results with Sonar database

Feature selection	Symbiotic_RBF		EvRBF		CoCoRBFN	
	RBF nodes	Test (%)	RBF nodes	Test (%)	RBF nodes	Test (%)
None	7	74,67	6	70,20	8	75,27
FS-GGA	12	84,61	8	80,49	8	70,47
FS-LFV	10	80,53	8	79,30	8	73,42
FS-SSGA	12	85,04	9	81,40	8	72,45
FS-BackwardLVO	15	85,41	10	81,34	8	73,68
FS-Focus	8	85,37	6	82,94	8	75,80
FS-ForwardLVO	6	82,61	4	80,90	8	74,51

Table 3. Experimental Results with Ionosphere database

Feature selection	Symbiotic_RBF		EvRBF		CoCoRBFN	
	RBF nodes	Test (%)	RBF nodes	Test (%)	RBF nodes	Test (%)
None	13	97,37	10	96,65	8	93,20
FS-GGA	16	95,51	12	93,55	8	83,39
FS-LFV	11	94,49	10	93,56	8	87,95
FS-SSGA	11	95,93	9	95,16	8	89,98
FS-BackwardLVO	15	96,90	14	96,34	8	91,61
FS-Focus	18	95,00	15	93,70	8	88,57
FS-ForwardLVO	11	95,06	9	95,05	8	92,49

The results show that Symbiotic_RBF can be effectively used to establish the parameter an evolutionary method needs, and, because of its nature of evolutionary algorithm, can be easily extended to any other algorithm. This property should be taken into account when statistical methods (like ANOVA) can not be considered because of the high number of parameters that must be evaluated. Actually, Symbiotic_RBF has been able to significantly reduce the number of parameters needed by EvRBF, so that only the parameters need to run the symbiotic method must be considered, and these ones can be set to standard values frequently used in literature.

On the other hand, the main disadvantage of the method is related to its computational cost. Taking into account that both EvRBF and Symbiotic_RBF are evolutionary algorithms, the time needed to find the final result seriously grows as the number of generations does. Nevertheless, the aim of the combined method is to find a net that performs a good classification and, due to the nature of these models, the exploitation phase is then performed very fast.

Table 4. Experimental Results with WDBC database

Feature selection	Symbiotic_RBF		EvRBF		CoCoRBFN	
	RBF nodes	Test (%)	RBF Nodes	Test (%)	RBF nodes	Test (%)
None	8	95,60	7	95,51	8	96,27
FS-GGA	12	96,55	12	96,42	8	95,85
FS-LFV	6	92,50	8	92,76	8	94,51
FS-SSGA	7	95,07	6	94,97	8	95,53
FS-BackwardLVO	6	95,14	8	95,45	8	96,37
FS-Focus	10	93,73	10	93,60	8	94,76
FS-ForwardLVO	7	95,20	9	95,40	8	96,48

5 Conclusions and Future Research

In this paper a system to automatically design models for classification using symbiosis is proposed. Thus, two populations coevolve in cooperative way contributing one to the evolution of the other and vice versa.

One of the populations is the method EvRBF, which provides the design of radial basis function neural nets by means of evolutionary algorithms. The second population evolves sets of parameters for the method EvRBF, being every individual of the above mentioned population a configuration of parameters for the method. The main goal of the proposed system, Symbiotic_RBF, is to find a suitable configuration of parameters necessary for the method EvRBF by means of coevolution.

The system Symbiotic_RBF has been evaluated across three data sets (Ionosphere, Sonar and Wdbc, from UCI repository), but every one has been filtered with 7 different features selection methods, resulting into 21 different datasets, and it has been compared with two other systems (EvRBF and CoCoRBFN) in order to evaluate the performance of the system.

The obtained results by Symbiotic_RBF, shown in tables 2, 3 and 4, are similar or even better than in the methods EvRBF and CoCoRBFN. In addition, Symbiotic_RBF obtains the best results of a total average for every method: Symbiotic_RBF, EvRBF and CoCoRBFN.

Future research will be driven into two main directions. Firstly, the number of the parameters for Symbiotic_RBF will be eliminated using CHC evolutive algorithm [5]. This way, we will be able to configure an evolutionary method (EvRBF) from scratch. Secondly, Symbiotic_RBF will be evaluated with functional approximation problems and extended in order to be used for time-series forecasting. In this case, Symbiotic_RBF will be able to both estimate the parameters for EvRBF and also select those intervals that needed to perform the optimal forecasting.

Acknowledgements

This work has been partially supported by the CICYT Spanish TIN2005-04386-C05-03.

References

1. Barbosa, H.J.C.: A coevolutionary genetic algorithm for a game approach to structural optimization. In: Proceedings of the Seventh International Conference on Genetic Algorithms, San Francisco, California, pp. 545–552 (1997)
2. Burdshall, B., Giraud-Carrier, C.: GA-RBF: A Self-Optimising RBF Network. In: ICANNGA 1997, pp. 348–351. Springer, Heidelberg (1997)
3. Castillo, P.A., et al.: G-Prop: Global optimization of multilayer perceptrons using GAs. *Neurocomputing* 35, 149–163 (2000)
4. Carse, B., Fogarty, T.C.: Fast evolutionary learning of minimal radial basis function neural networks using a GA. *Evolutionary Computing*, pp. 1–22. Springer, Heidelberg (1996)
5. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: First Workshop on Foundations of Genetic Algorithms, pp. 265–283. Morgan Kaufmann, San Francisco (1991)
6. Harpham, C., et al.: A review of genetic algorithms applied to training radial basis function networks. *Neural Computing & Applications* 13(3), 193–201 (2004)
7. Kriegel, H., Borgwardt, K., Kroger, P., Pryakhin, A., Schubert, M., Zimek, A.: Future trends in data mining. *Data Mining and Knowledge Discovery: An International Journal* 15(1), 87–97 (2007)

8. Mayer, H.A.: Symbiotic Coevolution of Artificial Neural Networks and Training Data Sets. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 511–520. Springer, Heidelberg (1998)
9. Merelo, J., Prieto, A.: G-LVQ, a combination of genetic algorithms and LVQ. In: Artificial Neural Nets and Genetic Algorithms, pp. 92–95. Springer, Heidelberg (1995)
10. Paredis, J.: Steps towards Co-evolutionary Classification Neural Networks. In: Brooks, R.I., Maes, P. (eds.) Proceedings Artificial Life, vol. IV, pp. 545–552. MIT Press / Bradford Books (1994)
11. Paredis, J.: Coevolutionary Computation. *Journal Artificial Life*, 355–375 (1995)
12. Perez-Godoy, M.D., Aguilera, J.J., Berlanga, F.J., Rivas, V.M., Rivera, A.J.: A preliminary study of the effect of feature selection in evolutionary RBFN design. In: Information Processing and Management of Uncertainty (IPMU 2008) (to appear, 2008)
13. Potter, M.A., De Jong, Kenneth, A.: Evolving Neural Networks with Collaborative Species. In: Proceedings of the 1995 Summer Computer Simulation Conference (1995)
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. Rivas, V.M., Garcia-Arenas, I., Merelo, J.J., Prieto, A.: EvRBF: Evolving RBF Neural Networks for Classification Problems. In: Proceedings of the International Conference on Applied Informatics and Communications (AIC 2007), pp. 100–106 (2007)
16. Rivera, A., Ortega, J., del Jesus, M., González, J.: Aproximación de funciones con evolución difusa mediante cooperación y competición de RBFs. In: AEB 2002, pp. 507–514 (2002)
17. Ros, F., Pintore, M., Deman, A., Chrétien, J.R.: Automatical initialization of RBF neural networks. *Chemometrics and intelligent laboratory systems* (2007)
18. Schwaiger, Roland, Mayer, H.A.: Genetic algorithms to create training data sets for artificial neural networks. In: Proceedings of the 3NWGA, Helsinki, Finland (1997)
19. Thompson, J.N.: *The Geographic Mosaic of Coevolution*. University of Chicago Press, Chicago (2005)
20. Whitehead, B.A., Choate, T.: Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction. *IEEE Trans. on Neural Networks* 7(4), 869–880 (1996)
21. Yao, X., Shi, Y.: A preliminary Study on Designing Artificial Neural Networks Using Co-Evolution. In: Proc. of the IEEE Singapore Intl. Conf. on Intelligent Control and Instrumentation, pp. 149–154. IEEE Singapore Section (1995)

Hybrid Multi-population Collaborative Asynchronous Search

Anca Gog, Camelia Chira, and D. Dumitrescu

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

Abstract. The paper explores connections between population topology and individual interactions inducing autonomy, communication and learning. A *Collaborative Asynchronous Multi-Population Evolutionary (CAME)* model is proposed. Each individual in the population acts as an autonomous agent with the goal of optimizing its fitness being able to communicate and select a mate for recombination. Different strategies for recombination correspond to different societies of agents (subpopulations). The asynchronous search process is facilitated by a gradual propagation of the fittest individuals' genetic material into the population. Furthermore, two heuristics are proposed for avoiding local optima and for maintaining population diversity. These are the dynamic dominance heuristic and the shaking mechanism, both being integrated in the CAME model. Numerical results indicate a good performance of the proposed evolutionary asynchronous search model. Particularly, proposed CAME technique obtains excellent results for difficult highly multimodal optimization problems indicating a huge potential for dynamic and multicriteria optimization.

Keywords: evolutionary algorithms, multi-agent systems, population topology, asynchronous search.

1 Introduction

Hybridization between multi-agent systems and evolutionary techniques has the potential to cope with difficult search/optimization problems. The paper explores connections between population geometry and agent interactions resulting in a *Collaborative Asynchronous Multi-Population Evolutionary (CAME)* model.

In the proposed *CAME* system each individual acts like an agent being able to communicate and select a mate for recombination. Individuals belong to one of the following three agent societies: *Local Interaction*, *Far Interaction* and *Global Interaction*. The society membership specifies the agent recombination strategy. The agent society of an offspring depends on the parents' agent society and a dominance relationship between societies.

The *CAME* population is endowed with a topological structure inducing a meaningful interaction between agent societies particularly for recombination purposes. Individuals are arranged according to their fitness using a predefined topology. A similar idea of placing population on a grid can be found in [1]. However, the latter

approach does not involve sorting of individuals according to the fitness or collaboration between individuals in different agent societies.

The introduced population model induces an asynchronous search scheme essentially based on the structure of the population (both geometrical and multiple-population). Furthermore, the induced asynchronous search process relies on the agent properties of autonomy and communication, which are vital for successful society interactions.

Numerical experiments prove the efficiency of the proposed technique by comparing it with the results obtained by recent evolutionary algorithms for several difficult unimodal and multimodal real-valued functions. The CAME model obtains excellent results for difficult highly multimodal optimization problems indicating a huge potential for dynamic and multicriteria optimization.

2 Population Hybrid Model

Exploring agent fundamental properties is a promising approach to designing evolutionary heuristics. This approach represents a shift in evolutionary paradigm design. The population evolution is not entirely controlled by some *a-priori* fixed operators. In the proposed agent-based approach, individuals can control to a certain extent their own evolution. This transfer of control towards individuals (that have agent capabilities) can trigger interesting behavioural types emerging from multi-individual interactions.

In order to facilitate a flexible search process in solving very difficult problems, population individuals can be considered members of a multi-agent system [2, 10, 12]. An agent can autonomously take decisions, acts on behalf of its creator, is situated in an environment and is able to perceive it, has a set of objectives and acts accordingly. The agent properties with a great potential for our approach refer to autonomy, communication and learning [4, 7, 8].

As an agent, each individual has the objective of optimizing its fitness. This objective is pursued by communicating with other individuals and selecting a mate for recombination based on individual strategies. The information exchanged by individuals via direct communication mainly refers to the fitness value. An individual invited to participate in a recombination process is able to make an autonomous decision about accepting or rejecting the request. The main factors of this decision include the strategy of the invited individual, the information received from the potential mate and a certain degree of randomness that induces autonomy.

For an efficient balance between search intensification and diversification specialized agent societies are considered. Agents from different societies are characterized by specialized behaviour in the recombination process.

The proposed CAME model implies three societies of individuals as follows:

1. *Local Interaction Agent (LIA) society*: LIA individuals select mates for recombination from their local neighbourhood (individuals geometrically situated on the previous layer).
2. *Far Interaction Agent (FIA) society*: FIA individuals select mates for recombination from more distanced layers on the population topology.
3. *Global Interaction Agent (GIA) society*: GIA individuals select mates for recombination on a global basis from the entire population.

LIA behavior emphasizes local search while FIA individuals are able to guide the search towards unexplored regions. The GIA society focuses on the global exploration of the search space realizing the connection between the LIA and FIA societies.

Similar to the autonomy in agent-based models, each individual invited to be a mate can accept or decline the proposal according to its own strategy. Individuals from LIA and FIA societies accept individuals from the same society or from the GIA society as mates. Individuals from GIA society accept any other individual as mate. Offspring are assigned to a certain society according to a dominance concept. The dominance in CAME refers to the LIA and FIA societies. Let us consider the case where LIA is the dominant agent society (LIA dominates FIA). In this situation, any combination of a GIA with a LIA individual results in an offspring belonging to LIA while a combination between a GIA and an FIA individual produces a GIA offspring. Furthermore, the information exchanged in the environment between individuals bias the decision about involvement in certain recombination processes.

The concepts of *Local* and *Far* Interaction societies receive a precise meaning if a certain topology of the entire population is specified. The CAME population is endowed with a topological structure that can be useful for selection and recombination purposes. The population topology can be associated to the asynchronous action of the search operators inducing a better balance between search intensification and diversification. An efficient exploitation of the useful genetic material already obtained in the search process is completed by maintaining population diversity.

Let $P(t)$ be the current population at iteration (epoch) t . The size of the population is fixed at n^2 , where n is an even number, in order to be compatible with the proposed topology. All individuals are sorted according to their fitness and are distributed over $n/2$ concentric layers starting with the fittest individuals on the most inner layer. The number of individuals placed on layer i , $i=0, \dots, n/2-2$, is $4(n-2i-1)$.

3 Proposed Hybrid Asynchronous Search Model

The proposed CAME model uses an asynchronous search scheme. Individuals from a layer are updated through proactive recombination and are involved in forthcoming recombination processes within the same epoch.

The individuals from the most inner layer (the fittest individuals in the population) are copied in the next population. Each individual from the population has the chance of being improved by getting involved in a recombination process. Population diversity is preserved as genetic material of both very fit and less fit individuals is considered in recombination. Decision about the second parent involved in each recombination process is based on agent society model, and this is the way the exploitation of the search space is pursued.

LIA individuals from layer c address mating invitations to individuals from layer $(c+1)$, where $c = 0, \dots, n/2-2$. FIA individuals from layer c address mating invitations to individuals from layer $(c+i)$, where $c = 0, \dots, n/2-3$ and $i \geq 2$ is randomly selected using an uniform distribution. FIA individuals from layer $(n/2-2)$ invite individuals from layer $(n/2-1)$. Individuals from the GIA society are more explorative. GIA individuals from a layer c may address mating invitations to individuals from any layer except layer c .

For each mating pair (x, y) the best offspring z obtained after recombination is mutated. The best between z and $mut(z)$ is compared to the first parent x and replaces x if it has a better quality. The elitist scheme that allows only better individuals to replace the first parents is mitigated by the fact that all individuals from the population are involved in recombination.

The useful genetic material collected from the entire population is propagated through the layers until it reaches the less fit individuals from the population. Furthermore, the co-existence of FIA and GIA individuals in the same population facilitates a more aggressive search space exploration.

The selection of a mate from a layer is performed using any selection scheme. Tournament selection scheme is considered for all the experiments reported in this paper.

4 Heuristics for Search Optimization in CAME

This section presents two strategies that adopted in CAME may improve the efficiency of the search process.

The first strategy refers to a *dynamic dominance* concept preventing the search to become trapped in local optimum. This strategy allows the change of the dominant agent society during the evolution of the population. If the quality of the solution detected does not improve after a specified number of epochs, CAME switches the dominance between LIA and FIA societies. The dynamic dominance strategy plays an important role in the way recombination results are valued and engaged during the search process.

The second strategy adopted in CAME engages a *shaking mechanism* to dynamically restructure the three CAME agent societies. If the quality of the detected solution does not improve after a certain number of epochs, the membership of individuals to one of the LIA, FIA and GIA agent societies is reassigned. Proposed shaking mechanism prevents the GIA society to take control over the entire population in a percolation-like behaviour. Moreover, this strategy emphasizes local optimization in the late stages of the search. This kind of local tuning can increase the accuracy of the detected solution and may be particularly useful in solving multimodal and dynamic optimization problems.

5 Numerical Experiments

The CAME model is engaged in set of numerical experiments and compared to several recently proposed evolutionary algorithms. A set of five unimodal and multimodal benchmark functions is considered [11]:

- f_1 - Shifted Sphere Function
- f_2 - Shifted Rotated Ackley's Function with Global Optimum on Bounds
- f_3 - Shifted Rastrigin's Function
- f_4 - Shifted Rotated Rastrigin's Function
- f_5 - Shifted Rotated Weierstrass Function

The proposed algorithm uses the following parameters: a population consists of 8×8 (64) individuals, the mutation probability is 0.2, the tournament size is $\frac{1}{2}$ of the considered group of individuals. Convex recombination and Gaussian mutation operators are used. The degree of randomness regarding the acceptance of a recombination request is 10%.

The CAME algorithm (CAMEA) has been compared to the following four evolutionary algorithms: Real-Coded Genetic Algorithm (RCGA) [5], Guided Differential Evolution Algorithm (GDEA) [3], Estimation of Distribution Algorithm (EDA) [13] and Self-adaptive Differential Evolution Algorithm (SaDEA) [9]. Tables 1 and 2 present the error values $err(x) = f(x) - f(x^*)$, where x^* is the real optimum and x is the detected solution. Each column corresponds to a method used for comparison. The

Table 1. Error values achieved in 25 runs for functions $f_1 - f_5$ with $D = 10$ after $1E+3$ FEs

		CAMEA	RCGA	GDEA	EDA	SaDEA
f_1	Best	4.6555E+02	1,5460E+04	7,8138E+02	2,8950E+03	8,1417E+02
	Mean	1.5679E+03	4,0250E+03	4,1422E+03	5,7163E+03	1,9758E+03
	StdAvg	7.6249E+02	1,2049E+03	1,5741E+03	1,6349E+03	6,5127E+02
f_2	Best	2.0441E+01	2,0286E+01	2,0480E+01	2,0437E+01	2,0385E+01
	Mean	2.0721E+01	2,0694E+01	2,0693E+01	2,0743E+01	2,0718E+01
	StdAvg	1.2488E-01	1,7609E-01	1,2400E-01	1,0947E-01	1,6960E-01
f_3	Best	1.3703E+01	5,4306E+01	5,1780E+01	5,7039E+01	3,6935E+01
	Mean	2.8350E+01	7,1658E+01	6,9319E+01	8,2039E+01	5,4397E+01
	StdAvg	8.5676E+00	9,3589E+00	7,6990E+00	9,9606E+00	7,5835E+00
f_4	Best	3.4433E+01	6,2407E+01	7,7355E+01	8,0269E+01	4,5212E+01
	Mean	5.9169E+01	8,9899E+01	9,4148E+01	1,0105E+02	7,5797E+01
	StdAvg	1.4648E+01	1,0229E+01	9,0170E+00	9,4880E+00	1,1696E+01
f_5	Best	5.5808E+00	9,9650E+00	9,7080E+00	9,2370E+00	8,9358E+00
	Mean	9.6909E+00	1,1539E+01	1,1407E+01	1,1885E+01	1,1408E+01
	StdAvg	1.7421E+00	6,4190E-01	8,2200E-01	8,1793E-01	9,5360E-01

Table 2. Error values achieved in 25 runs for functions $f_1 - f_5$ with $D = 10$ after $1E+4$ FEs

		CAMEA	RCGA	GDEA	EDA	SaDEA
f_1	Best	5.9605E-01	8,2384E-03	3,1630E+00	6,1428E-01	1,1915E-05
	Mean	7.7383E+00	4,0164E-02	1,0988E+01	2,6420E+00	3,8254E-05
	StdAvg	6.1572E+00	1,5428E-02	5,4600E+00	1,0717E+00	2,0194E-05
f_2	Best	2.0253E+01	2,0286E+01	2,0372E+01	2,0311E+01	2,0328E+01
	Mean	2.0483E+01	2,0504E+01	2,0510E+01	2,0530E+01	2,0506E+01
	StdAvg	1.4269E-01	1,1153E-01	9,1000E-02	9,7604E-02	9,5400E-02
f_3	Best	2.3687E+00	2,3915E+00	4,4930E+00	3,8268E+01	3,8698E+00
	Mean	4.5031E+00	1,1008E+01	1,5334E+01	4,9368E+01	6,6853E+00
	StdAvg	1.3205E+00	7,1721E+00	5,6630E+00	5,2638E+00	1,2652E+00
f_4	Best	1.2275E+01	1,6769E+01	1,5491E+01	2,6858E+01	2,4175E+01
	Mean	4.0541E+01	3,4098E+01	3,6792E+01	4,7850E+01	3,2230E+01
	StdAvg	1.5504E+01	5,7953E+00	1,0022E+01	8,4603E+00	5,4082E+00
f_5	Best	5.6636E+00	7,9186E+00	2,1150E+00	9,1362E+00	5,7757E+00
	Mean	8.5306E+00	9,9495E+00	4,7370E+00	1,0159E+01	8,0249E+00
	StdAvg	1.3069E+00	7,4983E-01	1,4020E+00	5,9719E-01	1,0255E+00

best and the average error values have been recorded after 1E+3 and 1E+4 function evaluations (FEs), after 25 runs of each algorithm for each function with dimension $D=10$. The obtained standard deviations are also listed in these tables.

Smaller error values are reported by CAME in approximately 68% of the considered cases. For 1000 FEs CAME performs even better. It is important to note that the best performance of the CAME technique is obtained for the most difficult multimodal functions considered. For these functions CAME clearly outperforms most rival methods considered after 10000 FEs as well. This can be considered as an indicator of the CAME potential for multimodal, dynamic and multicriteria optimization.

A statistical analysis is performed using the expected utility approach [6] to determine the most accurate algorithm. Let x be the percentage deviation of the solution supplied by a certain algorithm from the best known solution on a given function:

$$x = \left| \frac{sol - opt}{opt} \right| \times 100. \tag{3}$$

Let us define the following quantities:

$$\bar{b} = \frac{s^2}{x}, \quad \bar{c} = \frac{(\bar{x})^2}{s^2},$$

$$\bar{x} = \frac{1}{5} \sum_{j=1}^5 x_j, \quad s^2 = \frac{1}{5} \sum_{j=1}^5 (x_j - \bar{x})^2.$$

The expected utility function can be defined as [6]:

$$euf = \gamma - \beta(1 - \bar{b}t)^{-\bar{c}},$$

where a possible choice of parameters can be $\gamma = 500$, $\beta = 100$ and $t = 5.00E - 13$.

Table 3 presents the results of the statistical analysis test. It can be observed that CAME technique is ranked first for 1E+3 FEs.

Table 3. Statistical analysis for all considered algorithms on the average results obtained in 25 runs for the functions $f_1 - f_5$ with $D=10$, after 1E+3 FEs

	CAMEA	RCGA	GDEA	EDA	SaDEA
<i>euf</i>	398,24	395,32	395,17	393,08	397,75
<i>rank</i>	1	3	4	5	2

The statistical results obtained for 1E+4 FEs indicate a slightly weaker performance of CAME. One reason for this behavior is the fact that, even if the global behavior favors CAME algorithm, each other algorithm obtains better results for several functions. Another reason could be the fact that the recombination and mutation operators used for the current experiments (convex recombination and Gaussian mutation) are not among of the strongest operators that can be used for real codification. Better results could be obtained when using more fitted recombination schemes.

6 Conclusions and Future Work

A hybrid evolutionary model combining a topological individual arrangement with multi-population agent-based interactions is proposed. This technique called CAME relies on an asynchronous search scheme. Individuals are arranged according to their fitness using a predefined geometrical structure. Individuals from a layer are updated through proactive recombination and are involved in forthcoming genetic information exchange. Individuals' autonomy is encouraged by endowing them with agent-like features that allows them to choose and to accept mates based on their own strategy and information exchanged via direct communication. Different strategies correspond to different subpopulations that aim at preserving a good equilibrium between exploration and exploitation. A subpopulation dynamic dominance concept and a shaking mechanism are introduced as CAME integrated heuristics in order to further improve the efficiency of the search scheme. Experimental results indicate a good performance of the proposed search model and can be considered as an indicator of its potential for combinatorial optimization. Furthermore, numerical results emphasize that hybridization between multi-agent systems and evolutionary techniques is a promising approach to developing intelligent models able to cope with difficult search and optimization problems.

Ongoing work focuses on extending the proposed model to enable individuals with learning capabilities that allow them to make decisions about recombination strategy based on previous experience in its own evolution. Furthermore, dynamic optimization problems will be studied with the aim of adapting the CAME model for solving such problems. The agent property of reactivity can be emphasized in the CAME approach by enabling individuals to quickly respond to changes that occur in a dynamic environment. The potential of hybrid evolutionary agent-based models for dynamic optimization is huge as individuals can be endowed with the capability of not only reacting to changes but they can communicate making the latest information about the search space available in the system.

References

1. Alba, E., Giacobini, M., Tomassini, M., Romero, S.: Comparing Synchronous and Asynchronous Cellular Genetic Algorithms. 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. 601–610. Springer, Heidelberg (2002)
2. Bradshaw, J.M.: An Introduction to Software Agents. In: Bradshaw, J.M. (ed.) *Software Agents*. MIT Press, Cambridge (1997)
3. Bui, L.T., Shan, Y., Qi, F., Abbass, H.A.: Comparing Two Versions of Differential Evolution in Real Parameter Optimization, Technical Report, 4/2005, The Artificial Life and Adaptive Robotics Laboratory, University of New South Wales, TR-ALAR-200504009 (2005)
4. Chira, O., Chira, C., Tormey, D., Brennan, A., Roche, T.: An Agent-Based Approach to Knowledge Management in Distributed Design, Special issue on E-Manufacturing and web-based technology for intelligent manufacturing and networked enterprise interoperability. *Journal of Intelligent Manufacturing* 17(6), 737–750 (2006)

5. García-Martínez, C., Lozano, M.: Hybrid Real-Coded Genetic Algorithms with Female and Male Differentiation. In: Congress on Evolutionary Computation, pp. 896–903 (2005)
6. Golden, B.L., Assad, A.A.: A decision-theoretic framework for comparing heuristics. *European J. of Oper. Res.* 18, 167–171 (1984)
7. Jennings, N.R.: On Agent-Based Software Engineering. *Artificial Intelligence Journal* 117(2), 277–296 (2000)
8. Nwana, H., Lee, L., Jennings, N.: Coordination in Software Agent Systems. *BT Technology Journal* 14(4), 79–88 (1996)
9. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: Congress on Evolutionary Computation, pp. 1785–1791 (2005)
10. Russel, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 2nd edn. Prentice Hall, Englewood Cliffs (2002)
11. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore and KanGAL Report #2005005, IIT Kanpur, India (2005)
12. Wooldrige, M.: *An Introduction to Multiagent Systems*. Wiley & Sons, Chichester (2002)
13. Yuan, B., Gallagher, M.: Experimental results for the special session on real-parameter optimization at CEC 2005: a simple, continuous EDA. In: Congress on Evolutionary Computation, pp. 1792–1799 (2005)

An Evolutionary Approach for Tuning Artificial Neural Network Parameters

Leandro M. Almeida and Teresa B. Ludermir

Federal University of Pernambuco - Center of Informatics,
P.O. Box 7851, Cidade Universitária, Recife - PE, Brazil, 50732-970
{lma3, tbl}@cin.ufpe.br

Abstract. The widespread use of artificial neural networks and the difficult work regarding the correct specification (tuning) of parameters for a given problem are the main aspects that motivated the approach purposed in this paper. This approach employs an evolutionary search to perform the simultaneous tuning of initial weights, transfer functions, architectures and learning rules (learning algorithm parameters). Experiments were performed and the results demonstrate that the method is able to find efficient networks with satisfactory generalization in a shorter search time.

Keywords: Evolutionary algorithms, genetic algorithms, artificial neural network parameterization, initial weights, architectures, learning rules.

1 Introduction

There are a large number of articles in the literature devoted to the analysis and/or usage of Artificial Neural Networks (ANNs). Applications in commercial and industrial fields have contributed more strongly toward proving the importance, efficiency and efficacy of ANNs [9]. The success of ANN usage in fields such as pattern recognition, signal processing, etc. is incontestable [1], but a key problem concerning neural networks usage in practice remains as a challenge [2]. This problem is related to correctly building neural networks specially tailored to a specific problem, which is an extremely important task for attaining success in an application that uses ANNs [2], [3]. The search for ANN parameters is generally performed by a developer through a trial-and-error procedure. Thus, optimality or even near-optimality is not ensured, as the space explored is only a small portion of the entire search space and the type of search is rather random [6].

Manual tuning (trial-and-error search) of ANN parameters for a certain problem is considered a tedious, under-productive, error-prone task [1], [2], [3]. When the complexity of a problem domain increases and when near-optimal networks are desired, manual searching becomes more difficult and unmanageable [1]. An optimal neural network is an ANN tailored to a specific problem, thereby having a smaller architecture with faster convergence and a better generalization performance [1], [2], [3]. A near-optimal ANN is characterized by the choice of its specific, corrected parameters for a specific problem, thereby producing a satisfactory performance [2], [3]. The

construction of near-optimal ANN configurations involves difficulties such as the exponential number of parameters that need to be adjusted; the need for *a priori* knowledge of the problem domain and ANN functioning in order to define these parameters; and the presence of an expert when such knowledge is lacking [2], [3].

Considering the problems and difficulties related to the use of the manual method for tuning ANN parameters, the adoption of an automatic method for performing this tuning task emerges with the aim of avoiding such problems. A large number of papers are found in the literature devoted to constructing automatic methods for tuning ANN parameters. These can be categorized as evolutionary and non-evolutionary methods. One kind of evolutionary technique, the Genetic Algorithm (GA) [5], is often used to search near-optimal ANN models with topology optimization, as presented in [7], [8]. Others include transfer functions, initial weights and learning rules, as presented in [1], [6]. Automatic methods that use non-evolutionary techniques are focused mainly on the manipulation of ANN architectures and weights. Some of these non-evolutionary methods prune connections considered less significant [10], [11] or freeze weights when the same inputs are submitted to the network [10].

In this paper, we present a method denominated GANNTune (GA + ANN + Tune), which is an evolution of its antecessor NNGA-DCOD (aNN + GA - Direct enCODe), presented in [2]. The NNGA-DCOD method is mainly characterized by the adoption of Evolutionary ANNs (EANNs), a framework that makes possible the search for all ANN components needed for its functioning, as defined by Xin Yao [13]. EANNs include a sequential and nested layer search process, in which each layer has specific ANN information to be found by a specific GA. In NNGA-DCOD, the search for initial weights is performed in the lowest layer; the search for hidden layers, nodes per layer (architecture) and transfer functions occurs in the intermediate layer; and the search for learning algorithm parameters is performed in the highest layer. Thus, there are three GAs for searching these ANN parameters and, consequently, there is interdependence between these GAs [3]. While the NNGA-DCOD has three nested GAs, the GANNTune has only one GA. The aim of this approach is to reduce the time needed to perform the search for ANN parameters for a specific problem. This approach also seeks to preserve the NNGA-DCOD capability of finding solutions (ANN parameters) with similar or better quality. This paper is organized as follows: Section 2 presents the GANNTune method; Section 3 describes experimental results, including time consumption analysis; and Section 4 summarizes our conclusions and presents future work.

2 Evolutionary Approach for Tuning ANN Parameters

There are many different variants of Evolutionary Algorithms (EA) [5]. GA is a kind of EA that is widely employed to build automatic methods for tuning ANN parameters. The GANNTune is made up of a GA working directly with ANN parameters in their natural format. In other words, there is no encoding scheme. This approach was adopted with the intention of avoiding the frequent encoding and decoding tasks at each iteration of the search process, which occurs when the canonical GA is used [5]. Therefore, an individual is composed of all ANN parameters needed for its functioning: learning algorithm parameters, number of hidden layers and hidden nodes per layer, transfer functions and initial weights.

The individuals used by the GA for tuning ANN parameters are composed of real, integer and discrete information. Traditional genetic operators have been reformulated to deal with these kinds of values. These operators are the main contribution of this work, as there is an absence of such types of genetic operators in the literature.

The selection of N individuals is performed using the tournament strategy, with a random pressure rate of $P_p = 0.3$ and an elitism rate of $E = 0.1$. This information was found empirically. The tournament selection operator is used to select individuals to compose a new population (survivors) as well as individuals for recombination.

In recombination, given two selected individuals, a random variable is drawn from $[0,1)$ and compared to P_c . If the value is smaller than P_c , one offspring is created via the recombination of the two parents; otherwise, the offspring is created asexually by copying one of the parents [5]. The crossover rate used in this work was $P_c = 0.6$. After recombination, a new random variable is drawn from $[0,1)$ and compared to P_m . If the value is smaller than P_m , the mutation operator is applied to the individual; otherwise, nothing undergoes any changes. The mutation rate used is $P_m = 0.4$.

The execution of the genetic operator for recombination (and mutation) between individuals can be divided into three phases: First, the recombination of the learning algorithm information occurs; then the recombination of architecture information occurs; and, lastly, the recombination of initial weights occurs. The recombination of the learning algorithm information occurs with the generation of new values within a range based on the parent values. Based on probability, the mutation algorithm for the child performs an addition or subtraction operation in the child rates. Based on its current value, this operation causes a forty percent up or down (increasing or decreasing) change in the parameter. The evolutionary search for learning rules occurs with the search for the Back-Propagation (BP) algorithm parameters.

The recombination of architectures considers configurations with between one and three hidden layers and between one and 12 hidden nodes per layer. The selected individuals ("parents") are submitted to the crossover process. The process starts with the definition of the number of hidden layers that the child will have, which is obtained through the mean sum of hidden layers from the parents, rounded off based on probability problems. The dimensions of the hidden layers and the transfer functions for the child are then defined through random selection that considers all parent information.

The recombination operator for architectures produces individuals that are very similar to their parents. Thus, the mutation operator (algorithm in Figure 1) is applied after the crossover in order to maintain diversity in the population. Child selection for mutation is performed through a random process. The process starts with the number of children that will undergo mutation. A random number is then generated within the range $[0, 1)$. This value will be used to define whether an architecture will undergo an increase or decrease in the number of hidden layers or will undergo an increase ($P_{Inc} = 0.6$) or decrease ($P_{Dec} = 0.5$) in the number of hidden nodes per layer. If an architecture has only one hidden layer, the possibility of the number being three is great within all the possibilities. In the case of a selected random value failing to cause a significant change in the number of hidden layers, a set of values between $archval = [-2, 5)$ is generated and added to the current dimensions of the architecture. This range was defined empirically with the purpose of maintaining diversity among the individuals. If an architecture has two hidden layers, the possibility of the number being three is great, with the intention of reducing the number of hidden nodes per layer.

```

Data: childVet, n, m //children vector, number of children, mutation rate
Result: mutChild
Begin
  childNumber  $\leftarrow$  ((m/100) * n)
  while childNumber < 0 do
    probability  $\leftarrow$  rand(1)
    child  $\leftarrow$  raffle(childVet) //choose randomly one child
    if probability  $\leq$   $P_{inc}$  then
      if child.numHidLyrs = 1 or 2 then
        child  $\leftarrow$  addLayers(3)
      else
        child  $\leftarrow$  reduceLayersTo(2)
      else if probability  $\leq$   $P_{dec}$  then
        if child.numHidLyrs = 1 or 2 then
          child  $\leftarrow$  addLayers(1)
        else
          child  $\leftarrow$  reduceLayersTo(2)
        else
          child  $\leftarrow$  child + rand(archval)
        childVet  $\leftarrow$  childVet - child
        mutChild  $\leftarrow$  child
        childNumber  $\leftarrow$  childNumber - 1
      mutChild  $\leftarrow$  mutChild + childVet
  end

```

Fig. 1. Algorithm used for architectures mutation

In the recombination of initial weights, the parents can have different information regarding architecture; they have matrices of weights with different dimensions as well. For this reason, the recombination starts with generation of the initial weights randomly based on the architecture description *arch*. After random generation of the initial weights, an insertion of genetic material from the parents occurs, as the previous conditions have been satisfied. One of these conditions refers to the number of weight matrices (*nm*). The transfer of genetic material occurs up to the same number of matrices in the parents and child, and up to the minimal dimension of a matrix. The algorithm in Figure 2 describes the recombination of initial weights in more detail. The procedure `getDim` has the capability of identifying the dimensions of all matrices (number of lines and columns). The parameters $pflip = 0.5$ controls the source of genetic material to be transferred to child.

The selection of individuals for mutation is performed randomly, as individuals have no fitness yet. According to the mutation rate $P_m = 0.4$, forty percent of the children will undergo mutation. The mutation operator for initial weights consists of its adjustment (training) by 5 epochs with the BP algorithm using the data for training.

The method starts the search by randomly generating populations of individuals. Fitness is then calculated based on the ANN Mean Squared Error (MSE) achieved in the training set. The genetic operators maintain the diversity of individuals for the tournament search and a small range of random selection, where the new individuals generated for the next offspring must be distinct from individuals in the present offspring. The amount of individuals used in the GANNTune is small, but, as this method is iterative,

```

Data: prta, prtb, arch //parent A, parent B, architecture description
Result: child //new offspring
Begin
  weightsrange  $\leftarrow$  [-0.05, 0.05]
  weights  $\leftarrow$  getInitialWeights(arch, weightsrange)
  for id  $\leftarrow$  1:weights.nm do
    if prta.nm < index < prtb.nm then
      dim  $\leftarrow$  getDim(prta{id}, prtb{id}, weights{id})
      linmin  $\leftarrow$  min([dim.lina, dim.linb, dim.linc])
      colmin  $\leftarrow$  min([dim.cola, dim.colb, dim.colc])
      for e  $\leftarrow$  1:linmin do
        for g  $\leftarrow$  1:colmin do
          if rand(1) < pflip then
            weights.matrices{id}(e,g)  $\leftarrow$  prta.wgts{id}(e,g)
          else
            weights.matrices{id}(e,g)  $\leftarrow$  prtb.wgts{id}(e,g)
        end
      end
    end
  end

```

Fig. 2. Algorithm used for initial weights recombination

Table 1. Individual composition description

	Parameters for:	Values
Genetic Algorithm	Encoding	Direct
	Elitism	10%
	Recombination	80%
	Mutation	70%
	Pressure	30%
	Selection / Stopping criteria	Tournament / Max generations
	Population size	50
Generation	100	
ANNs	Type	MLP, Feed-forward
	Transfer functions	Pure-linear (P), Tang-sigmoid (T) and Log-Sigmoid (L)
	Hidden layers	Up to 3
	Hidden nodes per layers	Up to 16
	Training epochs	Up to 3
	Range of initial weights	[-0.05, 0.05]
	Output neuron	Linear
	Learning rate	[0.05 0.25]
Momentum rate	[0.05 0.25]	

many new individuals are created at every generation of each kind of search. Thus, the search space explored is large and satisfactory results are achieved.

3 Experimental Results

The experiments for evaluating the GANNTune were performed with seven well-known classification problems found in the UCI repository [12]. Cancer with 9 attributes (atb), 699 examples (exp) and 2 classes (cla); Heart-Cleveland with 35 atb, 303

exp and 2 cla; Pima-diabetes with 8 atb, 768 exp and 2 cla; Horse with 58 atb, 364 exp and 3 cla; Glass with 9 atb, 214 exp and 6 cla; Card with 51 atb, 690 exp and 2 cla; and Soybean with 82 atb, 683 exp and 19 cla. To perform the experiments, we used 30 two-fold iterations [4]. At each iteration, data were randomly divided into halves. One half was the input for the algorithm (70% for training and 30% for the validation set) and the other half was used to test the final solution (test set). The execution of one iteration corresponds to the creation of an initial population and execution of evolutionary search for 100 generations. After 30 iterations with different data division and initial populations, the best 30 ANN parameters are chosen based on training error, test error and architecture size. The results reported are the mean value from the 30 ANN parameters found for each classification problem.

The search for ANNs through trial-and-error was performed following the previously described methodology, using the same database split scheme and number of training epochs. We performed 30 runs in each fold for the network having between [2, 24] hidden neurons for one hidden layer with the (T) transfer function.

Table 2 displays the mean values from near-optimal ANNs found with GANNTune and trial-and-error methods. The mean of the architectures (Arch.) is based on the amount of hidden units/neurons. Regarding the number of hidden neurons, GANNTune found structurally worse networks (with many hidden neurons) than most of those found with the trial-and-error method. However, with application of the *t*-test, the results of GANNTune were statistically much better than the manual method for all problems considering the training and test errors.

Table 2. Description of near-optimal ANNs found with GANNTune and trial-and-error methods and time consumption of the developed method

Classification Problems	GANNTune				Trial-and-error		
	Time cons. in minutes	Arch.	Errors		Arch.	Errors	
			Training	Test		Training	Test
Cancer (ca)	14.36	25.4	3.03	3.63	16.8	7.16	8.14
Heart (he)	14.10	20.5	10.03	14.93	21.2	17.62	24.54
Pima (pi)	14.63	26.6	16.54	16.99	15.4	23.97	23.96
Horse (ho)	15.05	24.5	12.35	17.79	20	19.71	19.72
Glass (gl)	14.50	27.3	9.62	10.27	11.7	13.93	13.94
Card (cd)	14.65	18.6	8.55	12.35	12.3	13.03	16.07
Soybean (sy)	17.65	38.9	4.76	4.79	22.8	20.47	32.35

Considering the figures in Table 3, which are mean values, GANNTune is better than the other methods for searching near-optimal ANNs for the Pima-diabetes, Glass, Soybean and Horse problems regarding mean error (and similar to its antecessor NNGA-DCOD for the Horse problem), but the amount of nodes is high compared to the other methods. Comparing the GANNTune to NNGA-DCOD, the reduction in the mean performance for all problems is very expressive, but with GANNTune, the same does not occur with the number of nodes/connections.

None of the methods found in the literature on searching for near-optimal ANNs studied the time consumption of the search methodology. The second column in Table 2 displays information on the time GANNTune needs to search for near-optimal ANNs for each problem using all training algorithms. Time is related to the search

performed in one fold of the each problem in which the evolutionary search runs up to 100 generations. Therefore, the figures in second column of Table 2 are mean values from 30 iterations in minutes. The main drawback of methods that use evolutionary techniques (such as GAs) is the long search processing time. This is due to the fact that GAs considers a large search space (where near-optimal ANNs can be found) and, consequently, requires a large amount of time to explore this search space [6].

Table 3. Comparison between evolutionary and non-evolutionary (*) methods

Information	Problems	Methods						
		GANNTune	NNGA-DCOD [2]	GEPNET [8]	COVNET [8]	MOBNET [8]	COOPNN [7]	CNNDA* [10]
Test error	ca	3.63	6.22	-	-	-	1.38	1.15
	he	14.93	14.26	13.63	14.26	13.63	11.96	-
	pi	16.99	21.15	19.27	19.90	19.84	19.69	19.91
	ho	17.79	17.52	-	-	-	26.74	-
	gl	10.27	12.75	35.16	-	35.16	22.89	-
	cd	12.35	-	-	-	-	12.17	-
	sy	4.79	-	-	-	-	7.61	-
Number of hidden nodes	ca	25.4	12.8	-	-	-	5.89	3.5
	he	20.5	12.1	6.37	4.77	11.4	7.28	-
	pi	26.6	6.1	4.57	6.17	7.9	7.9	4.3
	ho	24.5	7.2	-	-	-	20.3	-
	gl	27.3	5.6	6.33	-	14.87	6.73	-
	cd	18.6	-	-	-	-	6.89	-
	sy	38.9	-	-	-	-	19.42	-

Compared to the time consumption of the NNGA-DCOD method, which is measured in hours of processing, the time consumption of GANNTune applied to the same classification problems is displayed in minutes of processing. This very significant reduction in time consumption was obtained with the adoption of only one GA to perform the search, whereas the NNGA-DCOD has three nested GAs to perform the same search.

4 Conclusions

The search for ANNs that are custom-tailored to a specific problem is considered a complex task and has mainly employed manual search methods. In this paper, we presented a new version of a hybrid method for automatically searching near-optimal ANNs. This method is composed of a combination of GAs and ANNs that search different ANN information. Experiments were performed and the results demonstrate that this method is able to achieve neural networks with satisfactory performances when compared to a simulation of the manual search method and other recent methods described in the literature. Moreover, the overall method requires just 15 minutes

to perform the search for ANN parameters (compared to 33.63 hours for its antecessor). In other words, the reduction in time consumption is clear. Future work will concentrate on refining the genetic operators and addressing the issues of pruning connections.

Acknowledgments

The authors would like to thank CNPq (Brazilian Research Council) for their financial support.

References

1. Abraham, A.: Meta learning evolutionary artificial neural networks. *Neurocomputing* 56, 1–38 (2004)
2. Almeida, L.M., Ludermir, T.B.: Automatically searching near-optimal artificial neural networks. In: 15th European Symposium on Artificial Neural Networks, pp. 549–554 (2007)
3. Almeida, L.M., Ludermir, T.B.: An improved method for automatically searching near-optimal artificial neural networks. In: International Joint Conference on Neural Networks (2008)
4. 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* 35(5), 915–927 (2005)
5. Eiben, A.E., Smith, J.E.: *Introduction to Evolutionary Computing*. Springer, Heidelberg (2003)
6. Ferentinos, K.P.: Biological engineering applications of feed-forward neural networks designed and parameterized by genetic algorithms. *Neural Networks* 18(7), 934–950 (2005)
7. García-Pedrajas, N., Hervás-Martínez, C., Ortiz-Boyer, D.: Cooperative co-evolution of artificial neural network ensembles for pattern classification. *IEEE Transaction on Evolutionary Computation* 9(3), 271–302 (2005)
8. García-Pedrajas, N., Ortiz-Boyer, D., Hervás-Martínez, C.: Cooperative co-evolution of generalized multilayer perceptrons. *Neurocomputing* 56, 257–283 (2004)
9. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, Englewood Cliffs (1999)
10. Islam, M.M., Murase, K.: A new algorithm to design compact two-hidden-layer artificial neural networks. *Neural Networks* 14(9), 1265–1278 (2001)
11. Ma, L., Khorasani, K.: New training strategies for constructive neural networks with application to regression problems. *Neural Networks* 17(4), 589–609 (2004)
12. Asuncion, A., Newman, D.: UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2007), <http://mllearn.ics.uci.edu/MLRepository.html>
13. Yao, X.: Evolving artificial neural networks. *Proceedings of the IEEE* 87(9), 1423–1447 (1999)

A Hybrid Evolutionary Multiobjective Approach for the Component Selection Problem

Andreea Vescan and Crina Grosan

Babes-Bolyai University, Computer Science Department, Kogalniceanu 1,
400084 Cluj-Napoca, Romania
{avescan, cgrosan}@cs.ubbcluj.ro

Abstract. Component selection is a crucial problem in Component Based Software Engineering (CBSE). CBSE is concerned with the assembly of pre-existing software components that leads to a software system that responds to client-specific requirements. We formulate the problem as multiobjective, involving 3 objectives. The approach used in this paper is an evolutionary computation technique combined with a greedy algorithm which is used to fine tune the solutions after the application of the genetic operators. The experiments and comparisons with Greedy technique show the effectiveness of the proposed approach.

Keywords: Component Selection, Hybrid evolutionary algorithms, Greedy.

1 Introduction

Component-Based Software Engineering (CBSE) is concerned with designing, selecting and composing components [1]. As the popularity of this approach and hence number of commercially available software components grows, selecting a set of components to satisfy a set of requirements while minimizing cost is becoming more difficult.

In this paper, we address the problem of automatic component selection. Informally, our problem is to select a set of components from the available component set which can satisfy a given set of requirements while minimizing the total cost of selected components. The dependencies between the components must be taken into account. To achieve this goal, we should assign each component a set of requirements it satisfies. Each component is assigned a cost which is the overall cost of acquisition and adaptation of that component. In general, there may be different alternative components that can be selected, each coming at their own set of offered requirements.

The paper is organized as follows: Section 2 starts with the problem formulation. Related work on Component Selection is discussed in Section 3. The proposed approach (that uses an evolutionary algorithm) is presented in Section 4. Greedy approach used for comparisons is described in Section 5. In Section 6 some experiments and comparisons are performed. We conclude our paper and discuss future work in Section 7.

2 Component Selection Problem: Formal Statement

A formal definition of the problem is as follows. Consider SR the set of final system requirements (target requirements) as $SR = \{r_1, r_2, \dots, r_n\}$ and SC the set of components available for selection as $SC = \{c_1, c_2, \dots, c_m\}$. Each component c_i can satisfy a subset of requirements $SR_{c_i} = \{r_{i_1}, r_{i_2}, \dots, r_{i_k}\}$.

In addition $Cost(c_i)$ is the cost of component c_i . The goal is to find a set of components Sol in such a way that every requirement r_j ($j = \overline{1, n}$) from the set SR can be assigned a component c_i from Sol where r_j is in SR_{c_i} , while minimizing $\sum_{c_i \in Sol} Cost(c_i)$ and having a minimum number of used components.

3 Related Work

Component selection methods are traditionally done in an architecture-centric manner. The authors of [2] present a method for simultaneously defining software architecture and selecting off-the-shelf components. They have identified three architectural decisions: object abstraction, object communication and presentation format.

Another type of component selection approaches is built around the relationship between requirements and components available for use. In [3] the authors have presented a framework for the construction of optimal component systems based on term rewriting strategies. Paper [4] proposes a comparison between a Greedy algorithm and a Genetic Algorithm. The selection function from the greedy approach take into consideration both number of provided (offered requirements) of the components and the cost of the component. In PORE [5] and CRE [6] the goal is to recognize the mutual influence between requirements and components in order to obtain a set of requirements that is consistent with what the market has to offer. The approach proposed in [7] considers selecting the component with the maximal number of provided operations. The algorithm in [8] considers all the components to be previous sorted according to their weight value. Then all components with the highest weight are included in the solution until the budget bound has been reached.

4 Proposed Hybrid Evolutionary Approach

The approach presented in this paper uses principles of evolutionary computation and multiobjective optimization [9] combined with a greedy approach used to fine tune the solutions obtained by applying genetic operators (mutation and crossover).

The following three objective functions are considered in this paper:

- the number of remain requirements to be satisfied, $fRemReq$;
- the total cost of the components used, $fCost$;
- the number of components used, $fNoComp$.

All the objectives are to be minimized.

There are several ways to deal with a multiobjective optimization problem. In this paper the Pareto dominance [10] principle is used.

Definition: Pareto dominance. Consider a maximization problem. Let x, y be two decision vectors (solutions) from the definition domain. Solution x *dominate* y (also written as $x \succ y$) if and only if the following conditions are fulfilled:

- (i) $f_i(x) \geq f_i(y); \forall i = 1, 2, \dots, n;$
- (ii) $\exists j \in \{1, 2, \dots, n\} : f_j(x) > f_j(y).$

That is, a feasible vector x is Pareto optimal if no feasible vector y can increase some criterion without causing a simultaneous decrease in at least one other criterion.

The approach used here is a steady state evolutionary algorithm. The main components of the hybrid evolutionary approach are presented in detail in the following subsections.

4.1 Solution Representation

A solution (chromosome) is represented as a string of size equal to the number of components from SC . The value of i -th gene represents the set of requirements the component satisfies.

An example of a chromosome for the experiment considered in Section 6 can have the following form: $[c_0[r_5], c_1[], c_2[], c_3[], c_4[r_1, r_3], c_5[], c_6[], c_7[r_4], c_8[], c_9[], c_{10}[], c_{11}[], c_{12}[], c_{13}[], c_{14}[], c_{15}[], c_{16}[], c_{17}[], c_{18}[], c_{19}[]]$, which means we are only considering the components c_0, c_4, c_7 . In the following we will only write the genes that are not empty. The values of the objective functions for this chromosome are: $fRemReq = 2, fCost = 42$ and $fNoComp = 3$.

4.2 Genetic Operators

The genetic operators used are crossover and mutation. Each of them is presented below.

4.2.1 Crossover Operator

We use a simple one point crossover scheme. A crossover point is randomly chosen. All data beyond that point in either parent string is swapped between the two parents.

For example, if the two parents are: $parent_1 = [c_0[r_5], c_6[r_2, r_4], c_{10}[r_0], c_{11}[r_3]]$ and $parent_2 = [c_0[r_5], c_{11}[r_4, r_2, r_3], c_{16}[r_0], c_{17}[r_1]]$ and the cutting point is 6, the two resulting offspring are: $offspring_1 = [c_0[r_5], c_6[r_2, r_4], c_{11}[r_4, r_2, r_3], c_{16}[r_0], c_{17}[r_1]]$ and $offspring_2 = [c_0[r_5], c_{10}[r_0], c_{11}[r_3]]$.

4.2.2 Mutation Operator

There are two types of mutation we propose for our problem and the application of the one or the other depends whether the chromosome has all the requirements satisfied or not.

First case: all the requirements are satisfied.

A gene whose corresponding component satisfies one requirement only (if any) is selected. This gene is replaced by another one (already used or not) which satisfies the same requirement. If there is no such gene, then we will replace this component (gene) with another one which is not yet used and which satisfies the same requirement. For example, having the chromosome: $[c_0[r_2, r_5], c_4[r_3], c_5[r_0, r_1], c_7[r_4]]$ with all

the requirements satisfied there exist two genes (components) having only one requirement offered. The requirement r_3 is satisfied by a different component: among the already used components there are no other ones that satisfy this requirement, and a different component is selected: c_{11} . The new obtained individual is $[c_0[r_2, r_5], c_5[r_0, r_1], c_7[r_4], c_{11}[r_3]]$.

Second case: *not all the requirements are satisfied.*

In this case, a requirement is randomly chosen from the set of requirements which are not currently satisfied. The same process is used when selecting the gene (component) to satisfy the chosen requirement. For example, having the chromosome: $[c_6[r_2], c_{10}[r_0], c_{12}[r_5]]$, not all the requirements are satisfied. The set of new possible added requirements is $\{r_1, r_3, r_4\}$. If the requirement that will be next added is r_3 , then there exists the component c_6 that satisfies this requirement. The new individual is: $[c_6[r_2, r_3], c_{10}[r_0], c_{12}[r_5]]$.

4.3 Solution Fine Tuning

An extra operation is used to fine tune the individuals after crossover and mutation. The fine tuning process consists of eliminating the requirements (in fact, the corresponding components) which are satisfied multiple times. We use a greedy based heuristic to select the components that will have the requirements removed.

The heuristic is as follow: for each gene in the chromosome a ratio of the number of the satisfied requirements and the cost of the gene (component) is computed. The remaining gene is the one with the maximum value of the proportion. For example, for the chromosome $[c_4[r_0, r_3], c_5[r_1, r_0], c_6[r_3, r_5, r_4], c_{10}[r_0]]$ the first, second and the last gene offer the first requirement r_0 . The values of computed ratios are: $2/17$, $2/12$ and $1/15$. The second ratio has the highest value and the second gene is going to be kept (the requirement r_0). Then the new chromosome is $[c_4[r_3], c_5[r_1, r_0], c_6[r_3, r_5, r_4]]$.

5 Greedy Approach

Greedy techniques are used to find optimum components and use some heuristic to generate a sequence of sub-optimums that hopefully converge to the optimum value. Once a sub-optimum is picked, it is never changed nor is it re-examined. The selection function is usually based on the objective function. We consider the proportion of number of requirements satisfied to the cost of the component as a measure to maximize our heuristic decision:

$$\left| \frac{SR_{c_i} \cup RSR}{Cost(c_i)} \right| is \max. \tag{1}$$

6 Experiments and Comparisons

A short and representative example is presented in this section. Starting with a set of six requirements and having a set of twenty available components the goal is to find a subset of the given components such that all the requirements are satisfied.

The set of requirements $SR = \{r_0, r_1, r_2, r_3, r_4, r_5\}$ and the set of components $SC = \{c_0, c_1, c_2, \dots, c_{19}\}$ are given. Table 1 contains for each component the provided services (in terms of requirements of the final system) and the cost of each component.

Table 1. Requirements elements of the components in SC

Component	r ₀	r ₁	r ₂	r ₃	r ₄	r ₅	Cost
c ₀			√			√	25
c ₁					√		7
c ₂				√			3
c ₃				√			3
c ₄	√	√		√			17
c ₅	√	√					12
c ₆			√	√	√	√	37
c ₇					√		5
c ₈			√				27
c ₉				√	√		8
c ₁₀	√						15
c ₁₁			√	√	√	√	34
c ₁₂			√			√	25
c ₁₃	√						13
c ₁₄			√				26
c ₁₅					√		6
c ₁₆	√						13
c ₁₇	√	√					12
c ₁₈						√	28
c ₁₉					√		5

6.1 Results Obtained by the Greedy Algorithm

In the current subsection we discuss the application of the greedy algorithm presented in Section 5.

The first step of the algorithm is the computation of the ratio among the number of requirements satisfied and the cost of the component. The component with the maximum ratio is chosen to be a part of the solution. In the first iteration of the algorithm the components c_2 and c_3 are the only components having the same (maximum) value for the calculated ratio. Randomly one of the components is chosen. Consider the component c_2 is selected. The set of already satisfied requirements contains only requirement r_3 . Next, we have to choose among the components c_7 and c_{19} . The seventh component is chosen randomly. The set of remaining requirements to be satisfied consist of $\{r_0, r_1, r_2, r_5\}$. Considering the new ratios, only two components - c_5 and c_{17} - have the maximum ratio value. The chosen component is c_5 and the new set RSR is $\{r_2, r_5\}$. The component c_0 is next chosen and the only remain requirement to be next satisfied is r_5 . Only one component has the maximum ratio value, the component c_{14} . The solution consists of five components and has the total cost 71, with the representation: $[c_0[r_2], c_2[r_3], c_5[r_0, r_1], c_7[r_4], c_{14}[r_5]]$.

6.2 Results Obtained by the Evolutionary Algorithm

We have performed 3 different experiments considering different population sizes and different number of generations. For all experiments we used the same values for mutation and crossover probabilities which are: mutation probability: 0.7; crossover probability: 0.7 and number of different runs: 100.

Some of the nondominant solutions are listed below: one is dominant by the cost and the other by the number of used components:

- solution dominant by the cost: the solution $[c_5[r_0, r_1], c_9[r_3, r_4], c_{14}[r_2, r_5]]$ has the cost 46 but has 3 components and the solution $[c_6[r_2, r_3, r_4, r_5], c_{17}[r_0, r_1]]$ has the cost 49 and only 2 components.
- solution dominant by the used number of components: the solution $[c_4[r_0, r_1, r_3], c_6[r_2, r_4, r_5]]$ uses 2 components but the cost is 54 and the solution $[c_0[r_2, r_5], c_9[r_3, r_4], c_{14}[r_2, r_5]]$ uses 3 components and the cost is only 45.

All the above discussed nondominant solutions have the values of the cost and the number of used components much better than the values of the solution obtained by the Greedy approach: cost 71 and 5 used components

Experiment 1

For the first experiment we considered the following parameters: population size: 10; number of iterations: 10. In Figure 1 the number of nondominated solutions obtained at each run is depicted. We can observe that in some situation we are obtaining 10 nondominated solutions which indicate that the whole final population is nondominated. The nondominated solutions obtained at the end of each run, cumulated for all 100 runs, are presented in Figure 2.

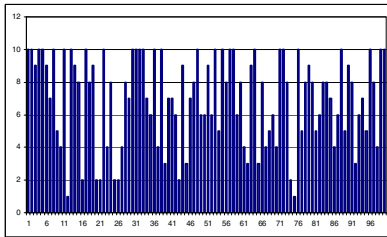


Fig. 1. Number of nondominated solutions obtained in 100 different runs considering 10 individuals and 10 iterations in each run

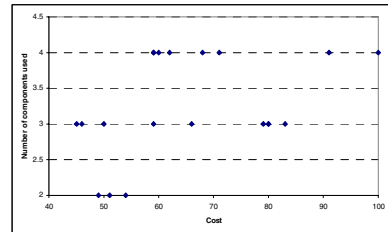


Fig. 2. The nondominated solutions obtained at the end of each run, considered over the 100 runs by using 10 individuals and 10 iteration

Experiment 2

Parameters used in this experiment are as follows: population size: 20; number of iterations: 20. The number of nondominated solutions obtained at each run is depicted in Figure 3. The nondominated solutions obtained at the end of each run, cumulated for all 100 runs, are presented in Figure 4.

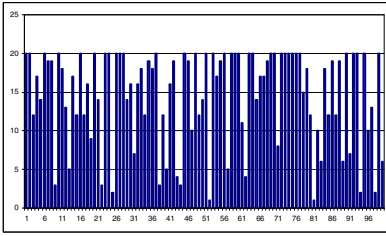


Fig. 3. Number of nondominated solutions obtained in 100 different runs considering 20 individuals and 20 iterations in each run

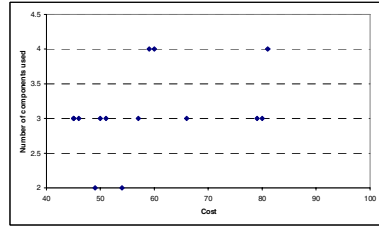


Fig. 4. The nondominated solutions obtained at the end of each run, considered over the 100 runs, by using 20 individuals and 20 iterations

Experiment 3

The third experiment performed considers the following parameters: population size: 50; number of iterations: 50. The number of nondominated solutions obtained in each of the 100 independent runs is depicted in Figure 5 and all the nondominated solutions obtained at the end of each run and cumulated over all 100 runs are depicted in Figure 6.

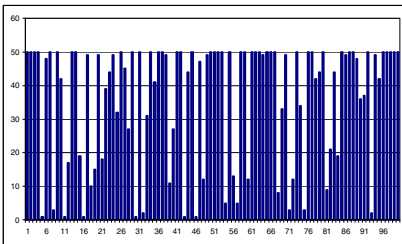


Fig. 5. Number of nondominated solutions obtained in 100 different runs considering 50 individuals and 50 iterations in each run

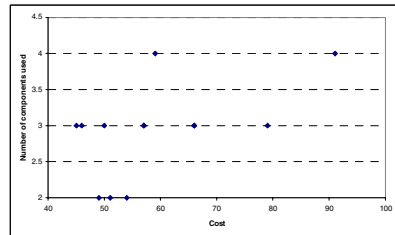


Fig. 6. The nondominated solutions obtained at the end of each run, considered over all 100 runs by using 50 individuals and 50 iterations

7 Conclusion and Future Work

The approach proposed in this paper combines the convergence efficiency of and evolutionary approach with a greedy approach whose role is to fine tune in an efficient way the solutions after the application of the genetic operators.

For the hybrid evolutionary approach, one can deduce that we are obtaining better results with a smaller population and a smaller number of individuals which shows that these parameters are not influencing that much the final result.

The proposed approach can be extended to a bigger set of components and requirements. We intend to extend our approach by specifying and proving the compatibility between two connected components. The protocol for each provided operations of a component have to be specified and included into the composition process. Another future extension is to use and define metrics for the computation of

the cost of a component, taking into consideration not only acquisition cost but also quality attributes (for non-functional requirements). A future work will discuss the current proposal using a real case study. New conditions probably will be imposed.

References

1. Crnkovic, I., Larsson, M.: Building Reliable Component-Based Software Systems. Artech House publisher (2002)
2. Mancebo, E., Andrews, A.: A strategy for selecting multiple components. In: ACM Symposium on Applied Computing, pp. 1505–1510. ACM Press, New York (2005)
3. Gesellensetter, L., Glesner, S.: Only the Best Can Make It: Optimal Component Selection. *Electron. Notes Theor. Comput. Sci.* 176, 105–124 (2007)
4. Haghpanah, N., Moaven, S., Habibi, J., Kargar, M., Yeganeh, S.H.: Approximation Algorithms for Software Component Selection Problem. In: The 14th Asia-Pacific Software Engineering Conference, pp. 159–166. IEEE Press, Los Alamitos (2007)
5. Alves, C., Castro, J.: Pore: Procurement-oriented requirements engineering method for the component based systems engineering development paradigm. In: International Workshop on Component Based Software Engineering, pp. 7–18 (1999)
6. Alves, C., Castro, J.: CRE: A systematic method for cots component selection. In: Brazilian Symposium on Software Engineering, pp. 193–207 (2001)
7. Fox, M.R., Brogan, D.C., Reynolds, P.F.: Approximating component selection. In: The 36th conference on Winter simulation, pp. 429–434. ACM Press, New York (2004)
8. Baker, P., Harman, M., Steinhofel, K., Skaliotis, A.: Search Based Approaches to Component Selection and Prioritization for the Next Release Problem. In: The 22nd International Conference on Software Maintenance, pp. 176–185. IEEE Press, Los Alamitos (2006)
9. Grosan, C.: A comparison of several evolutionary models and representations for multiobjective optimization. In: ISE Book Series on Real Word Multi-Objective System Engineering, ch. 3. Nova Science (2005)
10. Abraham, A., Jain, L., Goldberg, R.: Evolutionary Multiobjective Optimization: Theoretical Advances and Applications. Springer, London (2005)

A New Quantum Evolutionary Local Search Algorithm for MAX 3-SAT Problem

Abdessellem Layeb and Djamel-Eddine Saidouni

Lire laboratory, infography group, University Mentouri of Constantine Algeria
Layeb.univ@gmail.com, saidounid@hotmail.com

Abstract. The Max Sat problem is very known problem in computer science. It aims to find the best assignment for a set of Boolean variables that gives the maximum of verified clauses in a Boolean formula. Unfortunately, this problem was showed NP-Hard if the number of variable per clause is higher than 3. In this article, we propose a new iterative stochastic approach called QSAT based on a hybrid algorithm of Quantum Evolutionary Algorithm QEA and Local Search Algorithm LSA. QSAT is based on a basic core defined by a suitable quantum representation and an adapted quantum evolutionary dynamic enhanced by Local Search procedure. The obtained results are encouraging and prove the feasibility and the effectiveness of our approach. QSAT is distinguished by a reduced population size and a reasonable number of iterations to find the best assignment, thanks to the principles of quantum computing.

1 Introduction

A SAT problem is an assessment problem, to determine the satisfiability of a given logical formula. The Maximum Satisfiability (MAX-SAT) problem is an optimization variant of SAT problem. The problem consists to find an assignment of truth values that maximizes the number of satisfied clauses of a Boolean formula in Conjunctive Normal Form (CNF). In 1971, Stephen Cook [1] had demonstrated that the MAX SAT problem is NP-complete. This complex problem has several applications in different area such model checking, graph coloring, task planning, and so on. One type of SAT problems is 3-SAT problem where all Boolean expressions are written in CNF with 3 variables per clause. The following Boolean formula is in 3-CNF expression form: $(x_1 \text{ OR } x_2 \text{ OR } x_3) \text{ AND } (\text{not}(x_1) \text{ OR } x_2 \text{ OR } x_3) \dots \text{etc.}$

Where each x is a variable or a negation of a variable, and each variable can appear several times in the logic expression. To solve this problem, many algorithms were proposed. In fact, there are two classes of algorithms for solving instances of SAT in practice: Complete and Incomplete methods. The complete algorithms are able to verify the satisfiability or the unsatisfiability of the SAT problem, although they have an exponential complexity [2]. On the other hand, the incomplete methods find an optimal solution in good time, but they don't guarantee to give the exact solution of the problem. This class of methods contains Evolutionary Algorithms (EA) [3], Stochastic Local Search (SLS) methods [4] and hybrid methods of EA and SLS [5].

Far from SAT problem, Quantum computing is a new research field that covers investigations on quantum mechanical computers and quantum algorithms [6]. QC

relies on the principles of quantum mechanics like qubit representation and superposition of states. QC is able of processing huge numbers of quantum states simultaneously in parallel. QC brings new philosophy to optimization due to its underlying concepts. Recently, a growing theoretical and practical interest is devoted to researches on merging evolutionary computation and quantum computing [7, 8]. The aim is to get benefit from quantum computing capabilities to enhance both efficiency and speed of classical evolutionary algorithms. This has led to the design of Quantum inspired Evolutionary Algorithms QEA that have been proven to be better than conventional GAs. Unlike pure quantum computing, QEA doesn't require the presence of a quantum machine to work.

In this background, we propose in this article, a new iterative stochastic approach named QSAT based on a hybrid algorithm of Quantum Evolutionary and Stochastic Local Search algorithms. For that, a problem formulation in terms of quantum representation and evolutionary dynamic borrowing quantum operators were defined. The quantum representation of the solutions allows the coding of all the potential solutions of MAX SAT with a certain probability. The optimization process consists in the application of a quantum dynamic constituted of a set of quantum operations such as interference, quantum mutation and measure improved by the use of local search method in order to well explore the research space. The experiments carried out on QSAT showed the feasibility and the effectiveness of our approach.

Consequently, the remainder of the paper is organized as follows: Section 2 presents the problem formulation. A brief introduction to quantum computing is presented in Section 3. The proposed approach is described in Section 4. Section 5 illustrates some experimental results. Then, we terminate by giving conclusion and some perspectives.

2 Problem Formulation

Given a Boolean formula F expressed in CNF (*Conjunctive Normal Form*), let have n Boolean variables x_1, x_2, \dots, x_n , and m clauses. The problem of MAX SAT can be formulated as follow:

- An assignment to those variables is a vector $v = (v_1, v_2, \dots, v_n) \in \{0,1\}^n$.
- A clause C_i of length k is a disjunction of k literals, $C_i = (x_1 \text{ OR } x_2 \text{ OR } \dots \text{ OR } x_k)$
- Each literal is a variable or a negation of a variable
- Each variable can appear multiple times in the expression.

For some constant k , the k -SAT problem requests if there is a satisfying assignment that makes a formula $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$ true. The problem of MAX SAT can be defined by specifying implicitly a pair (Ω, SC) where Ω is the set of all potentials solution $(\{0, 1\}^n)$ and SC is a mapping $\Omega \rightarrow N$ called score of the assignment that is the number of true clauses. Each solution is viewed as binary vector. Consequently, the problem consists to define the best binary vector that makes the Boolean formula true.

3 An Overview of Quantum Computing

Quantum computing is a new theory which has emerged as a result of merging computer science and quantum mechanics. Its main goal is to investigate all the possibilities a computer could have if it followed the laws of quantum mechanics. The origin of quantum computing goes back to the early 80 when Richard Feynman observed that some quantum mechanical effects cannot be efficiently simulated on a computer. During the last decade, quantum computing has attracted widespread interest and has induced intensive investigations and researches since it appears more powerful than its classical counterpart. Indeed, the parallelism that the quantum computing provides reduces obviously the algorithmic complexity. Such an ability of parallel processing can be used to solve combinatorial optimization problems which require the exploration of large solutions spaces. The basic definitions and laws of quantum information theory are beyond the scope of this paper. For in-depth theoretical insights, one can refer to [6].

The qubit is the smallest unit of information stored in a two-state quantum computer. Contrary to classical bit which has two possible values, either 0 or 1, a qubit will be in the superposition of those two values. The state of a qubit can be represented by using the bracket notation:

$$|\Psi\rangle = a|0\rangle + b|1\rangle \quad (1)$$

where $|\Psi\rangle$ denotes more than a vector $\vec{\Psi}$ in some vector space. $|0\rangle$ and $|1\rangle$ represent respectively the classical bit values 0 and 1. a and b are complex number that specify the probability amplitudes of the corresponding states. When we measure the qubit's state we may have '0' with a probability $|a|^2$ and we may have '1' with a probability $|b|^2$. A system of m -qubits can represent 2^m states at the same time. Quantum computers can perform computations on all these values at the same time. It is this exponential growth of the state space with the number of particles that suggests exponential speed-up of computation on quantum computers over classical computers. Each quantum operation will deal with all the states present within the superposition in parallel. When observing a quantum state, it collapses to a single state among those states.

Quantum Algorithms consist in applying successively a series of quantum operations on a quantum system. Quantum operations are performed using quantum gates and quantum circuits. It should be noted that designing quantum algorithms is not easy at all. Yet, there is not a powerful quantum machine able to execute the developed quantum algorithms. Therefore, some researchers have tried to adapt some properties of quantum computing in the classical algorithms. Since the late 1990s, merging quantum computation and evolutionary computation has been proven to be a productive issue when probing complex problems. The first related work goes back to [7]. Like any other evolutionary algorithm, a quantum inspired evolution algorithm relies on the representation of the individual, the evaluation function and the population dynamics. The particularity of quantum inspired evolutionary algorithms stems from the quantum representation they adopt which allows representing the superposition of all potential solutions for a given problem. It also stems from the quantum operators it uses to evolve the entire population through generations [8, 9].

4 The Proposed Approach

The development of the suggested approach called QSAT is based mainly on a quantum representation of the research space associated with the problem and a quantum dynamic enhanced by local search procedure used to explore this space by operating on the quantum representation by using quantum operations.

4.1 Quantum Representation of MAX 3-SAT

Throughout this paper, N will represent the number of Boolean variables and M will denote the number of clauses in formula F . In order to easily apply quantum principles on Max 3-SAT problem, a quantum representation of the solutions is defined. The Boolean assignment is represented as binary vector of size N . In terms of quantum computing, each solution is represented as a quantum register of size N as shown in figure 1. The register contains superposition of all possible solutions. Each column represents a single qubit and corresponds to the binary digit 1 or 0. The probability amplitudes a_i and b_i are real values satisfying $|a_i|^2 + |b_i|^2 = 1$. For each qubit, a binary value is computed according to its probabilities $|a_i|^2$ and $|b_i|^2$. $|a_i|^2$ and $|b_i|^2$ are interpreted as the probabilities to have respectively 0 or 1. Consequently, all feasible solution can be represented by a Quantum Vector QV (fig 1) that contains the superposition of all possible solutions. This quantum vector can be viewed as a probabilistic representation of all the MAX 3-SAT solutions. When embedded within an evolutionary framework, it plays the role of the chromosome. Only one chromosome is needed to represent the entire population.

$$\left(\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_m \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_m \end{array} \right)$$

Fig. 1. Quantum representation of the SAT solution

4.2 Quantum Operators

The quantum operations used in our approach are as follows:

4.2.1 The Quantum Interference

This operation amplifies the amplitude of the best solution and decreases the amplitudes of the bad ones. It primarily consists in moving the state of each qubit in the direction of the corresponding bit value in the best solution in progress. The operation of interference is useful to intensify research around the best solution. This operation can be accomplished by using a unit transformation which achieves a rotation whose angle is a function of the amplitudes a_i, b_i and of the value of the corresponding bit in the solution reference (fig 2). The values of the rotation angle $\delta\theta$ is chosen so that to avoid premature convergence. It is set experimentally and its direction is determined as a function of the values of a_i, b_i and the corresponding element's value in the binary vector (table1).

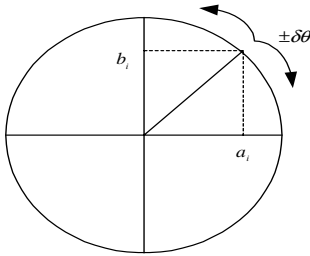


Fig. 2. Quantum interference

Table 1. Lookup table of the rotation angle

a	b	Reference bit value	Angle
> 0	> 0	1	$+\delta\theta$
> 0	> 0	0	$-\delta\theta$
> 0	< 0	1	$-\delta\theta$
> 0	< 0	0	$+\delta\theta$
< 0	> 0	1	$-\delta\theta$
< 0	> 0	0	$+\delta\theta$
< 0	< 0	1	$+\delta\theta$
< 0	< 0	0	$-\delta\theta$

4.2.2 Mutation Operator

This operator performs permutation between two qubits (fig 3). It allows moving from the current solution to one of its neighbors. It consists first in selecting randomly a register in the quantum matrix. Then, pairs of qubits are chosen randomly according to a defined probability. This operator allows exploring new solutions and thus enhances the diversification capabilities of the search process.

4.2.3 Measurement

This operation transforms by projection the vector matrix into a binary vector (fig.4). Therefore, there will be a solution among all the solutions present in the superposition. But contrary to the pure quantum theory, this measurement does not destroy the superposition. That has the advantage of preserving the superposition for the following iterations knowing that we operate on traditional machines. The binary values for a qubit are computed according to its probabilities $|a_i|^2$ and $|b_i|^2$

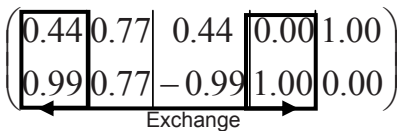


Fig. 3. Mutation operator

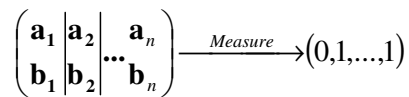


Fig. 4. Measure operation

4.3 Outline of the Proposed Framework

Now, we describe how the representation scheme including quantum representation and quantum operators has been embedded within an evolutionary algorithm and resulted in a hybrid stochastic algorithm performing variable truth assignment search.

Given a set S of SAT variables to be assigned, first, a quantum vector $QV(0)$ is constructed to represent all possible Boolean assignment. Starting from an initial binary vector $BV(0)$ extracted from the quantum matrix using the measurement operation, the algorithm progresses through a number of generations according to a quantum based dynamics. At each iteration, the following main tasks are performed: The assessment of the current assignment, the application of the interference operation, the application of the mutation operation, and the application of the measurement operation. The evaluation of the solutions is done by count of the number of satisfied clauses in the

Max 3-Sat problem. In order to enhance the capacity of space exploration, we have introduced the local search method in the quantum dynamics. QSAT uses a heuristic flip which consists in flipping the variables of the Boolean formula (fig. 5). The flip is accepted if there is improvement in the number of satisfied clauses. The process is repeated until there is no improvement. Our approach is flexible, so we can use other stochastic local search algorithms. Finally, the global best solution is then updated if a better one is found and the whole process is repeated until reaching a stopping criterion. In more details, the proposed QSAT can be described as in figure 6.

Input: a Truth assignment T

```

improvement=1;
repeat
    improvement=0;
    For i=1 to nVar
        flip the i-th variable of  $T$ ;
        compute the gain of flip;
        If (gain>=0)
            accept the flip;
            improvement= improvement +gain;
        End If
    End For
Until (improve==0)

```

Fig. 5. Flip heuristic procedure

Input: Boolean formula in CNF form A

- (1) Construct the initial Quantum vector QV
- (2) Generate the initial Binary vector BV'
- (3) Set $BV_{best} = BV'$ and $SC_{best} = SC (BV')$.

Repeat

- (4) Apply the interference operation on QV
- (5) Apply the mutation operation on QV
- (6) Apply the measurement operation on QV to derive a new binary vector BV_{new} .
- (7) Apply iteratively the local search procedure.
- (8) Evaluate the current assignment BV_{new} corresponding to QV .
- (9) Update BV_{best} .

Until a termination-criterion is reached.

Output: BV_{best} and $SC(BV_{best})$

Fig. 6. QSAT scheme

5 Implementation and Evaluation

QSAT is implemented in C++ and is tested on a microcomputer with a processor of 2 GHZ and 512 MB of memory. In order to assess the efficiency and accuracy of our approach, we perform experiments on a number of benchmark instances called AIM. The AIM instances are all generated with a particular Random-3-SAT instance generator [10]. In all experiments, the size of population is 1, the permutation probability is a tunable parameter which was set to 0.15, the interference angle is $\pi/20$, and the iteration number is 1000 iterations for quantum evolutionary algorithm and 1000 iterations for the local search procedure. The table 2 summarizes the obtained results. The results of the experiments are encouraging and prove the feasibility and the efficiency of our approach. Furthermore, we have compared our result with those of GSAT algorithm (table 2). Wilcoxon matched-pair signed-rank test were carried out to test the significance of the difference in the accuracy of our method and GSAT. At threshold $\alpha=0.05$, there is no significant difference between the QSAT and GSAT results. Finally, the performance of QEA without the local search algorithm is rather poor.

Table 2. Results for benchmark suite

Benchmark	Number of variables	Number of clauses	QSAT	GSAT
aim-50-1_6-no-1	50	80	79	79
aim-50-1_6-no-2	50	80	79	79
aim-50-1_6-no-3	50	80	79	79
aim-50-1_6-no-4	50	80	79	79
aim-200-2_0-no-1	200	400	399	399
aim-200-2_0-no-2	200	400	399	399
aim-200-2_0-no-3	200	400	398	399
aim-50-1_6-yes1-1	50	80	80	79
aim-50-1_6-yes1-2	50	80	80	79
aim-50-1_6-yes1-3	50	80	79	79
aim-100-2_0-no-1	100	200	199	199
aim-100-2_0-no-2	100	200	199	199
aim-100-2_0-no-3	100	200	199	199
aim-100-2_0-no-4	100	200	199	199
aim-100-3_4-yes1-1	100	340	335	335
aim-100-3_4-yes1-2	100	340	335	336
aim-100-3_4-yes1-3	100	340	335	335
aim-100-3_4-yes1-4	100	340	333	340

6 Conclusion

In our work, we described QSAT, a novel approach to solve the problem MAX 3-SAT. QSAT is based on a quantum evolutionary algorithm enhanced with local

search method. The quantum representation of the solutions allows the coding of all the potential variable truth assignment with a certain probability. The optimization process consists of the application of a quantum dynamics constituted of quantum operations such as the interference, the quantum mutation and measurement, enhanced by local search procedure. The choice of the local search procedure is crucial for the effectiveness of the resulting algorithm. The size of the population is considerably reduced thanks to the superposition principle. The experimental study proves the feasibility and the effectiveness of our approach. In addition, the proposed framework provides an extensible platform for evaluating different Local Search Algorithms.

References

- [1] Cook, S.A.: The Complexity of Theorem Proving Procedures. In: Proc. 3rd Ann. ACM Symp. On Theory of Computing, Association for Computing Machinery, pp. 151–158 (1971)
- [2] Alsinet, T., Manyà, F., Planes, J.: Improved Branch and Bound Algorithms for Max-2-SAT and Weighted Max-2-SAT. In: Proceedings of Sixth Catalan Conference on Artificial Intelligence (CCIA 2003) (October 2003)
- [3] Marchiori, E., Rossi, C.: A Flipping Genetic Algorithm for Hard 3-SAT Problems. In: Proc. of the Genetic and Evolutionary Computation Conference, vol. 1, pp. 393–400 (1999)
- [4] Marques-Silva, J.P., Sakallah, K.A.: GRASP: A Search Algorithm for Propositional Satisfiability. *IEEE Transactions on Computers* 48(5), 506–521 (1999)
- [5] Holger, H., Stützle, T.: Local search algorithms for SAT: An empirical evaluation. *Journal of Automated Reasoning* 24(4), 421–481 (2000)
- [6] Williams, C.P., Clearwater, S.H.: *Explorations in quantum computing*. Springer, Berlin (1998)
- [7] Han, K.H., Kim, J.H.: Quantum-inspired Evolutionary Algorithms with a New Termination Criterion, He Gate, and Two Phase Scheme. *IEEE Transactions on Evolutionary Computation* 8(2), 156–169 (2004)
- [8] Layeb, A., Saidouni, D.E.: Quantum Genetic Algorithm for Binary Decision Diagram Ordering Problem. The proceeding of *International Journal of Computer Science and Network Security* 7(9), 130–135 (2007)
- [9] Layeb, A., Meshoul, S., Batouhe, M.: Multiple Sequence Alignment by Quantum Genetic Algorithm. In: *The 7th International Workshop on Parallel and Distributed Scientific and Engineering Computing of the 20th International Parallel and Distributed Processing Symposium, Greece*, pp. 1–8 (April 2006)
- [10] Asahiro, Y., Iwama, K., Miyano, E.: Random Generation of Test Instances with Controlled Attributes. In: Johnson, D.S., Trick, M.A. (eds.) *Cliques, Coloring, and Satisfiability: The Second DIMACS Implementation Challenge*. DIMACS Series on Discr. Math. and Theor. Comp. Sci, vol. 26, pp. 377–394 (1996), <http://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/DIMACS/AIM/descr.html>

Neuro-evolutionary Decision Support System for Financial Time Series Analysis

Piotr Lipinski

Institute of Computer Science,
University of Wrocław, Wrocław, Poland
lipinski@ii.uni.wroc.pl

Abstract. This paper proposes a hybrid decision support system for financial time series analysis. It uses evolutionary algorithms to construct an artificial expert, consisting of some expert rules, which processes financial time series and provides an expertise. Such expert rules are either static coming from the domain knowledge or are in the form of multi-layer perceptrons representing expert rules extracted from data.

Results of a large number of experiments on real-life data from the Paris Stock Exchange confirm that the model of the hybrid decision support system is reasonable and it is able to generate reasonable solutions, not only over a training period, used in the training process, but also over a test period, unknown during constructing artificial experts.

1 Introduction

This paper proposes a hybrid decision support system for financial time series analysis. It combines evolutionary algorithms [3], applied to construct an artificial expert consisting of some expert rules, and neural networks [1], applied to extract such expert rules from data. It extends previous research on evolutionary discovering financial expertise [2], [6] by completing static expert rules, coming from the domain knowledge or human experts [9], with neural expert rules in the form of multi-layer perceptrons extracted from data [5]. In addition, our decision support system may contribute to more advanced frameworks, such as evolutionary decision making systems for stock trading [10] or evolutionary portfolio optimization systems [4], [7].

The paper is structured in following manner: Section 2 defines the expert rules and their extraction from data. Section 3 describes the artificial experts and their building from expert rules. Section 4 discusses results of experiments performed on real-life data from the Paris Stock Exchange. Finally, Section 5 concludes the paper and mentions some future extensions.

2 Expert Rules

Generally speaking, an expert rule is a function which evaluates a decision signal based on available information. This information is referred to as *a factual*

financial knowledge and denoted as \mathcal{K} . For instance, in the case of stock market trading, the factual financial knowledge \mathcal{K} consists of financial time series of historical stock prices. An expert rule is formalized in Definition 1.

Definition 1: An expert rule is a function

$$f : \mathcal{K} \mapsto y \in \mathbf{R} \quad (1)$$

which maps a factual financial knowledge \mathcal{K} to a real number y , which may be interpreted later as a decision signal. For instance, in the case of stock market trading, values lower than a certain threshold θ_1 correspond to a sell signal, values greater than a certain threshold θ_2 correspond to a buy signal, and remaining values correspond to no signal (in our experiments, $\theta_1 = -0.5$ and $\theta_2 = 0.5$). Let $D(y)$ denote the decision signal, i.e. a result y after such an interpretation.

2.1 Neural Expert Rules

In our approach, expert rules are either static coming from the domain knowledge or are in the form of multi-layer perceptrons representing expert rules extracted from data. Such a neural expert rule consists of a few multi-layer perceptrons, one perceptron for one possible decision signal. For instance, in the case of stock market trading, an expert rule consists of two multi-layer perceptrons, one for generating buy signals and one for generating sell signals.

In our experiments, each perceptron consists of an input layer of $M = 15$ units corresponding to the 15 preprocessing functions, described further, evaluated on the financial knowledge \mathcal{K} , one hidden layer of K neurons, where K varies in our experiments and is set separately for each of the two perceptrons, and an output layer of one single neuron corresponding to the response of the perceptron.

All the neurons in the hidden layer use the hyperbolic tangent activation function. The single neuron in the output layer usually uses the sigmoid activation function, but in some experiments also the softmax and linear activation functions were studied.

In the network, every two neurons in any two adjoining network layers are connected.

2.2 Extraction of Neural Expert Rules

Neural expert rules are extracted from data in a training process. Consider a specific time series and its observations recorded over a specific period composed of time instants t_1, t_2, \dots, t_T . Since \mathcal{K} varies through time, \mathcal{K}_{t_n} will denote the factual financial knowledge available at time t_n , i.e. the part of the time series recorded until the time t_n .

In neural expert rules, perceptrons are trained separately using different training data sets prepared from the time series considered recorded over the training period considered. Each data set consists of a set of input values and a set of corresponding target values.

Inputs of perceptrons are fed not with raw knowledge \mathcal{K}_{t_n} , but with preprocessed data $I_1(\mathcal{K}_{t_n}), I_2(\mathcal{K}_{t_n}), \dots, I_M(\mathcal{K}_{t_n})$, for $n = 1, 2, \dots, T$, where I_1, I_2, \dots, I_M denote some preprocessing functions, in order to fasten and improve training of perceptrons as well as to consider a more efficient form of expert rules (some preprocessing transformations cannot be incorporated in perceptrons themselves in their current form). In the case of stock market trading, data preprocessing focuses on the most popular 5 technical indicators [9], namely Moving Average (MA), Rate of Change (ROC), Stochastic Oscillator (SO), Relative Strength Index (RSI) and Easy of Movement (EMV). Each of them is applied with a horizon of 5, 10 or 15 time instants, which gives 15 preprocessing functions in total.

Target values for perceptrons represents perfect decision signals. In the case of stock market trading, target values for the perceptron generating buy signals consist of numbers $s_n \in \{0, 1\}$, where $s_n = 1$ when a local minimum of stock prices (a perfect buy signal) occurs at the time instant t_n , and $s_n = 0$ otherwise. Similarly, target values for the perceptron generating sell signals consist of numbers $s_n \in \{0, 1\}$, where $s_n = 1$ when a local maximum of stock prices (a perfect sell signal) occurs at the time instant t_n , and $s_n = 0$ otherwise. Both input data sets are standardized, so as to have zero mean and unit variance.

Training of each perceptron begins with randomly initializing weights of connections between neurons using the standard gaussian distribution. Afterwards, training input data vectors are fed one by one in a random order to the input layer of the perceptron and signals of neurons in consecutive layers are propagated up to the output layer. Next, the signal of the single neuron in the output layer is compared with the training target data corresponding to the chosen input vector. Weights of connections between neurons are optimized using Scaled Conjugate Gradient (SCG) algorithm [8]. Afterwards, new input vector is chosen and the entire process is repeated until the mean square error (MSE) is sufficiently low or a specific number of iteration is exceeded.

3 Artificial Experts

Expert rules produce advices for financial analysts, who may smoothly accept these advices when all the expert rules point in the same direction, but must wonder about a final decision when some expert rules are discordant. As, in practice, there are often opposing advices produced by expert rules, analysts must spend some effort to transform these advices into a final decision. One solution is to follow the majority of expert rules. An other solution is to define a set of favorite rules and consider only advices produced by this subset. A more complex solution is to use a weighted average of advices.

In a decision support system inspired from experience of financial analysts, the same problem appears. In this paper, in order to solve it, we try to select an efficient set of trading rules which defines the final decision by the advice proposed by the majority of expert rules in the chosen subset, ignoring the other expert rules. Such a set will be referred to as *an artificial expert*. It is formalized in Definition 2.

Definition 2: Let $\mathcal{R} = \{f_1, f_2, \dots, f_d\}$ be the entire set of available expert rules. An artificial expert is a subset of the entire set of expert rules

$$E \subset \mathcal{R}. \quad (2)$$

A result $E(\mathcal{K})$ of an artificial expert $E = \{f_{i_1}, f_{i_2}, \dots, f_{i_k}\}$, where $i_1 < i_2 < \dots < i_k$, for a given factual financial knowledge \mathcal{K} , is an arithmetic average of the results of the expert rules from the subset E

$$E(\mathcal{K}) = \frac{1}{k}(f_{i_1}(\mathcal{K}) + f_{i_2}(\mathcal{K}) + \dots + f_{i_k}(\mathcal{K})). \quad (3)$$

Obviously, in order to get a decision signal, the result of the artificial expert must be transformed using the auxiliary function D defined in the previous section.

3.1 Performance of Artificial Experts

Artificial experts are designed not to produce a single advice at a given date, but a sequence of advices at successive dates. In practice, a given advice may be accidental, depending on the context, while a sequence of such advices may suit accurately the financial market state.

Each artificial expert E proposes a sequence of decision signals at successive dates in the time period considered

$$D(E(\mathcal{K}_{t_0})), D(E(\mathcal{K}_{t_1})), \dots, D(E(\mathcal{K}_{t_{T-1}})). \quad (4)$$

Such a simulation over a specific time period characterizes better the behavior of the artificial expert than a single advice. Therefore, using such a sequence of advices, not a single advice, a performance of the artificial expert should be calculated.

In the case of stock market trading, for a sequence of decision signals to be useful, it is necessary to define the volume of stocks to be traded. In practice, this volume depends on individual trading abilities and preferences, but in decision support systems a common approach must be introduced.

In this paper, volumes of stock to buy or sell are obtained by simulating the behavior of an hypothetical investor. This investor is given an initial endowment (c_0, s_0) with c_0 the amount of cash and s_0 the initial quantity of stocks (in our experiments, $c_0 = 10000$ and $s_0 = 100$). Since trading generates transaction costs, they are assumed proportional with rate $\tau\%$ (in our experiments, $\tau = 0.2$).

At time t_0 , the investor takes a decision $D(E(\mathcal{K}_{t_0}))$. If the decision is to sell, i.e. $D(E(\mathcal{K}_{t_0})) = -1$, he sells $q\%$ of stocks. If the decision is to buy, i.e. $D(E(\mathcal{K}_{t_0})) = 1$, he invests $q\%$ of money in stocks. The parameter q in our experiments equals 50, which guarantees that the investor will not empty his account too fast. The transaction is executed at time t_1 and the investor's capital changes accordingly.

At time t_1 , the investor, makes a decision $D(E(\mathcal{K}_{t_1}))$, which is executed at time t_2 and the investor's capital again changes, and so on. Finally, c_0, c_1, \dots, c_T

denote the successive cash volumes and s_0, s_1, \dots, s_T the corresponding quantities of stocks over the time period considered. Sequences of cash volumes and quantities of stocks characterize performances of artificial experts.

In order to compare and valuate artificial experts, we use one of three different performance measures, introduced by financial analysts and market traders, which consider not only the future return rate, but also the risk related to achieving it [6]. The Sharpe ratio ϱ_{Sh} , the Sortino ratio ϱ_{So} and the Sterling ratio ϱ_{St} are defined by the following formulas:

$$\varrho_{Sh} = \frac{\mathbf{E}[R] - r_0}{\mathbf{Std}[R]}, \quad \varrho_{So} = \frac{\mathbf{E}[R] - T}{\mathbf{Std}[R|R < T]}, \quad \varrho_{St} = \frac{\mathbf{E}[R] - r_0}{\mathbf{MDD}[R]},$$

where R denotes the return rate, i.e. the ratio of the next-day capital to the previous-day capital, r_0 denotes the return rate of the risk-free asset (in our experiments, it equals 3% annually), T denotes a target return rate (in our experiments, it is the value of the stock market index) and \mathbf{MDD} denote the maximum drawn down of the return rate, i.e. the maximum loss during the time period considered.

3.2 Building Artificial Experts

For a given set \mathcal{R} of expert rules, a given performance measure ϱ , a given stock and a given training period, building artificial experts boil down to finding artificial experts with high values of the performance measure. In order to solve the optimization problem, our approach uses the Simple Genetic Algorithm (SGA) [3]. It evolves a population of N artificial experts E encoded in binary chromosomes $\mathbf{e} = (e_1, e_2, \dots, e_d) \in \{0, 1\}^d$, such that $e_i = 1$ if $f_i \in E$ and $e_i = 0$ otherwise. Each artificial expert is evaluated according to the objective function, which is the performance measure ϱ .

Figure 1 shows the framework of the SGA designed to optimize the objective function ϱ with a population \mathcal{P} composed of N individuals, where M denotes the number of children generated by the parent population \mathcal{P} , θ_C denotes the probability of crossover and θ_M denotes the probability of mutation (in our experiments, $\theta_C = 0.95$ and $\theta_M = 0.05$).

First, the algorithm creates an initial population \mathcal{P} at random (for each gene e_i of each chromosome \mathbf{e} , the probabilities of $e_i = 0$ and $e_i = 1$ both equal 0.5) and evaluates it. Afterwards, it evolves until a termination condition is satisfied: In *parent selection*, M parent individuals are randomly chosen from the population \mathcal{P} in such a way that the probability of choosing an individual is proportional to its value of the objective function. In *crossover*, each pair of parent individuals produces a pair of children either by one-point crossing over (with a large probability θ_M) or by simple copying (with a small probability $1 - \theta_M$). In *mutation*, each gene of each descendant is negated with a small probability θ_C . In *replacement*, best N children individuals replace the main population \mathcal{P} . Such an evolution terminates when all the individuals become identical or there is no increase in values of the objective function. More details on the SGA may be found in [3].

```

SIMPLE-GENETIC-ALGORITHM( $\varrho, N, M, \theta_C, \theta_M$ )
1   $\mathcal{P} \leftarrow \text{RANDOM-POPULATION}(N)$ ;
2  POPULATION-EVALUATION( $\mathcal{P}, \varrho$ );
3  while not TERMINATION-CONDITION( $\mathcal{P}$ )
4  do
5     $\mathcal{P}^{(P)} \leftarrow \text{PARENT-SELECTION}(\mathcal{P}, M)$ ;
6     $\mathcal{P}^{(C)} \leftarrow \text{CROSSOVER}(\mathcal{P}^{(P)}, \theta_C)$ ;
7    MUTATION( $\mathcal{P}^{(C)}, \theta_M$ );
8    REPLACEMENT( $\mathcal{P}, \mathcal{P}^{(C)}$ );
9    POPULATION-EVALUATION( $\mathcal{P}, \varrho$ );

```

Fig. 1. The Simple Genetic Algorithm designed to optimize the objective function ϱ with a population \mathcal{P} composed of N individuals, where M, θ_C, θ_M are algorithm parameters

4 Experiments

All the experiments concern real-life data from the Paris Stock Exchange, including financial time series of daily price quotations of about 40 stocks constituting the CAC40 index over a period starting on October 1, 2003 and lasting on December 31, 2006. Each experiment concerns a specific stock and a specific training and test period chosen randomly.

In order to illustrate the methodology, we present a few experiments concerning the stock Peugeot. In all these experiments, the training period for expert rule extraction starts on October 1, 2003 and lasts on November 25, 2005 (556 time instants). The test period starts on November 28, 2005 and lasts on December 31, 2006 (278 time instants). In order to assess and compare the effectiveness of the different perceptron structures, the example was studied several times with various numbers of hidden neurons.

First, 500 perceptrons generating buy signals were constructed and trained with different numbers of hidden neurons $K = 5, 10, 15, 30, 60$ (100 perceptrons for each K). Next, 500 perceptrons generating sell signals were constructed and trained with different numbers of hidden neurons $K = 5, 10, 15, 30, 60$ (100 perceptrons for each K). Afterwards, best 250 buy perceptrons and best 250 sell perceptrons were chosen and combined in pairs, which results in 250 expert rules. In addition to them, our system uses 250 static expert rules coming from the domain knowledge and human experts, formally defined by technical analysis indicators described in detail in [9]. Finally, the evolutionary algorithm was run to build an efficient artificial expert from these 500 expert rules.

Table 1 compares the results obtained with various algorithm configurations (appearing in the first column: the performance measure ϱ , the main population size N , the offspring population size M). Results concern the training period. In the second column, the performance $\varrho(e)$ is reported. Two following columns show return rates of the artificial expert and a benchmark strategy, so-called Buy-and-Hold (B&H), consisting in investing all the capital in stocks at the start of the period and keeping it until the end of the period under study. In

Table 1. Characteristics of solutions to the optimization problem of discovering an optimal investment strategy with different algorithm configurations for the stock Peugeot

Settings	Performance	Return	B&H	CAC40
Sharpe, $N = 100$, $M = 150$	0.0265	0.1104%	0.2979%	0.0167%
Sharpe, $N = 200$, $M = 300$	0.0291	0.1232%	0.2979%	0.0167%
Sharpe, $N = 500$, $M = 750$	0.0253	0.0927%	0.2979%	0.0167%
Sortino, $N = 100$, $M = 150$	0.0483	0.1282%	0.2979%	0.0167%
Sortino, $N = 200$, $M = 300$	0.0647	0.1336%	0.2979%	0.0167%
Sortino, $N = 500$, $M = 750$	0.0467	0.1183%	0.2979%	0.0167%
Sterling, $N = 100$, $M = 150$	0.0354	0.1487%	0.2979%	0.0167%
Sterling, $N = 200$, $M = 300$	0.0398	0.1633%	0.2979%	0.0167%
Sterling, $N = 500$, $M = 750$	0.0381	0.1311%	0.2979%	0.0167%
Single rule strategy (maximum)	0.0174	0.0845%	0.2979%	0.0167%

Table 2. Excess return rate of the investment strategy defined by the trading rule discovered over the return rate of the B&H strategy for 100 randomly prepared examples

Market Condition	Return over B&H	
	artificial expert	single rule strategy
extremely positive B&H, i.e. $0.05 \leq \text{B\&H}$	$0.03\% \pm 0.02$	$0.04\% \pm 0.02$
positive B&H, i.e. $0.00 \leq \text{B\&H} < 0.05$	$0.02\% \pm 0.01$	$0.01\% \pm 0.01$
negative B&H, i.e. $-0.05 \leq \text{B\&H} < 0.00$	$0.03\% \pm 0.02$	$0.02\% \pm 0.01$
extremely negative B&H, i.e. $\text{B\&H} < -0.05$	$0.04\% \pm 0.01$	$0.03\% \pm 0.02$

the last column, the return rate of the CAC40 index is given for comparison. Moreover, the last row shows results for an investment strategy based on one single expert rule (maximum of all the 500 expert rules).

In this example, the best results were obtained with the Sterling ratio, the population size $N = 200$ and the offspring population size $M = 300$. One may see that all the artificial experts significantly outperform single expert rules.

Experiments shows some dependencies between the two parameters N and M : too small population prevents the algorithm from an efficient optimization, as well as too large population slows down computations; the offspring population must be large enough to select promising individuals for the next iteration, usually $M = 1.5N$ leads to reasonable results.

For assessing the financial relevance of an artificial expert on the test period, its profitability is considered. It is defined by simulating the behavior of an hypothetical investor, as described in the previous sections. Finally, the profitability of the artificial expert is compared with the one generated by the Buy-and-Hold strategy.

Table 2 reports the excess return rate of the investment strategy defined by the artificial expert discovered over the return rate of the B&H strategy, separately for 4 types of stock market conditions defined by ranges of B&H values. Moreover, the last column shows results for an investment strategy based

on one single expert rule (maximum of all the 500 expert rules). One may see that the performance of trading rules depends on stock market conditions. However, in most cases, the approach proposed in this paper seems to overperform the B&H strategy and the single rule strategy.

5 Conclusions and Perspectives

This paper proposes a hybrid decision support system, which builds artificial experts for financial time series analysis based on neural expert rules extracted earlier from data. Experiments confirm that the artificial experts are able to generate reasonable decision signals, not only over a training period, used in the training process, but also over a test period.

Additional effort should be put on preprocessing functions, which may increase the efficiency of the model of expert rules, as well as on studying relations between performances of artificial experts over the training period and the test period. Moreover, such artificial expert might be applied in some complex systems, such as portfolio optimization systems [4].

References

1. Bishop, C.: *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford (1995)
2. Brabazon, A., O'Neill, M.: *Biologically Inspired Algorithms for Financial Modelling*. Springer, Heidelberg (2006)
3. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Reading (1989)
4. Korczak, J., Lipinski, P., Roger, P.: *Evolution Strategy in Portfolio Optimization*. In: Collet, P., Fonlupt, C., Hao, J.-K., Lutton, E., Schoenauer, M. (eds.) EA 2001. LNCS, vol. 2310, pp. 156–167. Springer, Heidelberg (2002)
5. Lipinski, P.: *Discovering Stock Market Trading Rules using Multi-Layer Perceptrons*. In: Sandoval, F., Gonzalez Prieto, A., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 1114–1121. Springer, Heidelberg (2007)
6. Lipinski, P., Korczak, J.: *Performance Measures in an Evolutionary Stock Trading Expert System*. In: Bubak, M., van Albada, G.D., Sloat, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3039, pp. 835–842. Springer, Heidelberg (2004)
7. Loraschi, A., Tettamanzi, A.: *An Evolutionary Algorithm for Portfolio Selection within a Downside Risk Framework*. In: Dunis, C.L. (ed.) *Forecasting Financial Markets*, pp. 275–286. Wiley, Chichester (1996)
8. Moller, M.: *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*. *Neural Networks* 6(4), 525–533 (1993)
9. Murphy, J.: *Technical Analysis of the Financial Markets*. NUIF (1998)
10. Tsang, E., Li, J., Markose, S., Er, H., Salhi, A., Iori, G.: *EDDIE In Financial Decision Making*. *Journal of Management and Economics* 4(4) (2000)

Optimization of Knowledge in Companies Simulating M6PROK[®]* Model Using as Hybrid Methodology a Neuronal Network and a Memetic Algorithm

Ana Maria Lara¹, Lourdes Sáiz¹, Joaquín Pacheco², and Rafael Brotóns³

¹ Department of Civil Engineering, University of Burgos, Spain
{amlara, lsaiz}@ubu.es

² Department of Applied Economy, University of Burgos, Spain
jpacheco@ubu.es

³ Grupo Antolín Deutschland
rabrocano@yahoo.es

Abstract. The pursuit of this paper is to give answers to the companies in order to know how profitable the knowledge they are acquiring, updating and transferring is. The scope of the application is carry out through a recent developed model, called *Model of the Six Profitable Stages* (M6PROK[®]) applied in twenty three companies of the service sector. Feasibility of the aforementioned model, results and conclusions are proved through the display of a Hybrid Architecture based in Neural Nets and Memetic Algorithms.

Keywords: Profitability Techniques, Knowledge Management Resolution, Neural Networks, Memetic Algorithms, Hybridation.

1 Introduction: General Theoretical Framework and Scope of Application

The creation and exchange of knowledge brings value when the three following minimum requirements are met [4]: A real personal commitment on the part of all staff in the firm to participate in the generation and the exchange of knowledge that is needed to carry out their tasks; the design and precise definition of a proper and adequate "space" both for the exchange of knowledge and for cooperation between participants and finally, the intensive use of information and communication technology (ICT) to collect and circulate knowledge. As pointed out by some authors, [1], Communities -not of interest nor of learning- but of Practice, are one of the most suitable approaches to ensure the integration of the proposed knowledge, because they are the analytical and participative units in which knowledge is created and where natural intrinsic acts of personal improvement and innovation take place. Thus, Knowledge Management grounded in the creation, maintenance and stimulation of Communities of Practice, might be the solution to challenges presented by knowledge-based firms. We can thus appreciate the mutations that companies have undergone over the last few hundred of

* Copyright number: 00/2006/3558. Author: Dr. Lara Palma.

years. It is therefore the precise and detailed review carried out in [2] what becomes the most accurate justification to the assertion: “Why is it that it has taken so long for knowledge to be considered as a key strategic factor!” The sector which is the object of this research, the “service sector” provides it with flavour. Here, knowledge management has no boundaries, as its use is not only limited to large firms where the volume of work, invoicing and its payroll are overwhelming, but also to small environments and on occasions, traditional family businesses. The real application to this 23 companies has been possible due to the recently developed model called M6PROK[®] and subsequently the used of a mathematical tool in the field of artificial neural architecture. Quantification of the intangible value of the service sector is the focus of this research.

2 Building Up a Picture of the Organization Knowledge, Using M6PROK[®] (Model of the Six Profitability Stages of Knowledge)

M6PROK[®] (showed in figure 1) is an own creation model and it is structured in a matrix setting. The architecture is based in a matrix setting using four intangibility measurement ratios that share two relevant characteristics; one is that their selection has a direct relationship with the cost factor, and, two, that they are caused by the lack of knowledge. These four ratios (cost of knowledge, cost of non-knowledge, the relevance of the thematic index and the state of knowledge) define the four axes on which the model is based on and on which results are determined. *Cost of knowledge* has been considered performance costs calculated on the basis of knowledge acquisition; *cost of non-knowledge* has been conceptualized as the cost deriving from insufficient awareness of concepts that pertain to the sector. *The relevance of the thematic index* is the relationship between the number of times that knowledge that belongs to a specific category is used and its economical repercussion on the company and, finally, *the state of knowledge* is necessary to quantify the level of knowledge that one worker has got in order to the required information needed to develop every activity of the sector; takes into account the situation that the sector is facing in relation to the learning capacity and the generation of new knowledge.

		KNOWLEDGE STATUS											
		NONE			LITTLE			ENOUGH			LOTS		
		LOW	MEDIUM	HIGH	LOW	MEDIUM	HIGH	LOW	MEDIUM	HIGH	LOW	MEDIUM	HIGH
RELEVANCE OF THE THEMATIC INDEX	MAXIMUM	COST OF KNOWLEDGE			COST OF KNOWLEDGE			COST OF KNOWLEDGE			COST OF KNOWLEDGE		
		LOW	MEDIUM	HIGH	LOW	MEDIUM	HIGH	LOW	MEDIUM	HIGH	LOW	MEDIUM	HIGH
		1	1	1	1	1	1	2	2	2	3	3	3
	HIGH	LOW	1	1	1	1	1	2	2	2	3	3	3
		MEDIUM	1	1	1	1	2	2	2	3	3	3	3
		LOW	1	1	2	2	2	2	3	3	3	3	3
	MEDIUM	LOW	1	1	1	1	1	2	2	2	3	3	3
		MEDIUM	1	1	2	2	2	2	3	3	4	4	4
		LOW	1	2	2	2	2	3	3	4	4	4	4
	LOW	LOW	1	1	2	2	2	3	3	3	4	4	4
		MEDIUM	1	2	2	2	3	3	4	4	4	5	5
		LOW	2	2	2	2	3	3	4	4	5	5	5

Fig. 1. M6PROK[®] Model

2.1 Handling and Displaying the Six Outputs of the M6PROK Model

The data sample comprises 1449 registry entries (individuals) that describe the quantity of knowledge used in such the different activities the workers can develop. Then, each cell contains one or more of these 1449 registry knowledge in order to the four parameters. The outputs of the M6PROK[®] are six profitability stages (answer considering periods of time); Profitability at very short-term (number one in the cells) leads into an immediate profit gain once the company acquires specific knowledge. Profitability at short-term (number two in the cells) -the acquisition of knowledge continues to be profitable, but that the return on the investment is not as immediate as in the previous case-. Profitability at medium-term (number three in the cells) - knowledge is being gained without urgency, allowing for it to be profitable in a relatively short period of time-. Profitability at long-term (number four in the cells) - the acquisition of knowledge continues to be a profitable exercise, although the results will only be seen in a longer period of time-. Strategic Profitability (number five in the cells) -the acquisition of knowledge is a strategy because there are just only estimated information about the period of time the investment will be done-. Non-Profitability (number six in the cells) -it does not exist a return on the investment-.

3 Display of the Hybrid Methodology Based in a Neural Network and Memetic Algorithm

In this research paper the richness is produced due to the neural algorithm has been applied to the four ratios (cost of knowledge, non knowledge, relevance of the thematic index and state of the knowledge). This is a value added regarding the methodology of [3] which work consisted on compiled only the three first. A model of multiple-layer perceptron is going to be considered for predicting the values of each ratio. A multiple-layer perceptron is a neural network model with several layers of nodes: an input layer, one or more hidden layer (in general not exceeds 2) and an output layer. The network reads an input values and the information propagates from one layer to another until the output is produced (“feed forward network”). In earlier applications of Neuronal Networks the classic backward Propagation is the training process. This method in general converges to local optimums and the Neuronal Network use a more global-optimum method based in Memetic Algorithms using some ideas from [18].

3.1 Neural Network Architecture

As the M6PROK[®] Model was developed together with the help of many experts working for several service companies; the purpose of this NN algorithm is to use this work to extrapolate the results to all different areas of the firm. Specifically a multiple-layer perceptron with a single hidden layer is going to be used. Let n denote the number of neurons in the input layer (which is logically the same as the number of input variables, so $n=4$), and m the number of neurons in the hidden layer (in this work $m=11$). For this work logically let us assume that the network has only one neuron in the output layer. Each neuron in the hidden layer, as well as in the output layer,

has a bias term which functions as a constant. The schematic representation is taken from [4]. Weights are sequentially numbered such that the weights that link the input layer with the first neuron of the hidden layer range from w_j to w_n where their bias term is w_{n+1} . The network reads an input vector $e=(e_j)$ and the information propagates from one layer to another until output s is produced. Thus, the input information of each neuron, $k=1, \dots, m$, of the hidden layer is given by z_k and the output neuron receives the following data. a_k are outputs from the hidden neurons. In our example a sigmoid function as a transfer function is used for the hidden layer and an identity function is used for the output neuron.

$$z_k = w_{k(n+1)} + \sum_{j=1}^n w_{(k-1)(n+1)+j} e_j \quad (1)$$

$$w_{m(n+2)+1} + \sum_{k=1}^m w_{m(n+1)+k} a_k \quad (2)$$

$$a_k = \frac{1}{1 + e^{-z_k}} \quad (3)$$

$$s = w_{m(n+2)+1} + \sum_{k=1}^m w_{m(n+1)+k} a_k \quad (4)$$

3.2 The Training Problem: The Memetic Algorithm

Training consists in adjusting the weight vector $w=(w_p)_{p=1..m(n+2)+1}$, in the following way: a set of training vectors S is presented, each consisting of an input and output vector; $\forall (e, y) \in S$, e is the input vector and y the expected output. Training consists of determining the weights that minimize the difference between the expected output and those produced by the network o when given the input vectors e . Using the least squares criterion this can be expressed with formula (5); E is the Mean Squared Error.

$$\min_w E = \frac{1}{|S|} \sum_{(e, y) \in S} (o - y)^2 \quad (5)$$

The variables are taken positives values, but it is not relevant in order to perform the proposed normalization; the normalization is a linear transformation, and the maximum value of the original variable corresponds with +1 in normalized variables and minimum value of original variable corresponds with -1; the output values are also normalized between -0.8 and +0.8. This normalization is recommended in different works, [5] and [6]. The algorithms based on gradient descent, such as the *back-propagation method* (developed in [7] and later in [8]; [9] and [10]), basically are iterative process that terminate when there is no further improvement in E . In fact, one of the limitations of these methods is convergence to poor local optimal and their dependence on the initial solution. In addition, they are not applicable to network models that use transfer functions for which derivatives cannot be calculated. The most recent global optimization methods, especially metaheuristic strategies, have

generally obtained better results than previous methods based on gradient descent. Thus, traditionally, *Genetic Algorithms* have been highlighted in works such as [11], [12], [13] and [14]. *Memetic Algorithms* have also yielded interesting results, [15], [16] and [17]. Excellent results have also been obtained in [5] and [6] with *Genetic Algorithms*, *Taboo Search* and *Simulated Annealing*, and in [4], where *Scatter Search* was used. In this work we have used a Memetic Algorithm -evolutionary methods with a population of solutions- for the training of this neural network model. One aspect to take into account is that due to the type of transfer functions used, these methods only seek weight values in the hidden layer: those in the output layer are obtained by the least squares method, ensuring good results, (this was suggested in [5]). Every solution is represented as a vector of weights $w = (w_1, w_2, \dots, w_{m(n+1)})$, the very well-known one-point crossover operator is used, mutation is changing a component of a solution from current value to other randomly in $(-0.5, +0.5)$; finally the improve method is based in local improvement method for continuous problems. More details can be found in [18].

Next is given a brief description of our Memetic Algorithm Procedure: *Generate randomly an initial population of solutions, Improve them with a pre-established method, Select a pair number of "parents" from population with probability proportional to its quality, Crossover: Randomly pair the parents from the mating pool and apply the crossover operator producing 2 "offspring" solution from each pair, Mutation: Apply with low probability the mutation operator to every offspring, Improve every offspring with the pre-established method, Replace the worst solutions in the population with the new offspring until reaching some stopping criterion.*

3.3 Application

Previous Processing and Performance of the Input and Output Parameters

The values of the parameters have been coded in the following way: For the variable Cost of Knowledge and Cost of Non-Knowledge: Low with 1 point, Medium with 2 points and High with 3 points. For the variable Relevance of the Thematic Index: Low with 0 point, Medium with 1 point, High with 2 points and Maximum with 3 points. For the output variable Profitability: At very short term with 1 point, At short term with 2 points, At medium term with 3 points, At long term with 4 points, Strategy Situation with 5 points and No Profitability with 6 points. Knowledge Status is always fixed. The six Profitability stages of Knowledge are changing depending on if the workers know none, little, enough or lots. Because it refers to variables with a marked ordinal character, this quantification does not distort the real meaning of the values (since it maintains the relative situation among them) and it also allows a more comfortable treatment with the neuronal net. The outputs obtained by the net will be continuous real values; for this reason they are transformed to the next in the 6 possible integers. For example, if the output value is 2.6, it is assigned the value of 3, which means, *Profitability at Medium Term*. Initially this approach has been the one used to determine the percentage of successes and the quality of the obtained model. In this same example (real exit 2.6), even though in smaller measure, it could also be assigned to the value 2, *Profitability at a Short Term*. Therefore the previous approach to define "success" has been relaxed and it has been considered as such if the real value resulting is less than one unit of the prospective value. This way in the previous

case, the real exit 2.6 is a success if the prospective value is 2 or 3 (Profitability *at a Short Term or Medium Term*). This result has made added up to near 10% the percentage of successes that are reported in the following subsection.

Computer System Programming Results

In this research we have a total size of 1449 cases in original database. We use the “1x10 cross-validation method” to measure the accuracy of the solutions obtained. This method is used very often in literature for checking classifiers and estimators methods. In abstract the 1x10 cross-validation divide randomly the database in 10 parts of the same size. Every time one part is used as test set the others joined nine parts as training set. In this way we can use 10 training set and the corresponding 10 disjoint test sets. Specifically the size of every training set is about 1750 cases and the size for test set is 194. In the next table are summarized the results (success ratio=percentage of “good predictions”) obtained by the model in training and test sets. The success ratios give enough information to conclude that the manner we have divided the cells in order to the different six profitable stages and the four parameters is feasible. In order to confirm the results, we have considered including a comparison of the results obtained with the multi-layer perceptron with another classification algorithm; we have used another classic methodology, Linear Regression, for comparing the performance with our Hybrid methodology. Specifically we have used the program SPSS 15.0 in order to apply Multinomial Logistic Regression (table 1).

Table 1. Percentage of success in prediction

Percentage of success in prediction using Hybrid methodology		Success ratio	Percentage of success in prediction using Logistic regression		Success ratio
Training Set	<i>Minimum</i>	75,7	Training Set	<i>Minimum</i>	75
	<i>Mean</i>	78,2		<i>Mean</i>	77
	<i>Maximum</i>	79,9		<i>Maximum</i>	78,1
Test Sets	<i>Minimum</i>	72,4	Test Sets	<i>Minimum</i>	71,6
	<i>Mean</i>	75		<i>Mean</i>	74
	<i>Maximum</i>	76,8		<i>Maximum</i>	75,9

The results are slightly better with our Hybrid methodology; in other hand these classic linear methods (as logistic regression or discriminate analysis) lead a more easy interpretation. Finally, the recently developed Hybrid Architecture used for this research and the learning algorithm has been programmed in language PASCAL with the compiler Borland Delphi (5.0). The tests have been carried out in a computer PC with processor Pentium IV at 2.4. Ghz. The stop approach for the training has been 200000 iterations of the memetic algorithm.

4 Discussion

This paper started explaining how learning is configured as the root that feeds creative minds. Without learning, creativity does not exist; neither does critical formulations, deviations in the generation of explicit ideas or originality. Learning is, therefore, a measurement of the organizational value, although, considering that it is people who

develop it and make it productive. Nowadays, the service sector environment is prone to achieve good management of its intellectual assets. M6PROK[®] provides a response to this need. This management strategy tool places the service sector in an advantage position against non-structured companies, by allowing these to set in operation the model's own capabilities and the body of knowledge developed for all the structures that make it up, be it the clients, workers, technicians, the competition and the lack of quality. In turn, and as a consequence of the use of the model, six levels of profitability become apparent (very short-term profitability, short-term profitability, medium-term profitability, long-term profitability, strategic profitability and non-profitability) providing enough information for a company to take corrective measures depending on the results of the four variables that make up the model.

The results obtained allow us also to propose one approach to the matter. Therefore it can be diagnosed where the service sector companies should invest their training budgets, in order to get the most out from their money and despite all difficulties, the model breaks the barriers between the departments of the company and forces them to work together and set common action plans to identify, share and acquire everything that they need to improve their development, in benefit of the whole company and employees. It forces also the company to arrange training plans for the employees and acts as a firewall against the brainscape. Acquiring knowledge is not anymore in fashion [19]. Now, companies have a model to measure what and how to learn and the most important point, when are they going to make up the profits and benefits of a good used of knowledge and learning [20]. These results allow us to conclude that: When knowledge status is non-existent (none), profitability is in the very short term. When knowledge status is little then profitability is in a short term. When knowledge status is enough, profitability will be in a medium term; and sometimes strategy situation can arise. When knowledge status is a lot, and depending the relevance of the thematic index three situations can arise. First, the profitability of the acquired knowledge can be in a long term; second, can be a strategic situation and third, in a critical percentage some occasions acquiring knowledge is not profitable.

Acknowledgements. Authors are grateful for financial support from the Spanish Ministry of Education and Science and FEDER funds (National Plan of R&D - Projects SEJ2005-08923/ECON) and from Regional Government of "Castilla y León" ("Consejería de Educación" -Project BU008A06).

References

1. Arbonés, A.: *Conocimiento para Innovar*. Madrid, Díaz de Santos (2006)
2. Nadal, J., Benaül, J.M., Sudría, C.: *Atlas de la Industrialización de España, 1750-2000*. In: Crítica, S.L. (ed.) *Fundación BBVA*, Barcelona (2003)
3. Lara, A.M., Sáiz, L.C., Pacheco, J.: A closed Model for Measuring Intangible Assets. A New Dimension of Profitability Applying Neural Networks. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) *IDEAL 2006*. LNCS, vol. 4224, pp. 815–822. Springer, Heidelberg (2006)
4. Laguna, M., Martí, R.: Neural Network Prediction in a System for Optimizing Simulations. *IEE Transactions* 34, 273–282 (2002)

5. Sexton, R.S., Alidaee, B., Dorsey, R.E., Johnson, J.D.: Global Optimization for Artificial Neural Networks: A Tabu Search Application. *European Journal of Operational Research* 106, 570–584 (1998)
6. Sexton, R.S., Dorsey, R.E., Johnson, J.D.: Optimization of Neural Networks: A Comparative Analysis of the Genetic Algorithm and Simulated Annealing. *European Journal of Operational Research* 114, 589–601 (1999)
7. Werbos, P.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard, Cambridge (1974)
8. Parker, D.: Learning logic. Technical Report TR-87', Center for Computational Research in Economics and Management Science. MIT, Cambridge (1985)
9. Lecun, Y.: Learning Process in an Asymmetric Threshold Network'. *Disordered Systems and Biological Organization*, pp. 233–240. Springer, Heidelberg (1986)
10. Rumelhart, D., Hinton, G., Williams, R.: Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition 1* (1986)
11. Montana, D.J., Davis, L.: Training Feedforward Neural Networks Using Genetic Algorithms. In: *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 379–384 (1989)
12. Schaffer, J.D., Whitley, D., Eshelman, L.J.: Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art. *COGANN-92 Combinations of Genetic Algorithms and Neural Networks*, 1–37 (1992)
13. Schaffer, J.D.: Combinations of Genetic Algorithms with Neural Networks or Fuzzy Systems. *Computational Intelligence: Imitating Life*, 371–382 (1994)
14. Dorsey, R.E., Johnson, J.D., Mayer, W.J.: A Genetic Algorithm for the Training of Feedforward Neural Networks. *Advances in Artificial Intelligence in Economics, Finance and Management 1*, 93–111 (1994)
15. Topchy, A.P., Lebedko, O.A., Miagkikh, V.V.: Fast Learning in Multilayered Networks by means of Hybrid Evolutionary and Gradient Algorithms. In: *Proceedings of International Conference on Evolutionary Computation and its Applications*, pp. 390–398 (1996)
16. Yao, X.: Evolutionary Artificial Neural Networks. *Int. Journal of Neural Systems* 4(3), 203–222 (1993)
17. Ichimura, T., Kuriyama, Y.: Learning of Neural Networks with Parallel Hybrid GA using a Royal Road Function. In: *IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 1131–1136 (1998)
18. Krasnagor, N., Aragon, A., Pacheco, J.: Memetic Algorithms in Metaheuristic Procedures for Training Neural Networks, pp. 225–248. Springer, Heidelberg (2006)
19. Corchado, E.S., Corchado, J.M., Sáiz, L., Lara, A.M.: Constructing a Global and Integral Model of Business Management Using a CBR System. In: Luo, Y. (ed.) *CDVE 2004. LNCS*, vol. 3190, pp. 141–147. Springer, Heidelberg (2004)
20. Corchado, E.S., Fyfe, C., Sáiz, L., Lara, A.M.: Development of a Global and Integral Model of Business Management Using an Unsupervised Model. In: Yang, Z.R., Yin, H., Everson, R.M. (eds.) *IDEAL 2004. LNCS*, vol. 3177, pp. 499–504. Springer, Heidelberg (2004)

STARMIND: Automated Classification of Astronomical Data Based on an Hybrid Strategy

Alejandra Rodríguez¹, Iciar Carricajo², Minia Manteiga², Carlos Dafonte¹,
and Bernardino Arcay¹

¹ Information and Communications Technologies Department, Faculty of Computer Science, University of A Coruña, 15071, A Coruña, Spain
{arodriguezf, dafonte, cibarcay}@udc.es

² Department of Navigation and Earth Sciences, University of A Coruña, 15011, A Coruña, Spain
{iciar, manteiga}@udc.es

Abstract. This paper describes the formulation and development of STARMIND, a hybrid system devoted to the automated classification of stellar spectra in the MK system. The MK system is an astronomical classification system used to cluster stars in morphological types based on stellar temperatures and luminosities. Our hybrid system is composed by a knowledge-based system that performs the first taxonomy in stellar types. A second-level system is based on Artificial Neural Networks and performs a more refined classification in stellar subtypes. Artificial Neural Networks were defined by selecting the optimal algorithms for training and architecture for each of the stellar spectra subtypes.

Keywords: Neural Networks, Knowledge-based Systems, Fuzzy Logic, Hybrid Systems, Spectral Features, Classification of Stars.

1 Introduction

The use of AI techniques for the analysis of astronomical data has been relatively frequent since the early nineties [1-3]. Starting with the two main branches of Artificial Intelligence - the symbolic-deductive systems (knowledge-based systems, KBS) and the so-called inductive or sub-symbolic approaches (artificial neural networks or connectionist systems) - we find that the vast majority of authors who have applied AI techniques in the field of Computational Astrophysics have done so according to the second paradigm, i.e. artificial neural networks (ANNs), and that no significant work can be found in the literature related to the application of Expert Systems to astronomical data analysis.

Astrophysics is moving towards a more rational use of the costly observational material by means of the intelligent exploitation of large terrestrial and spatial astronomical databases. Nowadays, every single project on astronomic instrumentation includes the creation of data archives and their future exploitation with automatic or pseudo-automatic analysis tools.

In the particular case of the classification of stellar spectra in the MK system, where spectral classes are defined by direct comparison with selected template spectra and the experience and expertise of the astronomer in classification tasks is completely relevant, we find that the nature of the problem fits with the scope of a specific KBS. This initial approach functions efficiently for a basic level of classification: the level of stellar types.

The MK stellar spectra classification system [4-6] establishes spectral types of stars in a sequence named O-B-A-F-G-K-M, ranging from the hottest or earliest stars (type O) to the coolest or latest (type M) stars. In addition, a luminosity class is assigned to the star depending on the intrinsic stellar brightness. The original rules for the classification of spectra in the MK system relied on the comparison with standard or template spectra, and were based on the strength and occurrence of atomic and molecular bands in stellar spectra in the optical, visible light. This includes the H Balmer lines, HeI and HeII lines, CaII lines, and a series of metallic lines and molecular bands (CH, TiO, etc).

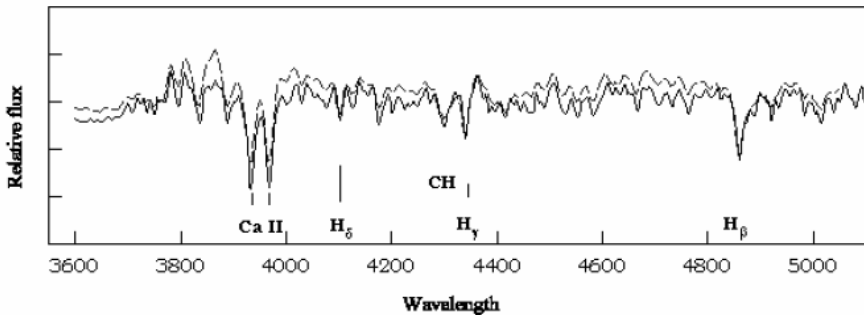


Fig. 1. Example of two stellar spectra showing some of the spectral features suited for stellar classification

If one tries to go beyond classifying types and assign stellar subtypes, i.e. a second level of spectral classification that ranges each of the stellar type (for instance B) in a numerical sequence starting from B0 until B9.5 in a decreasing order of the star's temperature, the borders between these sublevels are more difficult to concert among individual human classifiers. At this point, Artificial Neural Networks (ANNs) can be used to discern among the spectral subtypes.

We have carried out a sensibility analysis of the most relevant spectral features in order to define the different fuzzy sets, variables, and membership functions that determine the classification process. This analysis is described in section 2. Section 3 describes the details and performance of the Expert System, whereas section 4 is dedicated to the analysis and classification by means of specific ANNs. Finally, some classification results are presented and analysed in section 5. The system was tested against the NOAO Indo-US stellar spectral catalogue [7].

2 MK Astronomical Standard Spectra Database and Sensibility Indexes for Spectral Classification

Experts classify stellar spectra on the basis of a visual comparison with the templates that define the MK system. Experience has allowed us to conclude that in order to face the problem of MK automated spectral classification, it is very important to count with the reference of a spectral database that has the best possible resolution along all the subtypes. We have also noticed that it was imperative to use primary standard MK templates, the ones that originally define the system. So we prepared a spectra database composed by 340 items from both public “on-line” databases and “ad hoc” astronomical observation campaigns with the 2.5 m Nordic Optical Telescope at the *El Roque de los Muchachos* Astronomical Observatory (La Palma, Canary Islands).

In order to automatize the process of comparison with standard spectra, we must define a comprehensive set of morphological features that are to be measured in the spectra. A total of 65 spectral indexes were found to be well suited to characterize the spectral types. These features were measured in the spectra of the complete set of 340 MK templates. Figure 1 shows two of these spectral indexes as a function of MK types. Figure 2a illustrates how the measurement of the flux of a TiO band at 6208 angstroms gives increasing non zero values for types later than F9, whereas Figure 1b shows the behaviour of the H δ line as a function of the spectral type. Limiting numerical values were determined among wide groups of spectral classes (early, intermediate, late, OB, AF, GKM, etc). We then ranged their performance by determining, among all the spectra that populate a spectral type, the percentage that fulfils the expected interval value criteria for any individual index. As the next section will show, this has allowed us to assign an uncertainty factor to them.

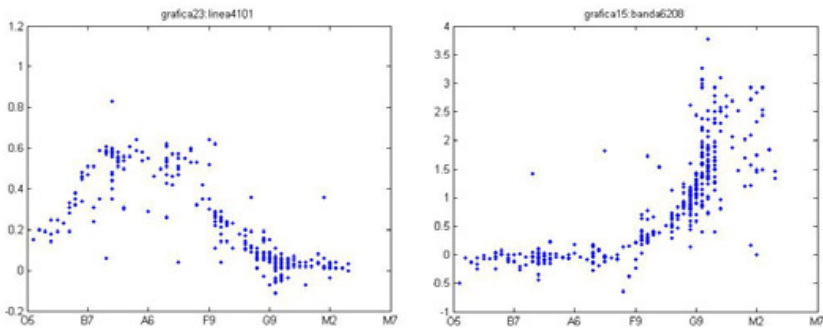


Fig. 2. Figure 2a (left) shows the performance of a spectral index based on a TiO band flux, Figure 2b (right) shows the behaviour of the H δ line flux as a function of spectral types

The developed system can be divided, from an operational point of view, into two main logical modules: the analysis module and the classification module. The analysis module is handled by the user and allows him to visualize, identify, and analyze the complete spectrum of each of the stars in a clear and understandable way. It carries out a morphological analysis of the stellar spectra, which includes the study and

measurement of all the spectral indexes needed for the classification. It can also extract and highlight diverse zones and even compare the problem-spectra to those of the reference catalogue by superposing them. The input of this module consists of the spectra that need classification; the output consists in a collection of values of the different spectral indexes, which will become the input data of the classification module.

The classification module is a KBS of which the user only gets to see the result, a levelled classification of the stars. The input of this module consists in the values of the spectral indexes that were obtained in the analysis module. The output is a classification of the input spectra that includes a credibility factor. This classification covers two levels: global classification (early, intermediate, and late), and a classification at the spectral type level. The results of the classification module are shown in the analysis module interface.

3 KBS Module Description

The selected parameters for spectral classification and the limiting values of each type and subtype were included in the expert system in the shape of fuzzy rules. The base of rules is the part of the system where the human classification criteria are reproduced. The STARMINd base of rules decision tree is shown in Figure 3, and some of the hierarchical decision rules and compliance percentages for spectral discrimination are presented in Table 1. We adopted production rules of the IF-THEN type to implement this module, because they easily reproduce the reasoning followed by the experts in the field. The conclusions allude to two levels of spectral classification (global and types).

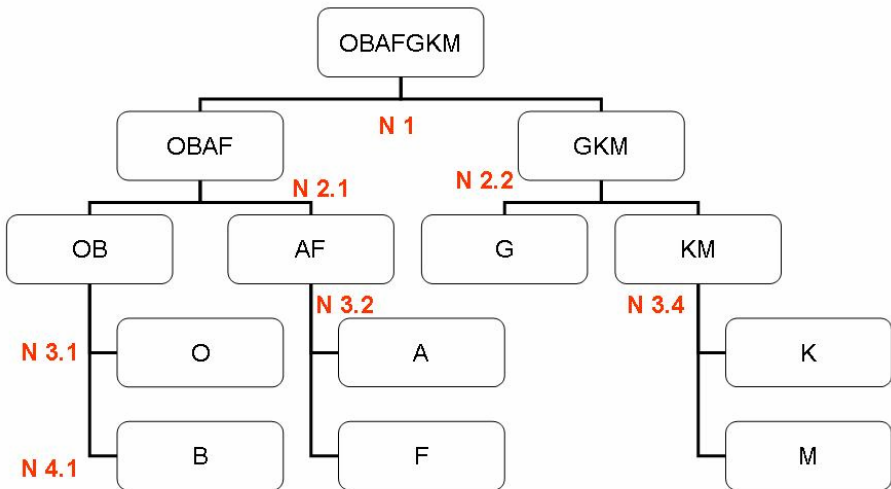


Fig. 3. STARMINd KBS base of rules decision tree

Table 1. Example of the rules at level N1 selected to discern among early-intermediate types (O-B-A-F) and intermediate-late (G-K-M) spectral types, showing the compliance percentages. The numbers refer to the wavelength of the features, and the chemical carrier is also indicated. Measurements of both spectral fluxes and EW (equivalent widths) of the lines were considered.

Rules N 1.1 ($\geq 95\%$)	Rules N 1.2 ($\geq 91\%$)	Rules N 1.3 ($\geq 90\%$)	Rules N 1.4 ($\geq 86\%$)
EW 4226	EW 5270	Band flux 6134	Band flux 6143
Ca II	FeI	TiO	TiO
Band flux 6208	Band flux 6154	EW 4455	Flux ratio 3933/4101
TiO	TiO	CaI	CaII, H δ
EW 4300	Band flux 4953	Band flux 4946	Flux ratio 4101/4144
Blend TiO-Fe	TiO	TiO	H δ /HeI
EW ratio 4226/4101	EW4077	Band flux 5886	Line flux 4101
CaII/ H δ	SrII	TiO	H δ
Band flux 5160	EW ratio 4300/4340	EW ratio 4226/4481	
TiO	Blend TiO-Fe/Hy	CaII/MgII	
	EW 4144		
	HeI		

The classification module actively communicates with the base of facts, where the descriptive information about the specific spectrum that is being analyzed at present is stored. We have used the Buchanan and Shortliffe methodology [8] to carry out an evolution that includes fuzzy sets and membership functions, contextualized for each spectral type and allowing superposition between them. In addition, we obtain the spectral classification of stars with a probability value that indicates the confidence grade. Sometimes this module can conclude an alternative classification of the spectra, e.g. in the case of obtaining a first classification with a significantly small truth value.

Since it could be interesting for the user to follow the reasoning process of the classification system, we have included an explanation module in the base of rules, so that the user can follow how the system has reached a final conclusion. This mechanism is based on transmitting the sequence of executed rules to the analysis module, which was shown to the user in a simple and accurate way. The output of the classification module is the two-level classification of the input spectra and the explanation of the system reasoning. These results are sent to the analysis module and shown to the users through the interface.

Table 2. STARMIND KBS for the derivation of spectral types. The system was tested over the NOAO-INDO-US spectral library.

Spectral Type	O	B	A	F	G	K	M
Number of stars	6	88	89	167	271	264	26
Success rate	100%	94.31%	91%	84.4%	87.08%	96.96%	100%

The results obtained by STARMINd KBS from a public spectral database, i.e. the NOAO-INDO-US library of stellar spectra [7], are shown in Table 2. This database is composed by 914 stellar individual spectra, which are in the same spectral region and spectral resolution as the MK standard set. The average success rate is of 91% (with 832 stars well classified in types).

4 ANN Module Description

When trying to refine the spectral classification beyond stellar types, by assigning MK subtypes to the spectra, the borders between these sublevels are more difficult to concert, even among individual human expert classifiers. This is when ANNs can be used to discern among those subclasses. Specific ANNs were defined by selecting the optimal algorithms for training and architecture, best suited for each of the stellar spectra subtypes. ANNs based on both supervised and non-supervised learning models were considered; mostly backpropagation, Kohonen, and radial basis functions (RBF) networks were implemented [9].

Finally, we opted for a feed-forward ANN architecture that uses the backpropagation training algorithm. The topology of the individual nets was adapted to each of the classification cases. The number of input nodes corresponds to the number of spectral indexes best suited to characterize each of the spectral types, with 6 to 29 nodes selected from the initial set of 65 indexes. These subsets were obtained after performing a sensibility analysis. The output node is the classification subtype value. We consider that the net has converged when the mean square error (MSE) is equal or lower than 0.05.

The training, validation, and testing patterns that were presented to the ANN were obtained automatically by adding the necessary functions to the spectral analyzer developed in the KBS system. Once the input values were obtained, they were normalized by means of a contextualized and specific sigmoidal function for each spectral subtype. The training set was composed by the MK standard stars belonging to each of the subtypes, and the evaluation of the system was performed on the NOAO-INDO-US spectral catalogue.

5 Results and Conclusions

ANN trainings and tests were performed considering the complete set of 340 MK standards (70% for training and 30% for tests), whereas the performance of the system was evaluated over a sample of 732 stars belonging to the already mentioned public library of stellar spectra NOAO-INDO-US. The results of the STARMINd hybrid system, combining both KBS and dedicated ANNs to classify stellar spectra at the level of spectral subtypes, are shown in Figure 4. The agreement between the classification listed in the library and that obtained by STARMINd is excellent for early and late stellar spectra types (O9 to F7, and G6 to M6). For intermediate types (F8, G0, G1, and G2), the success rate decreases considerably. There is an astrophysical explanation for this fact: in the case of the former, the chemical abundance of elements (in Astrophysics, the “stellar metallicity”) influences the appearance of some of

the most relevant spectral lines considered in the indexes. The MK system for stellar spectra classification was originally proposed for solar-type stars, with solar-like metallicity. This property is fulfilled in the case of the MK standard set of spectra, but not for the all the spectra in the NOAO-INDO-US. The results improve considerably (compliances $\geq 80\%$) when only solar-type metallicity spectra were taken into account for the evaluation of the system. This implies that our hybrid approach can be considered a very versatile and flexible automatic technique to classify the spectra of stars.

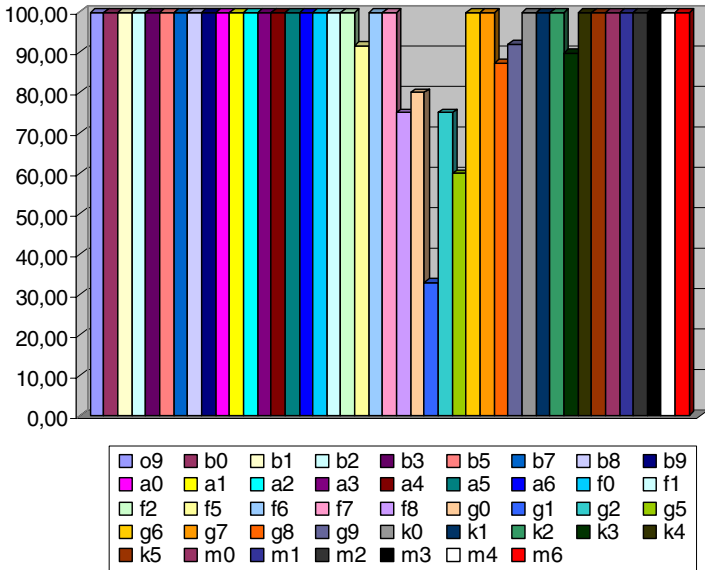


Fig. 4. STARMINd Hybrid System classification success rates for a sample of 732 spectra belonging to a public catalogue [7]

Acknowledgments. This research was funded by the Spanish Ministry of Education and Science under grant ESP2006-13855-C02-02. Allocation of observing time at the Nordic Optical Telescope at the *El Roque de los Muchachos Observatory* is also acknowledged.

References

1. Snider, S., Allende, C., Von Hippel, T., Beers, T., Sneden, C., Qu, Y., Rossi, S.: Three-dimensional spectral classification of low-metallicity stars using artificial neural networks. *The Astrophysical Journal* 562, 528 (2001)
2. MacDonald, D., Corchado, E., Fyfe, C., Merényi, E.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 649–654. Springer, Heidelberg (2002)

3. Weaver, W.: Spectral Classification of Unresolved Binary Stars with Artificial Neural Networks. *The Astrophysical Journal* 541, 298 (2000)
4. Morgan, W.W., Keenan, P.C., Kellman, E.: An atlas of stellar spectra, with an outline of spectral classification. University of Chicago Press (1943)
5. Morgan, W.W.: The MK Process and Stellar Classification. In: Garrison, R.F.(ed.), vol. 18. University of Toronto, David Dunlap Observatory, Toronto (1984)
6. Keenan, P.C., Yorks, S.B.: Revised MK spectral standards: stars GO and later. *Bull. Inf. Centre Donnees Stellaires* 29, 25 (1985)
7. Valdes, F., Gupta, R., Rose, J.A., Singh, H.P., Bell, D.J.: The Indo-US Library of Coude Feed Stellar Spectra. *The Astrophysical Journal Supplement Series* 152, 251 (2004)
8. Buchanan, B., Shortliffe, E.: *Ruled-based Expert Systems*. Addison-Wesley, Reading (1984)
9. Dafonte, C., Rodríguez, A., Arcay, B., Manteiga, M., Carricajo, I.: A Comparative Study of KBS, ANN and Statistical Clustering Techniques for Unattended Stellar Classification. In: Sanfeliu, A., Cortés, M.L. (eds.) *CIARP 2005. LNCS*, vol. 3773, p. 566. Springer, Heidelberg (2005)

Spatio-temporal Road Condition Forecasting with Markov Chains and Artificial Neural Networks

Konsta Sirvio¹ and Jaakko Hollmén²

¹ Sirway Ltd., Finland

`konsta.sirvio@sirway.fi`

² Dept. of Information and Computer Science, Helsinki University of Technology, Finland

P.O. Box 5400, FI-02015 TKK, Finland

`jaakko.hollmen@tkk.fi`

Abstract. Preservation of the road assets value in an efficient manner is an important aim for developed road administrations. The task requires accurate road maintenance that is planned in advance. Forecasting road condition in the future is a prerequisite for optimisation of maintenance treatments. In this study two hybrid methods are introduced for forecasting road roughness and rutting. Markovian models outperform artificial neural network models and roughness can be forecast more accurately than rutting.

Keywords: Road condition forecasting, k-means++, Markov chains, neural networks, clustering, machine learning.

1 Introduction

Road networks are major assets for countries providing accessibility and economic foster. The length of public roads comprising 188 countries is over 12 million kilometres [2]. Estimated accounting value of the main road network of Finland administered by the Finnish Road Administration (FRA) is about 15 billion Euros having the total length around 78 000 kilometres [4]. A great task of FRA is to maintain the network properly to preserve the asset value, enhance traffic safety and provide end users an economic means of transportation. To accomplish this task vast amount of data is collected from the network and used for planning and management purposes.

Due to budget constraints only part of the network is annually selected for maintenance treatments according to various criteria such as the sum of discounted cash flow i.e. Net Present Value (NPV). So annual maintenance plans are formed and funding needs are roughly known in advance. The process of road maintenance planing can be strictly divided into two stages: (1) Forecasting road condition and (2) selection of segments for maintenance according to the criteria.

Several condition variables are measured from the paved road network and they vary between countries. Variables represent condition of the road surface or lower layers. Most visible defects are cracks, potholes and rutting. A standardised condition

variable used worldwide is International Roughness Index (IRI) that measures road roughness i.e. driving comfort. Different models for road deterioration are developed to forecast the future values of measured variables. Often these models are linear in nature and they are applied to the whole country [3] while the models change between countries. In this study different computational methods are combined i.e. hybridised and used for forecasting road condition of Southern Finland for the first time.

2 Description of the Data

Data from several databases were combined together for this study. The data can be grouped into road characteristics, condition, road weather and traffic. Road characteristics include basic data on roads such as geometry, pavement type and width. IRI and longitudinal rutting of the road surface were selected of road condition. Condition and geometry values are measured and averages for 10-metre long segments are calculated. Weather data is collected from road weather stations. The data contains temperatures among others. Traffic data is collected by traffic count devices along the main roads. Weather and traffic data are for fixed spatial coordinates while the other data are for segments of roads [9]. Table 1 summarises the input variables. The figures in parentheses for number of input variables concern the neural network method.

Table 1. Description of the input variables

Variable type	Unit	Number of variables
IRI	mm/m	66
Rutting	mm	66
Curvature	°/km	33
Gradient	%	33
Number of lanes	-	1
Speed limit	km/h	1
Pavement type	-	1
Construction year	-	1
Average air temperature	°C	1 (3)
Average road temperature	°C	2 (6)
Average ground temperature	°C	2 (6)
Average of average wind speeds	m/s	1 (3)
Average of maximum wind speeds	m/s	1 (3)
Average wind direction	°	1 (3)
Average air humidity	%	1 (3)
Sum of rainfall	mm	1 (3)
Percentage of freezing temperatures	%	5 (15)
% of temperatures changes from °C > 0 to °C < 0	%	5 (15)
Number of vehicles in 7 different categories	-	7 (21)
Maximum number of vehicles per hour	1/h	7 (21)

All the data is combined into a matrix and pointwise data is taken from the closest measuring device. The selected data contains 126 510 rows with 302 variables for clustering and Markov Chain forecasting and 404 variables for neural network forecasting. Four cases were selected having all the data from 4 surveys in 2005-2006 [8].

The output variables that were forecast were IRI measured under the left and right tyres of the survey vehicle as well as rutting depth from the both sides. Spring survey data of 2007 was used to compare with the forecast results.

3 Research Methods

3.1 Clustering with k-means++

Clustering is used to find common features of a high-dimensional data grouping similar multidimensional data points together into clusters. A popular method has been k-means clustering algorithm due to simplicity and computation speed. In this algorithm k number of clusters and their centre points are selected. Then the data is browsed through and assigned to one of the clusters so that the error term is minimised. Slightly varied version k-means++ differs from the k-means only by the selection of the initial cluster centres. The detailed description can be found in [1].

3.2 Markov Chains

Markov chains can be used to depict a dynamic system where the state S_i of the system changes dynamically in time and thus the system becomes a chain of stochastic variables in time. In order to be called Markov chains the random variables have to satisfy the Markov property in Equation 1. According to the property the current state depends only on the previous state of the system.

$$P(X_n = x | X_{n-1} = x_{n-1}, \dots, X_1 = x_1) = P(X_n = x | X_{n-1} = x_{n-1}) \quad (1)$$

Transition probabilities from a state i to state j are defined by Equation 2.

$$p_{ij} = P(X_n = j | X_{n-1} = i) \quad (2)$$

3.3 Principal Component Analysis

Principal Component Analysis (PCA) linearly transforms the feature space and reduces the dimensionality of the space by selection of the most intrinsic dimensions after the transformation [7].

3.4 Multilayer Perceptron (MLP) Neural Network

MLP artificial neural networks are computational methods to model complex relationships between inputs and outputs of a system. The network consists of k layers of neurons that are the basic computational units. Neurons of the first hidden layer get inputs x weighted by w as their inputs. The output of the network with 1 hidden layer can be calculated by Equation 3 [7], which is used in the experiments.

$$y = \sum_{j=1}^n w_{2j} f_j \left(\sum_{i=1}^m w_{1i} x_i \right) + \lambda \sum_{k=1}^W w_k^2 \quad (3)$$

Each neuron in a layer is associated with a function f that is used in calculating the output of the neuron. Typically the non-linear functions are hyperbolic tangent or logistic. MLP neural networks require supervised training so that the weight vectors are calculated by minimising the error term between the given output and network generated output with given inputs and calculated weights. Ideally the network should have good generalisation power for new inputs and produce results of minimal error.

The last term of Equation 3 represents complexity penalty of the network by weight-decay procedure where λ is a regularisation parameter and W is the total number of weights in the network. Having the regularisation parameter greater than zero complexity of the network is penalised [5].

4 Experiments

4.1 Predicting Road Conditions by a Hybrid Markov Chain Model

The data set was first pre-processed by normalisation and then clustered with k-means++ -clustering algorithm into 5, 10, 15, 20 and 25 clusters. Five simulations were run for each number of clusters. Transition probabilities from a state of road conditions to the subsequent were calculated by dividing the roughness variable into bins of width of 0.2 and rutting to bins of 0.5. Predictions are calculated as weighted average of the state transition probabilities and mean value of the state.

Four roads were selected: 1222 (36.71 km), 3111 (9.28 km), 4111 (29.9 km), 7111 (37.29 km), where the numbers indicate road, carriageway, lane and survey direction in the given order. Cluster-specific transition probabilities were used to calculate the condition one time step ahead e.g. spring 2007. The forecast and surveyed values were compared and the mean square errors (MSE) are presented in Table 2 for left and right IRI. Here k signifies number of clusters and MM stands for a Markovian model. In P1 state transition probabilities were calculated with the data concerning the variable and surveys in question.

Previously used model in the pavement management system (PMS) of FRA is marked as PMS92 and the IRI forecasts are calculated by adding IRI change calculated by Equation 4 to the current values [3].

$$\Delta IRI = 0.016 + 0.0524 * IRI(t) \quad (4)$$

New implemented models for asphalt concrete roads are presented by PMS05 and the values are calculated by Equation 5, where T indicates the number of years from the previous maintenance treatment and ADT average daily traffic [3].

$$\Delta IRI = 0.35 * \frac{IRI(t)}{T} - 0.014 * \log_{10}(ADT) + 0.04 \quad (5)$$

Table 2. Mean Square Errors (MSE) for forecast IRI values. P2 calculated with all the data.

	Road 1222		Road 3111		Road 4111		Road 7111	
	IRI left	IRI right	IRI left	IRI right	IRI left	IRI right	IRI left	IRI right
Real IRI average	1.19	1.16	1.07	1.20	1.10	1.09	1.22	1.29
MM,k=1,P2	0.33	0.28	0.13	0.17	0.20	0.23	0.45	0.48
MM,k=1,P1	0.30	0.28	0.14	0.17	0.19	0.23	0.42	0.48
MM,k=5,P1	0.31	0.28	0.14	0.17	0.18	0.23	0.43	0.47
MM,k=10,P1	0.31	0.27	0.16	0.17	0.19	0.23	0.42	0.47
MM,k=15,P1	0.32	0.28	0.14	0.21	0.19	0.24	0.45	0.49
MM,k=20,P1	0.31	0.27	0.15	0.20	0.19	0.24	0.45	0.49
MM,k=25,P1	0.31	0.27	0.14	0.22	0.20	0.24	0.45	0.49
PMS92	0.46	0.39	0.15	0.19	0.23	0.31	0.69	0.73
PMS05	0.46	0.40	0.15	0.19	0.23	0.32	0.77	0.97

Generally the best performers are Markovian models with maximum of 5 clusters while the worst performers are PMS models. Similarly for right IRI the best performers are Markovian methods up to 10 clusters. In most cases the new PMS model seems to perform worst. Table 3 summarises results of the same methods used for forecasting left rutting values. When rutting values were available rutting changes from the previous surveys in PMS92 –model were calculated by using Equation 6 [3].

$$\Delta RUT = \frac{RUT(t) - 2}{T} \tag{6}$$

When rutting data was not available for a certain road segment the values were calculated by Equation 7 [3].

$$\Delta RUT = -3.34 + 0.57 * \log_e(ADT) \tag{7}$$

In PMS05 rutting depths were calculated by Equation 8 [3]. The linear models already in use outperform Markovian models in most of the cases. And such is the result also with right rutting values.

$$\Delta RUT = 0.49 * \frac{RUT(t)}{T} + 0.13 * \log_{10}(ADT) - 0.45 \tag{8}$$

4.2 Neural Network Generated Road Conditions

Two basic cases were selected for neural network forecasting: (1) training was performed with data consisting of three survey sets on the left side of Figure 1 and (2) training data contained only two previous survey data sets on the right side of Figure 1. Validation was done by each time leaving 1/10 of the data out for validation purposes and the rest was used for training. The 10-fold cross-validation [6] was then repeated for 5 times.

Table 3. Mean Square Errors for forecast rutting values. P2 calculated with all the data.

	Road 1222		Road 3111		Road 4111		Road 7111	
	Rut left	Rut right	Rut left	Rut right	Rut left	Rut right	Rut left	Rut right
Real rut average	7.01	7.04	8.38	7.24	9.79	8.93	8.11	6.86
MM,k=1,P2	7.29	7.36	18.98	14.18	17.64	16.88	15.36	10.05
MM,k=1,P1	6.82	7.22	18.54	14.28	15.86	16.75	14.74	9.59
MM,k=5,P1	7.22	7.50	17.94	12.18	17.18	20.62	14.64	10.02
MM,k=10,P1	7.38	7.82	19.15	13.71	20.99	24.27	15.08	10.68
MM,k=15,P1	7.64	8.79	15.66	13.02	30.53	31.20	17.61	12.12
MM,k=20,P1	7.67	8.76	15.76	13.37	30.95	31.74	17.50	12.25
MM,k=25,P1	7.79	8.92	17.34	13.21	29.82	27.44	17.65	11.79
PMS92	5.49	6.13	15.26	14.15	10.63	11.72	15.85	10.89
PMS05	5.35	5.86	15.89	14.46	10.82	11.91	15.53	10.64

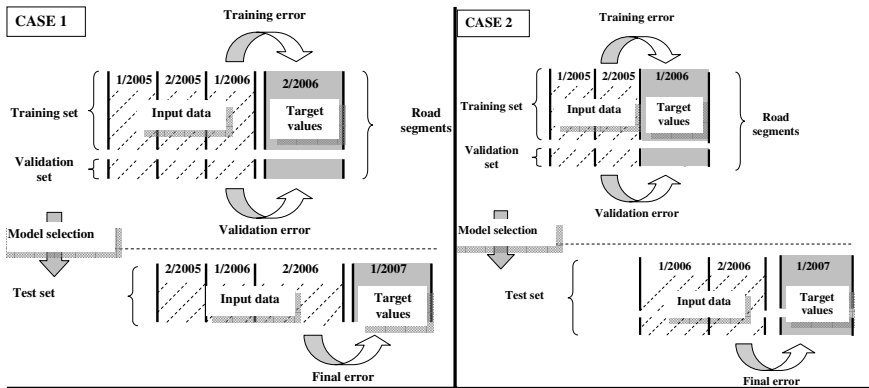


Fig. 1. Training, validation and test sets of the road survey data

As the number of input variables was high the dimensions were reduced by using the first 32 principal components resulting now approximately at 80 % of the variance of the data. Number of hidden neurons was set to 20 and number of training epochs to 150. Bayesian regularisation back propagation was used as the training method [5]. Logarithmic sigmoid functions were used as transfer functions and linear activation function were used at the output layer.

Training, validation and testing MSE are presented in Table 4 for left and right IRI values. In most of the cases all the errors are smaller when input data is formed of the surveys of the previous 2 surveys. However, neural network method does not seem to outperform neither Markovian nor the traditional linear models. This may be due to overly complex prediction functions of MLP neural network.

Table 4. Mean Square Errors (MSE) for forecast IRI values

	Road 1222		Road 3111		Road 4111		Road 7111	
	IRI left	IRI right	IRI left	IRI right	IRI left	IRI right	IRI left	IRI right
ANN,case 1 (training)	0.09	0.04	0.05	0.01	0.05	0.01	0.09	0.04
ANN,case 1 (validation)	0.51	0.39	0.53	0.23	0.43	0.19	0.46	0.47
ANN,case 1 (testing)	0.87	0.29	5.72	2.71	6.06	0.36	0.54	0.53
ANN,case 2 (training)	0.05	0.02	0.08	0.04	0.05	0.07	0.10	0.20
ANN,case 2 (validation)	0.22	0.19	0.35	0.39	0.24	0.23	0.53	0.55
ANN,case 2 (testing)	0.31	0.29	0.33	0.41	0.23	0.26	0.49	0.50

Table 5 contains errors for forecasted rutting values and here the experiments with input data of three previous surveys outperform when validation error is concerned, but the testing error is lower when input data of 2 previous surveys are selected. In some cases the test error is lower than with the Markovian or linear models.

Table 5. Mean Square Errors (MSE) for forecast rutting values

	Road 1222		Road 3111		Road 4111		Road 7111	
	Rut left	Rut right	Rut left	Rut right	Rut left	Rut right	Rut left	Rut right
ANN,case 1 (training)	2.70	1.05	0.44	0.40	0.75	0.46	0.16	0.36
ANN,case 1 (validation)	7.32	8.51	2.40	2.23	7.61	8.59	10.11	1.63
ANN,case 1 (testing)	15.29	14.47	36.92	27.47	13.72	26.33	11.72	33.84
ANN,case 2 (training)	0.19	0.25	0.84	0.28	0.97	0.97	1.20	0.48
ANN,case 2 (validation)	2.91	3.56	7.14	6.74	3.92	8.34	13.34	6.73
ANN,case 2 (testing)	14.26	15.67	26.03	17.76	27.86	25.27	20.89	12.55

5 Discussion

PMS models were created by and are used for data of summer surveys averaged by every 100 metres while here they were tested with data surveyed in spring and autumn averaged by every 10 metres and therefore the results may not be as good as with the data the models were originally designed for.

Markovian models give better forecasting results for IRI than currently used models. Markov models are considerably faster and simpler than neural networks and results at better forecasts according to this study. One reason can be the fact that the networks get over-trained and do not generalise well enough with changed input data. This can be noted by the difference between validation and testing MSE. The reason can be high noise level in the measured data or low impact of the measured variables on road condition. There may be other variables having greater impact on the future

condition. Such missing variables are for example detailed road geometry and drainage condition [9]. The problem with Markovian models is lack of data of roads in poor condition and thus the results become biased and long-term forecasting becomes unreliable. Partly this could be rectified by using geographically larger sample.

Being simple and relatively fast the future research of road condition forecasting should emphasise Markovian models. Different clustering techniques could be used in grouping similar road segments and thus trying to gain more accurate results. This requires temporally and spatially larger data sets. Artificial neural networks proved to be time-consuming to compute and the results were not as satisfactory as with the Markovian models. However, it might be worthwhile to test simple neural networks with only a few important input variables such as IRI, rutting and drainage condition.

Acknowledgements

We are grateful for help from I. Iso-Heiniemi, T. Toivola, M. Hämäläinen, M. Varis, J. Meriläinen, J. Laakkonen, E. Pilli-Sihvola, J. Kantonen, E. Kenttälä and J. Torniainen, J. Juutinen, M. Knuuti, J. Äijö and Helsinki University of Technology for the scholarship.

References

1. Arthur, D., Vassilvitskii, S.: k-means++ The Advantages of Careful Seeding. In: Symposium on Discrete Algorithms (SODA) (2007)
2. International Road Federation. IRF World Road Statistics 2006. International Road Federation, Geneva, Switzerland (2006)
3. Finnish Road Administration. PMSPro:n kuntoennustemallit 2004. Tiehallinnon selvityksiä 9/2005. Finnish Road Administration, Helsinki, Finland (2005)
4. Finnish Road Administration. Finnish Road Statistics 2006. Edita Prima Oy. Helsinki, Finland (2007)
5. Foresee, D., Hagan, M.: Gauss-Newton Approximation to Bayesian Learning. In: Proceedings of the International Joint Conference on Neural Networks (June 1997)
6. Hand, D., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press, Cambridge (2001)
7. Haykin, S.: Neural Networks. A comprehensive foundation. Prentice-Hall, London (1999)
8. Kuntorekisteri (Road condition database), Tiesää tietokanta (Road weather database), LAM tietokanta (Database of automatic traffic counts), Finnish Road Administration (2008)
9. Saarenketo, T.: Developing drainage guidelines for maintenance contracts. Results of a ROADEX III pilot project in Rovaniemi Maintenance Area in Finland. Roadscanners Oy. Rovaniemi, Finland (2007)

Parameter Extraction from RVS Stellar Spectra by Means of Artificial Neural Networks and Spectral Density Analysis

Diego Ordóñez¹, Carlos Dafonte¹, Minia Manteiga², and Bernardino Arcay¹

¹ Information and Communications Technologies Department, Faculty of Computer Science, University of A Coruña, 15071, A Coruña, Spain
{dordonez, dafonte, cibarcay}@udc.es

² Department of Navigation and Earth Sciences, University of A Coruña, 15011, A Coruña, Spain
manteiga@udc.es

Abstract. The Gaia Galactic Survey Satellite is among the key future missions of the European Space Agency, who plans its launch near the end of 2011. Gaia will carry out what is being called a “Galaxy Survey”, gathering exact information on the nature of the galaxy’s main constituents, including the spectra of objects. A first approach to the extraction of atmospheric parameters from spectra (Effective Temperature (Teff), Gravity (log G), Metallicity ([Fe/H]), and Alpha elements abundance ([α /Fe])) was their indirect measurement by means of the Morgan-Keenan (MK) classification system. However, traditional methods based on MK are slow and subjective. This is why an automatic and robust method has become an essential requirement. This work presents the preliminary results of a study on the automatic extraction potential of the main atmospheric stellar parameters: Teff, log G, [Fe/H], and [α /Fe] in the spectral region of the RVS, between 8470 and 8740 Å. The study was carried out by means of an algorithm based on artificial neural networks which is a technique widely used in that kind of automatic spectral parameterisation.

Keywords: stellar spectra, signal processing, spectral density, FFT, SNR, Artificial Neural Network, feed-forward, backpropagation, wavelet.

1 Introduction

Spectral parameterization is a well-known problem in astrophysics, and many previous studies have worked with a wide range of data sources [1], [2], [3], [4], [5], [6], [7], [8]. The goal of the study is the electromagnetic radiation spectra which is radiated from stars and other objects. The spectroscopy can be used to guess many star properties and distant galaxies such as the chemical composition of their atmosphere. A first approach to the extraction of atmospheric parameters was their indirect measurement by means of the Morgan-Keenan (MK) classification system [9]. Each spectral type of the MK system is a description of the spectrum that, together with the identified luminosity level, is directly related to the temperature and gravity of the

star. The advantage of this model is that it functions well, even with low resolution spectra, and that it does not depend on other models.

However, traditional parameterization methods such as the above, based on MK, are slow and subjective. This is why an automatic and robust method has become an essential requirement, both for the homogeneity of the results and the repeatability of the process. The currently existing (and planned) data sources have also triggered this interest for automatic classifiers.

Modern telescopes are made with spectrometers that cover a large number of objects per frame. The current and future data sources and spatial missions, such as the Sloan Sky Survey or the GAIA mission (launch planned for 2011), will gather large amounts of stellar spectra that belong to various components of our galaxy. Such an input of data can only be managed with automatic means.

Historically, the techniques that have most often been applied for automatic spectra parameterization are artificial neural networks and the minimal distance methods. Various studies have tried to determine the physical parameters of stellar spectra using artificial neural networks (ANN) and synthetic data sets [2], [3], [13]).

The launch of ESA's GAIA satellite [10] is planned for 2011-12. We must follow a complex and long data analysis in order to transfer the GAIA signal to physical and positional parameters that scientists can use to test the variable nature of the astronomical objects.

The main objective is to determine the stellar atmospheric parameters, particularly effective temperatures, superficial gravities, metallicities, possible overabundances of alpha elements, and individual abundances of certain chemical elements.

The manipulation, analysis, and classification of all the information concerning the visible celestial bodies up to magnitudes 17-18 is undoubtedly a challenge for both Astrophysicists and Computer and Artificial Intelligence Scientists. This work presents our first results on the automatic obtention of atmospheric stellar parameters in the spectral region of the RVS by means of ANNs that are trained with synthetic stellar spectra.

2 Synthetic Data

The data that were used proceed from the simulations of stellar spectra in the RVS band for the GAIA mission. The features of the data for the training and test sets are the following:

- Total Number of examples. Training: 9408. Test : 3060
- Wavelength: initial 847.5803, final 873.5999, wavelength increase 0.0268 (nm), number of points per spectrum 971.
- Ranges of the parameters: See table 1 for the parameters of the training spectra. The range is identical for the test case, but the values are interpolated, which means that the temperature values are not necessarily, for instance, 4500, 4750,.. Since there could exist a value 4623.

In a real situation, the spectra satellite is not likely to collect spectra without any distortions, so we must always take the noise into account. The training data have been considered both clean and with noise, at various SNR ratios: 5, 10, 25, 50, 75, 100, 150, 200, 100000. The lower the ratio, the more noise; considering all the data,

Table 1. Parameter ranges

Heading level	Example
Teff	4500 7750 4500 4750 5000 5250 5500 5750 6000 6250 6500 6750 7000 7250 7500 7750
Log G	-0.5 5 -0.5 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
[Fe / H]	-5 1 -5 -4 -3 -2 -1.5 -1 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 1
[α /Fe]	-0.2 0.4 -0.2, 0, 0.2, 0.4

we can observe how noise affects the prediction of parameters and how we can mitigate its effect by more or less detecting its intensity with regard to the signal. This is described in the following section.

3 SNR Detection

The reliability of the parameterization method depends largely on the wavelength coverage, the spectral resolution, and the SNR. From the point of view of the design, we must know the quality of the parameterization for a given set of observations. The wavelength coverage and the resolution are determined beforehand, so the most important problems to be solved are the nature and the intensity of the noise [3].

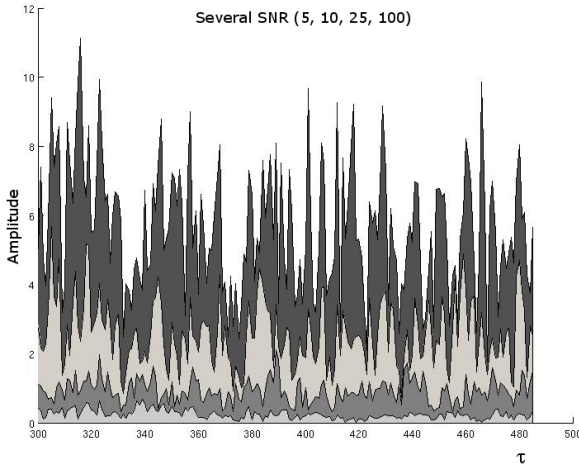


Fig. 1. The same sample with several SNRs (5,10, 25,100). τ is the transformed variable.

If we suppose an additive white Gaussian noise, we know that its spectral density is continuous in all the frequencies. Considering both this aspect, and the fact that the Fourier transform of the analysed stellar spectra is band limited, we designed a mechanism that detects the SNR of a given spectrum. The spectral density decreases inversely with the frequencies, and since the spectrum is band limited, it is entirely cancelled in the highest frequencies. The analysis for noise level detection is based

precisely on the assumption that the highest frequencies correspond with spectral components that taxonomize the noise.

Figure 1 shows the described behaviour: a signal with more noise has more intensity values in the highest frequencies.

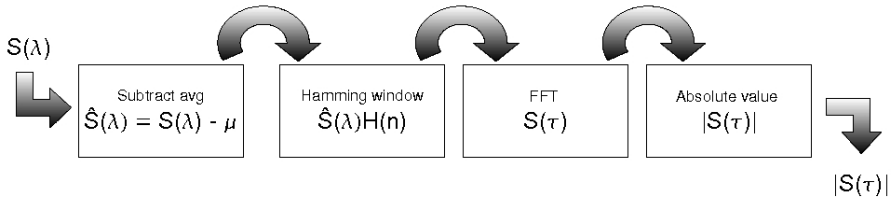


Fig. 2. Preprocessing algorithm

Figure 2 presents the processing algorithm that was created to prepare the detection of the noise level. The input to the algorithm is the signal in wavelength, the value in each λ corresponds to a flux value of the spectrum; as we can observe, it consists in four phases. The first processing step is the subtraction of the spectrum average, because it does not contain any relevant information and introduces undesired spectral components. The second processing step is multiplication by a window, as a step previous to the Fast Fourier Transform (FFT) algorithm; in an ideal situation, the wavelength of our signal would extend between $-\infty$ and ∞ , but since this is impossible, the signal is limited by two given wavelengths and we obtain only intermediate values. In this case, we can see the rectangular window and the unity value between the described points. A window of this type has the disadvantage of contributing in all the frequencies by its border effect. Other types of windows (Hann, Hamming, Blackman, Flat-top, Gauss, Bartlett, etc.) try to smoothen the border transitions. Various window configurations have also been tested, with the Hamming configuration giving the best results for this type of analysis.

The third step consists in applying the Cooley-Tukey algorithm [11] in order to obtain the FFT. It is important to point out that the X axis of the analysed signal does not refer to the time but to wavelengths, and that it is therefore not correct to speak in terms of frequencies in the transformed domain. The transformed variable is called τ (see Figure 2): if it were a temporary variable, we would speak of frequencies. The last step in the process consists in calculating the absolute value of the signal points. The purpose of this step is to work only with real numbers, since the FFT algorithm produces values that belong to the domain of the complex numbers.

The algorithm that detects the SNR consists of two phases. During the first phase, it calculates the integral of the N last points of the spectrum (by means of a trapezoidal method). During the second phase, it compares this result with some ranges assigned to each SNR. We extract these ranges from the analysis of the complete training set, for which the SNR is known. According to the value of the integral, we more or less know the SNR of the signal (SNR classification) and we can apply a specific algorithm.

This detection constitutes an important step, because we will apply certain pre-processing techniques according to the amount of noise present in the signal.

4 Parameterization Algorithm

The parameterization algorithm is based on a system of *feed-forward* neural networks trained with the error *backpropagation* algorithm [12]. The *network design* is as follows:

- **Input layer**, the number of process elements in this layer is determined by the preprocessing of the spectra and varies from **385** (fourier processing, removing highest frequencies) to **971** (wavelength domain) process elements, and activation function of the linear type.
- **Hidden layer**, variable range (between **100** and **200** process elements), and activation function of the tangent hyperbolic type.
- **Output layer**, only **one** process element, the output value provides us with one parameter value. The output activation function is of the sigmoid type, which provides values between 0 and 1.

In this sense, it is obvious that we will need to train an ANN for each parameter that is to be predicted: Effective Temperature (Teff), gravity (log G), metallicity ([Fe/H]), and abundance of alpha elements ([α /Fe]). We will need to select four networks in order to obtain the complete set of parameters for a given spectrum.

The network will provide us with the parameterization, but the quality of the adjustment depends to a large extent on the selection of the network training set and the specific preprocessing for each SNR case. The noise level detection algorithm allows us to select the training set. Figure 3 shows an example of the complete parameterization process. We start by detecting the noise level, applying the specific preprocessing, and obtaining parameters with the neural network. The network is specialized in two senses: on the one hand, each network is able to extract a single parameter; on the other hand, the quality of the results is directly related to the noise level of the data used for the ANN training (we dispose of different ANN for different noise levels).

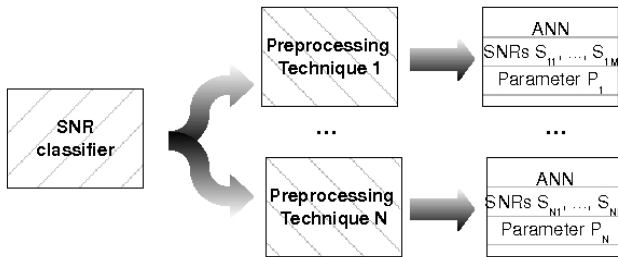


Fig. 3. The Parameterization algorithm steps

With clean or almost clean spectra, the combination of the above processing algorithm and a neural network for each parameter that is to be predicted gives good results, as can be seen in Figure 4. However, the results are less satisfactory for signals with a high SNR ratio in transformed domains, because the noise distorts the spectrum too much and is not band-located (if it were, we could solve the problem with a

band-pass filter), we are using this information to perform the SNR detection, because in highest frequencies we have *only* noise and we can characterize it.

We must also consider the fact that when we use the described processing technique, as a step prior to parameterization with the neural network, we are not taking into account the last points of the spectrum, since, as mentioned above, these correspond to spectral components of the noise.

5 Spectral Parameterization Results

Neural Networks represent a parameter adjustment technique in which it is essential to work with independent training data that can be evaluated. We work with data from 9408 spectra for training and data from 3060 spectra for validation (see section 2).

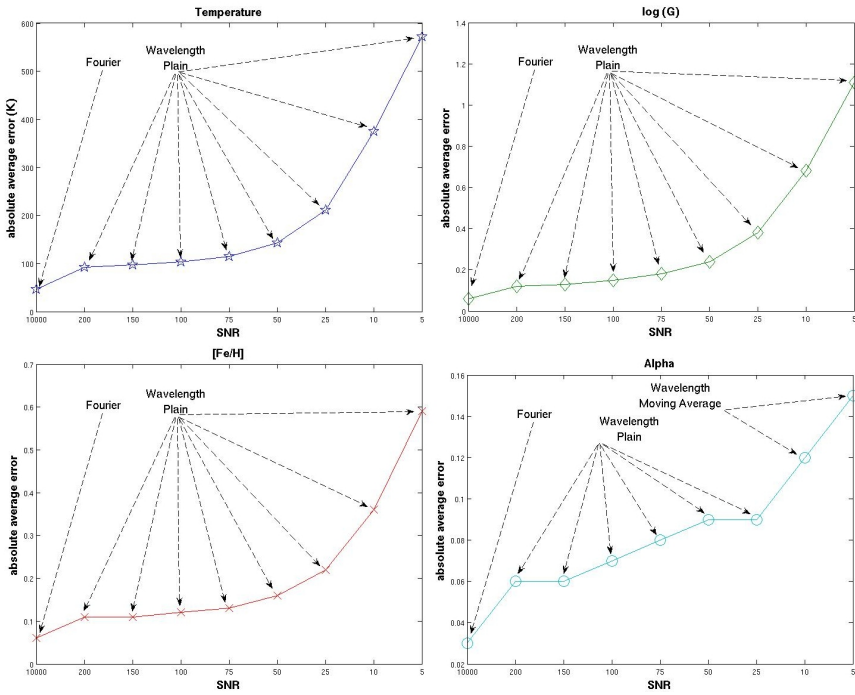


Fig. 4. Results per parameter

We have applied various preprocessing techniques to the spectra and trained networks with the resulting data sets. One of these techniques is the application of the algorithm described in section 3 (SNR level detection) to the input data; we have also tried to train the spectra using the wavelength values as input (without preprocessing), and in cases of low SNR we have filtered the spectra (*moving average*).

The values of the parameters of the spectra that were selected for the training set are described in Table 1. The distribution of the values that were used for the training introduces an error in the learning process of the network, because during the training

they are sampled at fixed intervals (e.g. at intervals of 250 grades Kelvin). On the other hand, the test spectra are interpolated, the values of the parameters are not discrete and therefore adjust better to reality. Even so, the relation found by the neural network to make an input correspond to an output is an interpolation of the spectra selected for the training: the finer the sampling of the parameters, the more realistic the parameterization. This intrinsic interpolation of the algorithm allows us to obtain smaller errors than the sampling distance between parameters, as can be observed in Figure 4.

The error measure that was applied is the average error in absolute value for each parameter: the error measures refer to the difference between the value of the observed parameter and the value obtained by the network. The results of the parameterization process can be observed in Figure 4. For the effective temperature with little noise (SNR 10000), the best results were obtained with the preprocessing algorithm mentioned in section 3, which trains and parameterizes the network in the transformed domain. For the temperature, the best result was an average error of 46 grades Kelvin; the result for the same intensity and trained with the spectra in wavelength was 76 grades Kelvin. As the intensity of the noise increases, the efficiency decreases and the networks trained with spectra in wavelength gain importance. With respect to the gravity, we worked with this parameter in logarithmic units ($\log G$) and obtained the best result in transformed domains, with an average error of 0,06 units; the best results for the metallicity and abundance of alpha elements were 0,06 and 0,03 respectively.

We have carried out tests by applying filters to the spectra in wavelength and obtained remarkable improvements in some cases, such as the case of the abundance of alpha elements in the presence of much noise (SNR 10 and 5), where we obtained the best results. The filtering algorithm was a moving average of five points that acts as a low-pass filter. This test was repeated for all the parameters, and even though it gave similar results, the results were not improved in any of the remaining cases.

6 Conclusions

The results show that a completely automatic system, based on ANN, can successfully determine the main physical parameters of stars on the basis of their spectroscopic data. This work has used spectra in the spectral region of the RVS, between 8470 and 8740 Å and SNR (from 5 to 10000); the ANN can approach very closely the real parameters in the presence of noise. However, the further the objects, the less precision in the parallaxes and hence in the flow level measurements. By ignoring this problem, the shape of the spectrum can still be a good indicator of, for instance, the temperature, and the networks will continue to function well [3].

We can also deduce from these results that the SNR definitively influences the learning process, and that characterizing the noise and training with noisy spectra are essential to improve the functioning of the networks.

The good results obtained parametrising clean spectra training neural networks in transformed domains (FFT) open the door to future explorations with noisy spectra.

In this sense, future work will be oriented towards studying the behaviour of the networks by means of the transformed wavelet and as such expose those features of the data that allow us to improve the precision of the parameterization.

Acknowledgment. This work is supported by the Spanish Ministry for Science and Education, project MEC ESP 2006-13855-C02-02.

References

1. Bailer-Jones, C.A.L.: A method for exploiting domain information in astrophysical parameter estimation. In: *Astronomical Data Analysis Software and Systems XVII*. ASP Conference Series, vol. XXX, London (2008)
2. Kaempf, T.A., Willemsen, P.G., Bailer-Jones, C.A.L., de Boer, K.S.: Parameterisation of RVS spectra with Artificial Neural Networks – First Steps. In: *10th RVS workshop*, Cambridge (September 2005)
3. Bailer-Jones, C.A.L.: Stellar parameters from very low resolution spectra and medium band filters. *Astronomy and Astrophysics* 357, 197–205 (2000)
4. Christlieb, N., Wisotzki, L., Graßhoff, G.: Statistical Methods of automatic spectral classification and their application to the Hamburg/ESO survey. *Astronomy and Astrophysics* 391, 397–406 (2002)
5. Von Hippel, T., Allende, C., Sneden, C.: Automated Stellar Spectral Classification and Parameterization for the Masses. In: *The Garrison Festschrift conference proceedings*, June 10 - 11 (2002)
6. Allende, C., Rebolo, R., Garcia, R., Serra-Ricart, M.: The INT Search for Metal-Poor Stars. Spectroscopic Observations and Classification via Artificial Neural Networks. *The Astronomical Journal* 120, 1516–1531 (2000)
7. Fiorentin, P.R., Bailer-Jones, C.A.L., Lee, Y.S., Beers, T.C., Sivarani, T., Wilhelm, R., Allende, C., Norris, J.E.: Estimation of stellar atmospheric parameters from SDSS/SEGUE spectra. *Astronomy and Astrophysics* 467, 1373–1387 (2007)
8. Recio-Blanco, A., Bijaoui, A., de Laverny, P.: Automated derivation of stellar atmospheric parameters and chemical abundances: the MATISSE algorithm. *R. Astron. Soc.*, 1–11 (2002)
9. Morgan, W.W., Keenan, P.C., Kellman, E.: *An atlas of stellar spectra with outline of spectral classification*. Astrophys. Monographs. University of Chicago Press (1943)
10. GAIA: The galactic census problem, <http://www.rssd.esa.int/gaia/>
11. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* 19, 297–301 (1965)
12. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* 323, 533–536 (1986)
13. Harinder, P., Gulati, R.K., Gupta, R.: Stellar Spectral Classification using Principal Component Analysis and Artificial Neural Networks. In: *MNRAS*, vol. 295, pp. 312–318 (1998)

Supervised Classification Fuzzy Growing Hierarchical SOM

Rafael del-Hoyo¹, Nicolás Medrano², Bonifacio Martín-del-Brio², and Francisco José Lacueva-Pérez¹

¹ Instituto Tecnológico de Aragón, C/ María Luna n°6,
50005 Zaragoza, Spain

{rdelhoyo, fjlacueva}@ita.es

² Departamento de Electrónica y Comunicaciones, University of Zaragoza,
C/ María Luna 1, 50005 Zaragoza, Spain

{nmedrano, bmb}@unizar.es

Abstract. This paper introduces a fuzzy-extension of the Kohonen Self Organizing Map model called Fuzzy Growing Hierarchical SOM that is able to extract Fuzzy rules in hierarchical way. The main idea of the FGHSOM is to provide an architecture that can be initialized with prior knowledge and without, and can be trained directly using SOM learning methods. The training is carried out using competitive methods in such a way that the learning result is interpretable in the form of linguistic fuzzy if-then rules and rules are organized in a tree-like structure. The structure allows increasing the information using parent/child relationships. The FGHSOM is successfully compared with different neuro-fuzzy algorithms in different classification problems.

1 Introduction

An automated classification system must present high flexibility in its structural implementation. In a real situation it should learn from data, extracting knowledge and incorporating both the initial set of knowledge and rules for a future human supervised explanation and understanding.

Neuro-Fuzzy algorithms have emerged as a new and very powerful technique and they have shown in many applications their capability for resolving, in real situations, supervised classification problems. They are hybrid systems that combine the learning capacity of neural nets with the linguistic feature of fuzzy inference systems. There are several neuro-fuzzy networks proposals in the literature. Most significant algorithms are FALCON [1], ANFIS [2], GARIC [3], EFUNN [4].

The strength of neuro-fuzzy systems involves two contradictory requirements in fuzzy modeling: interpretability versus accuracy. In practice, one of the two properties prevails. A typical fuzzy interpretable Mamdani rule is like:

if XI is A and X2 is B then Z is C,

Where XI and X2 are features or input variables and Z is the output variable; A, B, C are linguistic terms characterized by appropriate membership functions. However, the most important neuro-fuzzy algorithms generate Sugeno rules, more difficult to

understand than Mamdani, but with better accuracy. In fact, automatic extraction of rules is valuable in the meaning that the extracted rules are meaningful and comprehensible to human observers, like Mamdani kind of rules. Unfortunately, most of the existing neuro-fuzzy systems exhibit several major drawbacks. These drawbacks are the curse of dimensionality (i.e., fuzzy rule explosion), not a membership function transparency and their lack of ability to extract knowledge from a given set of training data.

The proposed solution is based on balancing precision level, transparency and number of rules. Fuzzy Growing Hierarchical Self Organizing Map (FGHSOM) defines a modulator mechanism by means of horizontal growth and hierarchical rules generation. Hierarchical rule-extraction and controlled evolution of rules can reduce the rule number explosion for this kind of methods. It will be shown that this system offers a reasonable trade off between the approximation quality, the complexity and the transparency of the model.

FGHSOM is built in Kohonen SOM model [5] and it has been developed by extending the description of the ESOM available in [6] and making it WEKA compatible for analysis benchmarking. Sect. 2 contains an outlook of the FGHSOM and its learning schemes. In Sect. 3, we describe the problems selected to represent a hierarchical rule extraction for classification. Finally, the results of applying FGHSOM to several problems are presented in Sect. 4.

2 Fuzzy Growing Hierarchical SOM (FGHSOM)

The FGHSOM model is a Kohonen SOM extension that allows hierarchical fuzzy rules generation and introduces a final associative supervision during the learning process by means of local tuning in each rule weights.

There are several different proposals in self organizing map rule extraction literature, as neuro-fuzzy variants from the model [7,8,9]. One of the most interesting proposals is the Nomura's one [10]; having the fix map as a handicap, this leading to a limit in the number of rules.

The FGHSOM performs a fuzzy partitioning of the input space, in similar way like fuzzy k-means and maximum entropy approach based on competitive learning like Kohonen SOM. In FGHSOM each neuron weights are Gaussian member functions, learning the centers of clusters and deviation around them using a competitive learning. Furthermore when rules are extracted, each SOM node represents a single fuzzy rule. One of the shortcomings of the Kohonen SOM approach lies with its fixed architecture, obtaining so many rules as neurons have the map.

In our proposal, we use a hierarchical growing variant of the Kohonen SOM [11] that allows a hierarchical growth of the rules. The FGHSOM algorithm is a new architecture which tries to reach a balance between precision (a greater precision needed implies a greater complexity of the hierarchical rule tree) and transparency (in the top level of the hierarchical rule tree, we have a better understanding of the system). Thus, going to deeper levels in the rule tree allows an accurate solution of the problem, with the drawback of having not such a clear vision.

Let's denote 'i' as the neuron label, and 'w_i' the corresponding weights or prototypes associated to i, Consider I the input space of the problem, F_{ij} is the one-dimensional Gaussian membership function (fig 1), assigned as the weight between the j-th input node and the i-th the output node:

$$w \in \mathfrak{R}^m \quad I \in \mathfrak{R}^n \text{ Fuzzy Set } F_{ij} = e^{-\frac{1(\mu_{ij}-x)^2}{2\sigma_{ij}}} \tag{1}$$

For an n-dimensional input pattern I, the node output value is given by

$$F_i(I) = \prod_{j=1}^n F_{ij}(I_j) = e^{-\frac{1}{2} \sum_{j=1}^n \frac{(\mu_{ij}-I_j)^2}{\sigma_{ij}}} \tag{2}$$

The prototype associated to a neuron is an n-dimensional Gaussian member ship function with center $\mu_i=(\mu_1, \mu_2, \mu_3, \dots, \mu_n)$ and width $\sigma_i=(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_n)$. Let's note that $F_i(I)$ is the degree of inclusion into the fuzzy set in n-dimensional Euclidean space represented by the membership.

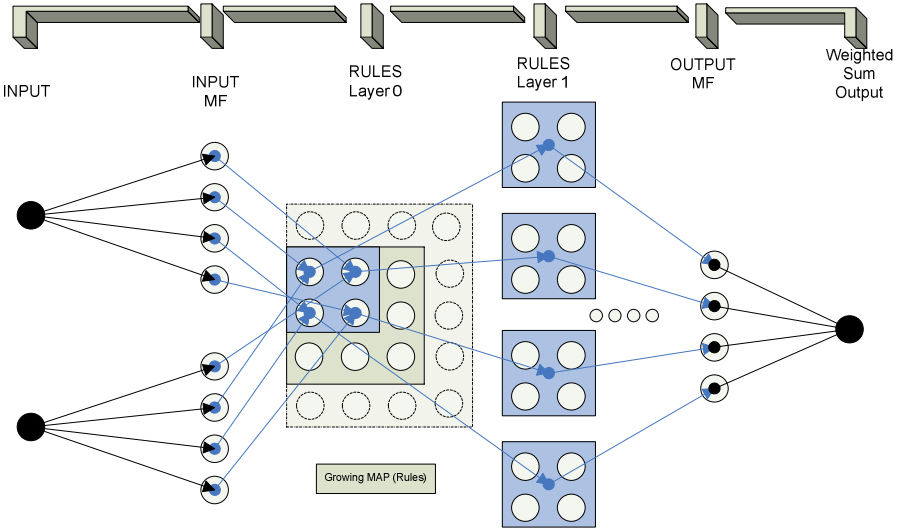


Fig. 1. FGHSOM architecture, it represents the different layers

2.1 The FGHSOM Description

$$dist = \sum_{j=1}^n (\mu_{ij}-I_j)^2 \tag{3}$$

According to [10] is possible to obtain the following update rules [eq 4,5,6] for the parameters by gradient descent.:

$$\mu_{ij}(t+1) = \mu_{ij}(t) + \gamma f_i(I_j - \mu_{ij}) \tag{4}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + 2\gamma f_i \sigma_{ij}(t) \left((\mu_{ij} - I_j)^2 - \sigma_{ij}^2(t) \right) \tag{5}$$

$$f_i = \frac{\left(e^{-\frac{1}{2} \sum_{j=1}^n \frac{(\mu_{ij} - I_j)^2}{\sigma_{ij}}} \right)^{1/h}}{\left(\sum_{\hat{i}=1}^m e^{-\frac{1}{2} \sum_{j=1}^n \frac{(\mu_{ij} - I_j)^2}{\sigma_{ij}}} \right)^{1/h}} = \frac{F_i(I)^{1/h}}{\sum_{d=1}^m F_d(I)^{1/h}} \quad (6)$$

The learning parameters are $\gamma > 0$ and $h > 0$; γ element is associated to the SOM learning algorithm and $h=1$ for FGHSOM model. Each neuron represents a fuzzy rule (if the neuron is not activated no rules exist) [eq. 7]

$$R_i : \text{If } I_1 \text{ is } F_{i1} \text{ and } I_2 \text{ is } F_{i2} \text{ and... } I_n \text{ is } F_{in}, \text{ then } y_j \text{ with weight } w_{ki} \quad (7)$$

The procedure (fig. 2) of training the system and extracting rules is defined in the following sequence:

1. Initialize randomly the parameters μ and σ and set de value $Q_{ki}=0$, it represents the activation done for the winner neuron for each output node of the FGHSOM layer in each F_{IN} [eq. 8].
2. For each k ($k=1 \dots L$) number of neurons and each I_1 (input variable) update de parameters for T times (T is a given integer)

$$Q_{ki}(t+1) = Q_{ki}(t) + F_{ki}(I_1) \quad (8)$$

3. determine the degree of confidence in the rules by $w_{ki} = \frac{Q_{ki}}{\sum_{d=1}^m Q_{kd}}$
4. The output of the system is fitted using a Widrow-Hoff algorithm to obtain the best solution [eq. 9].

$$z = \sum_{j=1}^m F_j w_j y_j \quad (9)$$

5. If the error of z is bigger than \mathcal{E}_1 (grown parameter), a new row is added between the winner neuron and the neighbor neuron closest to the last trained (map grown condition).
6. If the number of patterns associated to a neuron is $> \mathcal{E}_2$ a new fuzzy map is introduced belong to that neuron and rule. All process is repeated for the new map, except the map evolution.

One of the most important differences with the SOM model is the supervised response. The output of the SOM is the index of the winner neuron, but the FGHSOM output is z and it is fitted using Widrow-Hoff algorithm. In addition, the SOM fuzzification not only provides a better solution, but it also makes possible the extraction of underlying classification rules. Appropriate fuzzy inference methods can then

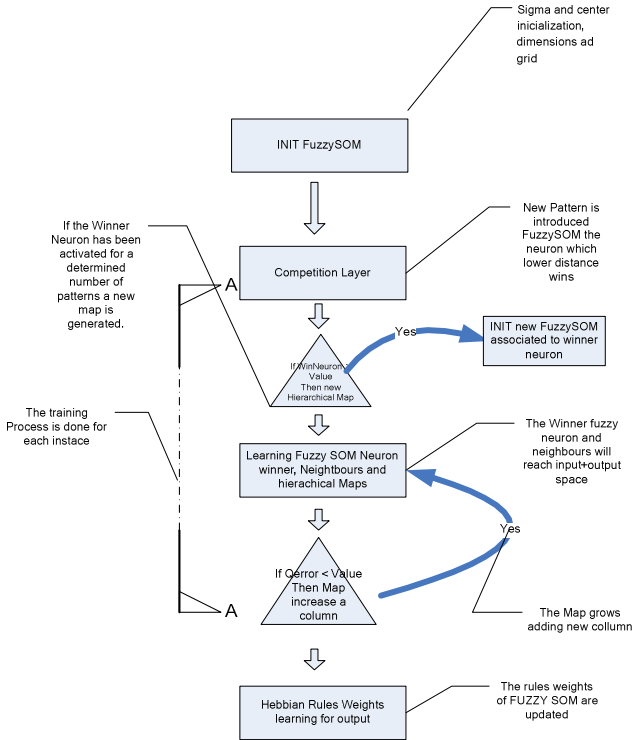


Fig. 2. Sequence diagram of FGHSOM algorithm

be applied over the extracted rules, thus making possible to use a fuzzy rule-based system for the classification task. Finally the SOM hierarchy allows for reducing the rule number and a better rules understanding. The output response error is evaluated as the mean squared error (MSE), so the comparison with other neural models based in the MSE is straightforward. The FGHSOM also presents all the interesting properties of the SOM model, like the possibility of the visual inspection of the topological map representation. For the map training in batch mode and weight initialization we have used the functions available in WEKA and ESOM [11].

3 Case Studies

3.1 Iris Categorization

The Fisher-Anderson iris data consists of four input measurements, sepal length, sepal width, petal length and petal width, on 150 specimens of iris plant. Three species of iris are involved, Iris Sestosa, Iris Versicolor and Iris Virginica, and each species includes 50 instances.

Next we apply the FGHSOM scheme on an Iris classification problem of finding the mapping between the four input variables and the corresponding classes. First of all, the classes are converted into binary values, and one FGHSOM model is built for

each class. Three FGHSOM for each class are generated in order to clearly choose to which class an input belongs. The rules obtained will be in the following type.

Next we apply the FGHSOM scheme on an Iris classification problem of finding the mapping between the four input variables and the corresponding classes. First of all, the classes are converted into binary values, and one FGHSOM model is built for each class. Three FGHSOM for each class are generated in order to clearly choose to which class an input belongs. The rules obtained will be in the following type.

IF sepal length is A and sepal width is B and petal length is C and petal width is D
THEN is Setosa with E

A, B, C and D are fuzzy sets and E is the membership function to class Setosa. Then, a pattern of the dataset will be included in the three classes with different ratings.

There are 150 samples in the data set and we apply 10 Cross Validation method. We constructed three networks and trained them with the learning data. The adjusted networks were then evaluated with the testing dataset. The discriminating power of the classifier was validated, showing the achieved results in table 1. Additionally our fuzzy approach gives insights of the data characteristics.

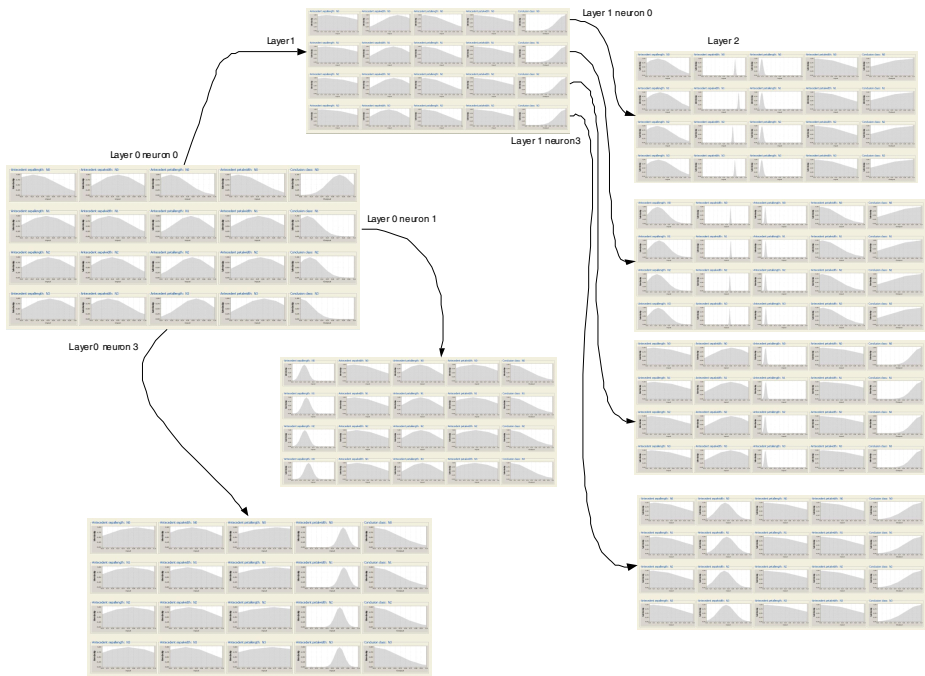


Fig. 3. The figure represents the hierarchical rules description of FGHSOM for Setosa class. The columns rules antecedents correspond with sepal length, sepal width, petal length, and petal width and the class membership.

Figure 3 shows the membership functions and rules for the Setosa class. The FGHSOM shown has three layers; the first layer includes 4 rules (4 neurons in SOM). On the other hand, the main disadvantage of FGHSOM is the number of member functions generated, however the number of member functions can be controlled. The model shown in fig. 3, provides the following linguistic approximation member functions labeled by us:

- Sepal length (slength):** low, medium.
- Sepal width (swidth):** low-medium, medium.
- Petal length (plength):** low, medium. medium-high.
- petal width (pwidth):** low, medium. medium-high.
- Class Setosa:** low, low-medium, high.

Using the assumption labels presented below the rules extracted in layer 0 are:

- R1: IF** slength is low **AND** swidth is medium **AND** plength is low **AND** pwidth is low **THEN** is Setosa: Weight 0.31
- R2: IF** slength is medium **AND** swidth is low **AND** plength is medium **AND** pwidth is low **THEN** is low-medium Setosa: Weight 0.08
- R3: IF** slength is medium **AND** swidth is low **AND** plength is medium **AND** pwidth is low **THEN** is low-medium Setosa: Weight 0
- R4: IF** slength is medium **AND** swidth is low **AND** plength is med-high **AND** pwidth is med-high **THEN** is low Setosa: Weight 0.61

Table 1. Comparison performance for the fghsom classifier in fisher-anderson iris problem

Algorithm	N ^a Rules	Correctly Classified	Incorrectly Classified	% Percent Correctly Classified	% Percent Incorrectly Classified
FGHSOM	76 ,73,78	145	5	96.66	3.33
FGHSOM	38 ,38 ,38	141	9	94	6
FGHSOM	19, 16, 20	135	15	90	10
EFUNN	24,24,24	137	13	91.33	8.66
EFUNN	60,60,60	140	10	93.333	6.66
MLP		146	4	97.33	2.66
J48	9	144	6	96	4
JRIP	3	140	10	93.33	6.66

As we can see, Rules 2 and 3 are identical but the weight associated to the rule 3 is 0. If the rules are compared with J48 and JRIP algorithm, we can verify that the results achieved are comparable. The first FGHSOM is similar to JRIP (petal length <= 0.152542) implies class setosa) and J48 (petal width <= 0.208333 implies class setosa) first rule.

3.2 Spanish Bankruptcy Crisis

The present section shows the capabilities of FGHSOM for the analysis and representation of financial data and for financial decision-making support. For this purpose, we analyze the Spanish banking crisis of 1977–1985 [12].

For Spanish crisis bank validation, 10 cross validation trees are grown, each uses 90% of the learn sample for tree growth and the remaining 10% as a test sample with which to estimate error rates for the nodes in the cross validation tree. The results can be observed in table 2: Two maps have been trained in order to classify bankruptcy and non-bankruptcy.

One feature of the FGHSOM is Kohonen SOM visualization. Fig. 4 shows the evolution of the map obtained in this case.

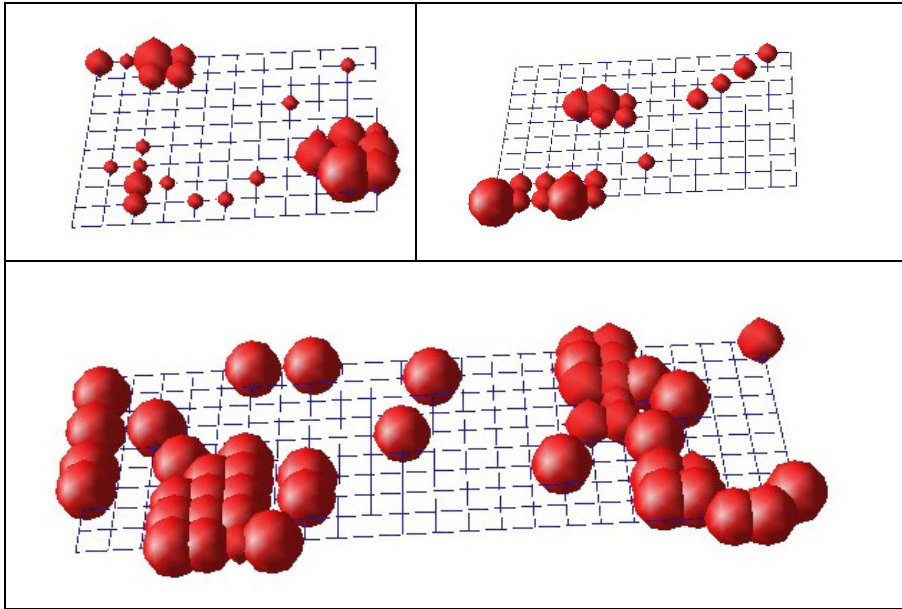


Fig. 4. Map representation for Spanish Bank crisis, the red balls represents the neuron/rules, and the ball size represents the rule weight. In this case we have tested a one layer growing map. The upper-left figure represents the initial map (9x10 neurons), the upper-right figure is an evolved map (9x12 neurons) in the training process and the lower figure is the final map representation.

Table 2. Comparison performance for FGHSOM classifier in Fisher-Anderson iris problem

Algorithm	N ^o Rules	Correctly Classified	Incorrectly Classified	% Percent Correctly Classified	% Percent Incorrectly Classified
FGHSOM	20,20	54	12	81.8182	18.1818
FGHSOM	40,40	56	10	84.8485	15.1515
FGHSOM	80, 80	57	8	86.3636	12.1212
EFUNN	35,35	48	18	72.7273	27.2727
EFUNN	61,61	53	13	80.303	19.697
JRIP	3	55	11	83.3333	16.6667
J48	3	55	11	83.3333	16.6667
MLP		57	9	86.3636	13.6364

4 Conclusions

A new algorithm based on fuzzy neural networks and Kohonen SOM called *Fuzzy Growing Hierarchical SOM* (FGHSOM) network has been developed for pattern classification. The FGHSOM classifier employs hybrid supervised and unsupervised competitive learning scheme to generate hierarchical fuzzy extraction. The network is a self-organized classifier with the capability of adaptively learning new fuzzy sets. FGHSOM allows achieving a good approximation accuracy and model transparency in a data-driven fuzzy modeling approach. FGHSOM is a new architecture which tries to reach a commitment between precision, transparency and compactness. two benchmark datasets, the Fisher's Iris dataset and the Spanish crisis Bank, were used to show the performance of the FGHSOM classifier. Comparisons between the FGHSOM classifier and some existing methods were made. The results show that in terms of classification performance, the FGHSOM classifier is competitive with or even better than many well-known classifiers, including the MLP, the EFUNN.

Acknowledgments. This work has been supported in part by the Grupo de Ingeniería Avanzada (GIA) of the Technological Institute of Aragón.

References

1. Lin, C.T., Lee, C.S.G.: Neural-network-based fuzzy logic control and decision system. *IEEE Trans. Comput.* 40, 1320–1336 (1991)
2. Jang, J.S.: ANFIS: Adaptive-network based fuzzy inference systems. *IEEE Transactions on Systems* 23(3), 665–685 (1993)
3. Berenji, H.R., Khedkar, P.: Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Trans. Neural Networks* 3, 724–740 (1992)
4. Kasabov, N.: Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning. *IEEE Trans. Syst., Man, Cybern.* 31(6), 902–918 (2001)
5. Kohonen, T.: *Self-Organizing Maps*, 3rd edn. Springer, Berlin (2001)
6. Ultsch, A., Moutarde, F.: U*F Clustering: a new performance cluster-mining method based on segmentation of SOM. In: *Proc. Workshop on Self-Organizing Maps*, Paris, pp. 25–32 (2005)
7. Hailong, L., Jue, W., Chongxun, Z.: Mental tasks classification and their EEG structures analysis by using the growing hierarchical self-organizing map. In: *2005 First International Conference on Neural Interface and Control*, Proceedings, May 26–28, pp. 115–118 (2005)
8. Pascual-Marqui, R.D., Pascual, A., Kochi, K., Carazo, J.M.: Smoothly distributed fuzzy c-means a new self-organizing map. *Pattern Recognition* 34, 2395–2402 (2001)
9. Kaburlasos, V.G., Papadakis, S.E.: Granular self-organizing map (grSOM) for structure identification. *Neural Networks* 19(5), 623–643 (2006)
10. Nomura, T., Miyoshi, M.: An Adaptive Rule Extraction with the Fuzzy Self-Organizing Map and a Comparison with Other Methods. In: *Proc. ISUMA-NAFIPS 1995*, pp. 311–316 (September 1995)
11. Bezdek, J.C., Tsao, E.C.-K.: Fuzzy Kohonen clustering networks. In: *IEEE International Conference on Fuzzy Systems*, March 8–12, pp. 1035–1043 (1992)
12. Martín, B., Serrano Cinca, C.: Self Organizing Neural Networks for the Analysis and Representation of Data: some Financial Cases. *Neural Computing & Applications* 1(2), 193–206 (1993)

Self Optimizing Neural Networks SONN-3 for Classification Tasks

Adrian Horzyk

University of Science and Technology, Department of Automatics
Mickiewicza Av. 30, 30-059 Cracow, Poland
horzyk@agh.edu.pl
<http://home.agh.edu.pl/~horzyk>

Abstract. The paper introduces new valuable improvements of a performance, a generalization ability and a topology optimization of the Self-Optimizing Neural Networks (SONNs). The described SONN-3 integrates the very effective solutions used in the SONN-2 together with the very effective ADFA algorithms for an automatic conversion of real input features into binary vectors. The integration provides not only a simple sum of valuable features of the both methods but it makes able to substantially improve a performance and generalization properties of these networks reducing SONN-3 topology sizes in comparison to SONN-2 topology sizes. The paper describes the construction of the SONN-3 and compares its performance with the SONN-2 and other AI computational methods applied to an exemplar classification task.

Keywords: binary factorization, automatic conversion, discrimination, topology and weights optimization, ontogenic neural networks.

1 Introduction

Various types of ontogenic and constructive neural networks and other incremental learning algorithms play even more important role in neural computations. These hybrid algorithms usually provide an incremental way of building neural networks with an ability to reduce topology for classification problems. Furthermore, these neural networks produce a multilayer hybrid topologies, which together with the weights, are determined automatically by the constructing algorithm and thus avoid the search for finding a proper neural network architecture. Another advantage of these algorithms is that convergence is guaranteed by the method [1],[3],[4],[8],[10]. A growing amount of current research in neural networks is oriented towards this important topic. Providing constructive methods for building hybrid neural networks can potentially create more compact models which can easily be implemented in hardware and used on various embedded systems.

This paper introduces a new generally improved and extended version of constructive hybrid ontogenic Self-Optimizing Neural Networks (SONNs) [9]. This version is not only able to fully automatically develop a network architecture and suitably compute weights but also automatically convert any integer and real data into discriminative binary vectors, optimize them and an input data dimension as well.

The previous version (SONN-2) [9] was not be able to use integer and real input training vectors. It forces the user to somehow convert training data into binary input vectors. Such manual conversion are usually not effective nor optimal and do not assure discriminative properties of the binary vectors after the conversion. Moreover, the unsuitable conversion can spoil generalization properties and results of a later constructed neural network. Furthermore, the manual conversion takes user time.

The SONN-3 have been supplemented by the automatic and very effective conversion algorithms (ADFA) which can automatically convert real input vectors to binary ones without lost of TD discrimination properties. Input TD converted by the ADFA better cover input data space and positively influence generalization properties of a neural network. Moreover, the automatic factorization of real input vectors provides smaller and better optimized SONN-3 topologies and better generalization results than for the SONN-2. The mentioned thesis is illustrated by means of an example from MLRepository (fig. 2) and performance comparisons (tab. 1).

2 The Construction and Mathematical Model of the SONN-3

The introduced construction process of the SONN-3 starts from a training data (TD) analysis. First, the SONN-3 examines all input features (inputs) of all training samples for all classes, then automatically factorizes integer and real inputs into binary bipolar vectors. Next, it counts and estimates all factorized input features and starts the process of a neural network hybrid architecture development and a weights computation (3)-(4). The SONN-3 integrates and compresses same data features of training samples as much as possible by grouping them together and by transforming features of same discrimination coefficient values into single connections. This process is associated with the process of training samples divisions into subgroups that are used to create neurons. The described algorithm creates still smaller subgroups of training samples unless all training samples are correctly discriminated.

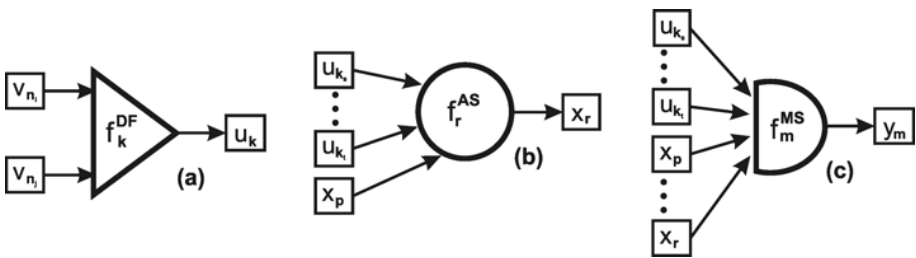


Fig. 1. Three types of SONN-3 neurons: (a) Discriminatively-Factorization Neurons (DFNs) (b) Aggregation-Strengthening Neurons (ASNs), (c) Maximum-Selection Neurons (MSNs)

The hybrid SONN-3 topologies consist of three types of neurons (fig. 1) that play a very important role in an input data transformation from raw real input vectors into a final classification achieved on network outputs. The neurons are arranged in some number of layers. The number of layers is dependent on training data correlations. If training data of different classes are more correlated then a neural network architecture

is also more complicated and it has got more layers and neurons and *vice versa*. If there are little correlations between same class training samples there are more neurons in layers and *vice versa*. The SONN-3 outputs determine the similarities of a given input vector to classes defined in a TD set. The SONN-3 qualifies some groups of inputs to be more important and pays special attention to them. On the other hand, inputs which are less or by no means important for a classification are simply reduced and not used in a classification process. The SONN-3 can automatically minimize a real data input dimension and a factorized binary data dimension (fig. 2), so this solution is free from “the curse of dimensionality problem”.

The important part of a data transformation is proceeded by Aggregation-Strengthening Neurons (ASNs) (fig. 1b) placed in the middle part of the SONN-3 architecture (fig. 2). These neurons demand bipolar binary inputs during their adaptation process. They aggregate inputs of same discrimination coefficient values together (without an information lost) and appropriately strengthen these of them that better discriminate training samples in-between various classes. The strengthening factors are called global discrimination coefficients (7). The ASNs produce their outputs (1) in the range of [-1;+1]. The ASNs are interconnected to next ASNs in such a way to promote the most discriminative features and to propagate discrimination properties of previous connections to next layers (i.e. it propagates the sum of discrimination coefficients (2) of all previous connections of a considered neuron).

$$x_r = f_r^{AS} (u_{k_s}, \dots, u_{k_t}, x_p) = w_0^{x_p} x_p + \sum_{j \in \{k_s, \dots, k_t\}} w_j^{x_r} u_j \quad (1)$$

$$d_0^{AS_p} = \sum_{j \in J} d_j \quad (2)$$

$$w_0^{AS_p} = \frac{d_0^{AS_p}}{\sum_{j \in \{k_s, \dots, k_t\}} d_j} \quad (3)$$

$$w_j^{AS_r} = \begin{cases} \frac{u_i^n \cdot d_i^+}{\sum_{j \in \{k_s, \dots, k_t\}} d_j} & \text{if } u_i^n \geq 0 \\ \frac{u_i^n \cdot d_i^-}{\sum_{j \in \{k_s, \dots, k_t\}} d_j} & \text{if } u_i^n < 0 \end{cases} \quad (4)$$

The bipolar binary inputs for the ASNs are computed by the Discriminatively-Factorization Neurons (DFNs) (fig. 1a) that can automatically factorize integer and real training data inputs (5) in such a way that existing discriminative properties of raw training data are not lost. They use ADFA algorithms (shortly described below) to automatically compute factorization ranges. The ADFA-1 uses two criteria:

1. The optimal range should contain maximum quantity of samples of the same class.
2. The optimal range containing maximum samples of the same class (there can be more than one) should be as wide as possible and be placed as far as possible from

data sample values of other classes (i.e. a maximum size of the range and maximum distances of it from values of samples of other classes are looked for).

The ADFA-1 looks for the factorization ranges $[L'_m; P'_m]$ (5) in the following steps:

1. It sorts all TD after each input feature using heapsort and mark all training samples as not discriminated.
2. All input features are looked through in order to find ranges that contain a maximum quantity of training samples of the same class to meet the criterion 1 described above. If there are two or more ranges that meet this criterion there is selected this one which meets the criterion 2. Each optimal range is described by the minimal and maximal values and the feature for which this range was found. This feature is called the winning input feature.
3. All not discriminated samples which winning input feature is in the chosen optimal range are marked as discriminated.
4. All input data features, except the winning one, are looked through in order to remove the indexes from the sorted index tables if only these indexes point out the already discriminated data samples.
5. This algorithm (steps 2-4) is repeated until all training samples are discriminated and the sorted index tables for all input data features are empty or there can not be fixed any new range of the not discriminated samples of the same class.

The discrimination ranges are transformed into DFNs which convert real inputs for the computed factorization ranges into binary bipolar outputs (5):

$$f_k^{DF}(v_{n_i}, \dots, v_{n_j}) = \begin{cases} 1 & \text{if } \bigcap_{t=n_i}^{n_j} v_t^{L'_m} \leq v_t \leq v_t^{P'_m} \\ -1 & \text{otherwise} \end{cases} \quad (5)$$

The Maximum-Selection Neurons (MSNs) (fig. 1c) produce the final output classification (6) computing the maximal values from connected ASNs that represent the aggregated most important input features of training samples of appropriate same classes. Sometimes, there can be some connections between factorized inputs and these neurons (fig. 2) for some trivial TD or very correlated training samples of a same class.

$$y_m = f_m^{MS}(u_{k_s}, \dots, u_{k_t}, x_{n_i}, \dots, x_{n_j}) = \max\{u_{k_s}, \dots, u_{k_t}, x_{n_i}, \dots, x_{n_j}\} \quad (6)$$

The above described three types of neurons are used to develop a hybrid partially connected multilayer neural network topology that can be precisely adjusted to an almost whatever TD set and a classification task. Such adjustment is similar to the plasticity [5] processes that take place in natural nervous systems [11].

The SONN-3 is an extended and generalized version of the SONN-2 and the SONN-1 widely described and discussed in many papers (e.g. [4],[8],[9]). The most important aspect of the SONN-3 is the ability to automatically factorize integer and real input data and thanks this to achieve a more optimized and better representative neural network topology and better generalization results. The SONNs require all data to be available at the beginning of the adaptation process and constant during it. The SONNs can globally estimate an importance of each input feature for all training samples and adapt a network architecture and weights reflecting it. The importance of

each k -th input feature for each n -th training sample u_n of m -th class C^m is globally measured by the use of the defined so called global discrimination coefficients (7):

$$d_{k+}^n = \begin{cases} \frac{\hat{P}_k^m}{(M-1) \cdot Q^m} \sum_{h=1 \& h \neq m}^M \left(1 - \frac{\hat{P}_k^h}{Q^h}\right) & \text{if } u_k^n \geq 0 \& u^n \in C^m \\ 0 & \text{if } u_k^n < 0 \& u^n \in C^m \end{cases} \quad (7)$$

$$d_{k-}^n = \begin{cases} \frac{\hat{N}_k^m}{(M-1) \cdot Q^m} \sum_{h=1 \& h \neq m}^M \left(1 - \frac{\hat{N}_k^h}{Q^h}\right) & \text{if } u_k^n \leq 0 \& u^n \in C^m \\ 0 & \text{if } u_k^n > 0 \& u^n \in C^m \end{cases}$$

where M denotes a number of classes in a TD set, Q is the number of training samples, L is the number of raw input features v_k^n , K is the number of factorized input features u_k^n and P_k^m , N_k^m , Q^m are defined by the following formulas:

$$\forall_{m \in \{1, \dots, M\}} Q^m = \left\| \left\{ u^n \in U \cap C^m : n \in \{1, \dots, Q\} \right\} \right\| \quad (8)$$

$$\forall_{m \in \{1, \dots, M\}, k \in \{1, \dots, K\}} P_k^m = \sum_{u_k^n \in \{u^n \in U \cap C^m : u_k^n > 0 \wedge n \in \{1, \dots, Q\}\}} u_k^n \quad (9)$$

$$\forall_{m \in \{1, \dots, M\}, k \in \{1, \dots, K\}} N_k^m = \sum_{u_k^n \in \{u^n \in U \cap C^m : u_k^n < 0 \wedge n \in \{1, \dots, Q\}\}} -u_k^n \quad (10)$$

Those globally discrimination coefficients (7) qualify the discrimination property of each k -th feature for the n -th training sample of the m -th class and have the following properties:

- they are insensitive for various quantity of training samples that represent various classes,
- a discrimination property of each feature of all training sample is precisely globally estimated,

The SONN topology optimization process is so designed to compute a minimal network structure using only these features that have maximal discrimination properties for each class and unambiguously classify all training samples [5],[8],[9]. The middle part of the SONN-3 topology is created in a specific process of TD divisions and same features aggregation. Each division produces a single neuron that represents a subgroup of training samples that meets the division criteria [9]. The SONN-3 topology optimization process is based on global discrimination coefficients (7) computed for all training data samples as well as for some subgroups of them.

SONN weights and a topology are computed simultaneously [4],[8],[9]. Such strategy makes possible to precisely assign each feature representing any subgroup of TD

to an accurate weight value (3)-(4) arising from associated discrimination properties (7) of its input features. Training samples and relations between them and classes can be very different. The SONN-3 is built up after the most important, well-differentiating and discriminating features of all TD samples. As a result, it automatically excludes artifacts of TD because they are never representative for them.

3 Experiments and Comparisons

The SONN-3 has been applied to many classification problems in order to compare this method with achieved results of a previous version of this method and with other AI methods. The figures 2 and the table 1 illustrate the architectures and compares various AI methods applied to the Wine data set from MLRepository.

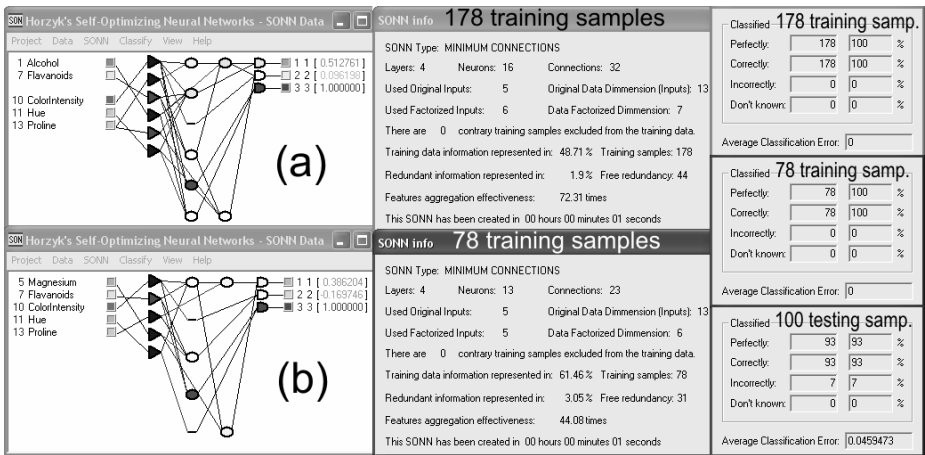


Fig. 2. The comparison of the automatically constructed SONN-3 topologies and parameters (a) for all 178 Wine samples and (b) for only 78 training and 100 testing Wine samples

The Wine data have been divided into 78 training samples and 100 testing samples in order to demonstrate the generalization comparisons for an uneasy classification task because the quantity of training samples is only 44% of all Wine samples. The data have been manually factorized for the SONN-2. The GhostMiner 3.0 with the default training parameters has been used for training and evaluation of the other AI models. The SONNs do not need to set any training parameters at all. First, all classification methods have been trained and adapted to fit 78 training samples and then all methods have been tested for 100 testing samples which were not presented during the training. The classification results comparisons are shown in the table 1. The figure 2 shows the comparisons of the SONN-3 network architectures automatically created for all and for 78 selected training Wine samples. The presented sample illustrates that:

- The original input data dimensions have been really reduced. There have been used only 5 from 13 inputs available for the Wine data.

- The factorized input dimension for the ASNs have been also reduced. There have been used only 6 from 7 available inputs for all Wine data and only 5 from 6 available inputs for the training Wine data.
- There have been selected almost the same input features in the both cases showing the stability of the algorithm that computes the discrimination properties of the features and selects the most discriminating ones. The most discriminating features are almost the same regardless of the SONN-3 has been created for all 178 or for only 78 training samples.
- The features reductions have not spoil the discrimination properties of the proceeded data and have enabled the correct classification of them.
- In spite of creation of only a few neurons and connections always all training samples and almost all test data are classified correctly.
- The whole process of the neural networks development and optimization and the weights computation have taken less than 1 second.
- The generalization properties of the created SONN-3 are at the top level in the comparison with the other AI methods (tab. 1).

Table 1. The comparison of the classification results for the various AI methods

DATA	TRAIN RESULTS		TEST RESULTS		TOTAL	
	correct	errors	correct	errors	correct	errors
Wine 78 : 100						
SONN-3	78	0	93	7	171	7
FSM	78	0	92	8	170	8
SONN-2	78	0	84	16	162	16
k-Nearest Neighbors	78	0	71	29	149	29
SSV Tree	77	1	70	30	147	31
SVM	78	0	36	64	114	64
IncNet	63	15	40	60	103	75

4 Conclusions

The paper deals with the fully automatic construction of the universal ontogenic neural network classifier SONN-3 that is able to automatically adapt itself using binary, integer and real data without any limitations. It can analyze and proceed TD very fast and find an near optimal SONN-3 architecture and weights for various TD. The SONN-3 methodology contains many optimization algorithms that minimize or maximize some data or network parameters. In each step of TD analysis and in each step of NN construction, there are used the maximally discriminating parameters coming from TD analysis. It is not easy to substantiate and proof if the constructed NN solutions are in all cases and for all TD optimal but the achieved small topologies as well as classification and generalization results show that the SONN-3 solutions are very good. This method is founded on a precise TD analysis that strictly estimates discrimination properties of all individual input features for all training samples. The results of this analysis are then grouped together, integrated and transformed into a neural network topology without any information lost. All important information coming from TD are represented strictly in a constructed NN topology and parameters. No approximations

nor rounding are used during this construction. The SONN-3 can also optimize input data dimension - automatically selecting most discriminating inputs for all TD and classes - what can have a special value in many practical uses. Not many methods can automatically select the most important inputs for given TD. On the other hand, too many inputs or a wrong selection of them (the curse of dimensionality problem ([1],[3])) can substantially spoil classification results and generalization properties of many other neural network and AI solutions. On the contrary, the SONN-3 automatically reduces original real inputs to a set of best discriminating real inputs and then reduces computed binary bipolar inputs as well. Moreover, the described SONN-3 methodology uses two kinds of information for the described optimization processes: a discriminating existence and non-existence of some input values in some classes. The majority of AI models and methods use only the information about existence of some input values in some classes and rarely the second one. Finally, the SONN-3 is very fast, cost-effective, user friendly and fully automatic tool for various classification tasks.

References

1. Duch, W., Korbicz, J., Rutkowski, L., Tadeusiewicz, R. (eds.): *Biocybernetics and Biomedical Engineering*. EXIT, Warszawa (2000)
2. Dudek-Dyduch, E., Horzyk, A.: Analytical Synthesis of Neural Networks for Selected Classes of Problems. In: Bubnicki, Z., Grzech, A. (eds.) *Knowledge Engineering and Experts Systems*, Wroclaw. OWPN, pp. 194–206 (2003)
3. Fiesler, E., Beale, R. (eds.): *Handbook of Neural Computation*. IOP Publishing Ltd./ Oxford University Press, Bristol, New York (1997)
4. Horzyk, A.: A New Extension of Self-Optimizing Neural Networks for Topology Optimization. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) *ICANN 2005*. LNCS, vol. 3696, pp. 415–420. Springer, Heidelberg (2005)
5. Horzyk, A., Tadeusiewicz, R.: Comparison of Plasticity of Self-Optimizing Neural Networks and Natural Neural Networks. In: Mira, J., Álvarez, J.R. (eds.) *IWINAC 2005*. LNCS, vol. 3561, pp. 156–165. Springer, Heidelberg (2005)
6. Horzyk, A.: Interval Basis Neural Networks. In: Ribeiro, B., Albrecht, R.F., Dobnikar, A., Pearson, D.W., Steele, N.C. (eds.) *Adaptive and Natural Computing Algorithms*, pp. 50–53. Springer, Wien (2005)
7. Horzyk, A.: Interval Basis Neural Networks as a New Classification Tool. In: *Proc. of Int. Conf. AIA*, pp. 871–876. ACTA Press, Anaheim Calgary Zurich (2005)
8. Horzyk, A., Tadeusiewicz, R.: Self-Optimizing Neural Networks. In: Yin, F.-L., Wang, J., Guo, C. (eds.) *ISNN 2004*. LNCS, vol. 3173, pp. 150–155. Springer, Heidelberg (2004)
9. Horzyk, A.: Self-Optimizing Neural Networks SONN-2. *Neurocomputing*
10. Jankowski, N.: *Ontogenic neural networks*. EXIT, Warszawa (2003)
11. Kalat, J.: *Biological Psychology*. Thomson Learning Inc., Wadsworth (2004)

Efficient MRI Reconstruction Using a Hybrid Framework for Integrating Stepwise Bayesian Restoration and Neural Network Models in a Memory Based Priors System

D.A. Karras

Chalkis Institute of Technology, Dept. Automation and Hellenic Open University,
Rodu 2, Ano Iliupolis, Athens 16342, Greece
dakarras@ieee.org, dakarras@teihal.gr, dakarras@usa.net

Abstract. The goal of this paper is to present a new image restoration methodology for extracting Magnetic Resonance Images (MRI) from reduced scans in k-space. The proposed approach considers the combined use of Support Vector Machine (SVM) models in a memory based Bayesian priors system and a novel Bayesian restoration, in the problem of MR image reconstruction from sparsely sampled k-space, following several different sampling schemes, including spiral and radial. Effective solutions to this problem are indispensable especially when dealing with MRI of dynamic phenomena since then, rapid sampling in k-space is required. The goal in such a case is to make measurement time smaller by minimizing scanning trajectories. It is suggested here that significant improvements could be achieved, concerning quality of the extracted image, applying to the k-space data a novel Bayesian restoration method based on progressively approximating the unknown image, involving a neural model based priors system with memory specifically aimed at minimizing first order image derivatives reconstruction errors. More specifically, it is demonstrated that SVM neural network techniques could construct efficient memory based Bayesian priors and introduce them in the procedure of a novel stepwise Bayesian restoration. These Priors are independent of specific image properties and probability distributions. They are based on training SVM neural filters to estimate the missing samples of complex k-space and thus, to improve k-space information capacity. Such a neural filter based Bayesian priors system with memory is integrated to the maximum likelihood procedure involved in the Bayesian reconstruction and aims at minimizing image derivatives and image reconstruction errors. It is found that the proposed methodology leads to enhanced image extraction results favorably compared to the ones obtained by the Integrated Bayesian MRI reconstruction involving simple SVM models priors as well as memory based priors minimizing image reconstruction errors instead of its derivatives errors, by the traditional Bayesian MRI reconstruction as well as by the pure Neural filter based reconstruction approach.

Keywords: MRI Reconstruction, SVM, MLP, Bayesian Restoration, Bayesian priors, Systems with memory, k-space.

1 Introduction

A data acquisition process is needed to form the MR images. Such data acquisition occurs in the spatial frequency (\mathbf{k} -space) domain, where sampling theory determines resolution and field of view, and it results in the formation of the \mathbf{k} -space matrix. Strategies for reducing image artifacts are often best developed in this domain. After obtaining such a \mathbf{k} -space matrix, image reconstruction involves fast multi-dimensional Inverse Fourier transforms, often preceded by data interpolation and re-sampling.

Sampling the \mathbf{k} -space matrix occurs along suitable trajectories [1,2,3]. Ideally, these trajectories are chosen to completely cover the \mathbf{k} -space according to the Nyquist sampling criterion. The only way to shorten scan time in MRI when needed, as for instance in functional MRI, is to reduce the overall waiting time by using fewer trajectories [1,2,3], which in turn should individually cover more of \mathbf{k} -space through added curvatures. Although, however, such trajectory omissions achieve more rapid measurements, they entail under-sampling and violations of the Nyquist criterion thus, leading to problems for image reconstruction.

The above mentioned rapid scanning in MRI problem is highly related with two other ones. The first is the selection of the optimal scanning scheme in \mathbf{k} -space that is the problem of finding the shape of sampling trajectories that more fully cover the \mathbf{k} -space using fewer trajectories. Mainly three such alternative shapes have been considered in the literature and are used in actual scanners, namely, Cartesian, radial and spiral [1], associated with different reconstruction techniques. More specifically, the Cartesian scheme uses the inverse 2D FFT, while the radial and spiral scanning involve the Projection Reconstruction, the linogram or the SRS-FT approaches [1,2,3].

The second one is associated with image estimation from fewer samples in \mathbf{k} -space that is the problem of omitting as many trajectories as possible without attaining worse reconstruction results. The main result of such scan trajectories omissions is that we have fewer samples in \mathbf{k} -space than needed for estimating all pixel intensities in image space. Therefore, there is infinity of MRI images satisfying the sparse \mathbf{k} -space data and thus, the image restoration problem becomes ill-posed. Additionally, omissions usually cause violation of the Nyquist sampling condition. Despite the fact that solutions are urgently needed, in functional MRI for instance, very few research efforts exist in the literature. The most obvious and simplest such method is the so called “zero-filling the \mathbf{k} -space”, that is, all missing points in \mathbf{k} -space acquire complex values equal to zero. Subsequently, image restoration is achieved as usually, by applying the inverse Fourier transform to the corresponding \mathbf{k} -space matrix. Instead of zero-filling the \mathbf{k} -space or using linear estimation techniques [2,3] it might be more advantageous to interpolate it by using nonlinear interpolation procedures, like Artificial Neural Networks (ANN) [6]. The Bayesian reconstruction approach, developed in [1], briefly presented in the next section is another alternative solution. Such a solution could yield good results concerning MR image restoration performance [1]. The main contribution, however, of this paper is to develop a novel MR image restoration methodology by involving both Bayesian and Neural restoration (based on SVM approximation) techniques and present its competence and advantages over other rival approaches. This line of research was first presented in [7] and herein is extended to show that a memory based ANN priors integration into the Bayesian formalism Framework offers advantages over the simple integration approach outlined in [7] for MLP ANN priors.

2 On a New Stepwise - Progressive Bayesian MRI Restoration Approach

The Bayesian restoration approach proposed in [1], attempts to provide solutions through regularizing the problem by invoking general prior knowledge in the context of Bayesian formalism. The algorithm amounts to minimizing the following objective function [1], by applying the conjugate gradients method,

$$|\underline{\mathbf{S}} - \mathbf{T} \underline{\mathbf{I}}|^2 / (2\sigma^2) + (3/2) \sum_{x,y} \log \{ \alpha^2 + ({}^x\Delta_{xy})^2 + ({}^y\Delta_{xy})^2 \} \quad (1)$$

with regards to $\underline{\mathbf{I}}$, which is the unknown image to be reconstructed that fits to the sparse k-space data given in $\underline{\mathbf{S}}$. The first term comes from the likelihood term and the second one from the prior knowledge term of the Bayesian formulation [1]. The parameter σ amounts to the variance encountered in the likelihood term and in the herein conducted simulations is set equal to 1. In the above formula, $T((k_x, k_y), (x, y)) = e^{-2\pi i(xk_x + yk_y)}$ represents the transformation from image to k-space data (through 2-D FFT). The second term symbols arise from the imposed 2D Lorentzian prior knowledge. ${}^x\Delta_{xy}$ and ${}^y\Delta_{xy}$ are the pixel intensity differences in the x- and y- directions respectively and α is a Lorentz distribution-width parameter. Assuming that $P(\mathbf{I})$ is the prior, imposing prior knowledge conditions for the unknown MRI image, then, the second term of (1) comes as follows.

The starting point is that $P(\mathbf{I})$ could be obviously expanded into $P(\mathbf{I}) = P(I_{0,0}) P(I_{1,0} | I_{0,0}) P(I_{2,0} | I_{0,0}, I_{1,0}) \dots$. If, now, it is assumed that the intensity $I_{x,y}$ depends only on its left neighbor ($I_{x-1,y}$), then the previous $P(\mathbf{I})$ expansion takes on the form $P(\mathbf{I}) = \prod_{(x,y)} P(I_{x,y} | I_{x-1,y})$, provided that the boundaries are ignored. Next, we assume that $P(I_{x,y} | I_{x-1,y})$ is a function only of the difference between the corresponding pixels. This difference is written down as ${}^x\Delta_{xy} = I_{x,y} - I_{x-1,y}$. It has been shown that the probability density function of ${}^x\Delta_{xy}$ is Lorentzian shaped (see [1,2,3]). These assumptions and calculations lead to computing the prior knowledge in the Bayesian reconstruction as in the second term of (1).

The starting point in the above optimization process is an image I_0 (usually the zero filled reconstructed image, that is, the image derived via a 2-D DFT when all unknown k-space data become equal to zero).

Although this Bayesian restoration approach tackles the problem of handling missing samples in k-space, it exhibits, however, the disadvantage that assumes the existence of special probability distributions, given in closed form descriptions, for representing the unknown ones occurred in MRI. In this paper we attempt to remedy this problem by proposing additional priors in the Bayesian formulation in order to capture the probability distribution functions encountered in MRI. These priors are constructed through applying specifically designed Support Vector Machine (SVM) neural filters for interpolating the sparsely sampled k-space.

Additionally, the above Bayesian formalism might suffer either from the fact that the zero-filled reconstructed image is not always a good starting point or from the fact that in case it is a good starting point, the conjugate gradient mechanism might follow a path far-away from such good solutions. Therefore, we propose a progressive approach to the above Bayesian formalism.

The proposed formula becomes,

$$| \underline{\mathbf{S}} - \mathbf{T} \underline{\mathbf{I}}(\mathbf{T}) - \mathbf{T} \Delta \underline{\mathbf{I}}(\mathbf{T}) |^2 / (2\sigma^2) + (3/2) \sum_{x,y} \log \{ \alpha^2 + ({}^x\Delta_{xy})^2 + ({}^y\Delta_{xy})^2 \} \quad (2)$$

where, $\underline{\mathbf{I}}(\mathbf{T})$ is the image found as a result of the application of (1) above and $\Delta \underline{\mathbf{I}}(\mathbf{T})$ is the first order error with regards to $\underline{\mathbf{I}}(\mathbf{T})$ of the application of (1). In (1)' the unknown to be found by applying the conjugate gradient method is exactly $\Delta \underline{\mathbf{I}}(\mathbf{T})$, while $\underline{\mathbf{I}}(\mathbf{T})$ is considered known. Obviously, in (1)' the image $\underline{\mathbf{I}}$ to be reconstructed is $\underline{\mathbf{I}} = \underline{\mathbf{I}}(\mathbf{T}) + \Delta \underline{\mathbf{I}}(\mathbf{T})$. Therefore, (1) and (1)' consist the two steps of an integral Bayesian formalism attempting in a progressive way to minimize errors in MRI reconstruction. The second step, namely application of (1)' above, could be considered as a conjugate gradient search for an optimum solution near a good solution found by application of (1) above. To this end, the weight of the first term of (1)', determined by parameter σ , should be increased. Therefore, during this step, the search becomes more specific by decreasing this variance σ with respect to its value in (1). The second term of (1)' remains the same as during application of (1) and aims as previously at finding smooth images.

3 Design of Efficient Memory Based Bayesian Priors Using Neural Adaptive Filters

The method herein suggested for designing efficient memory based Priors for the Bayesian reconstruction formalism, is based on the attempt to extract prior knowledge from the process of filling in the missing complex values in k-space from their neighboring complex values. Thus, instead of assuming a Lorentzian prior knowledge to be extracted from the neighboring pixel intensities in MRI, as a constraint to be applied in the conjugate gradient based Bayesian reconstruction process, the proposed strategy doesn't make any assumption. Instead, it aims at extracting priors by transforming the original reconstruction problem into an approximation one in the complex domain. While linear interpolators have already been used in the literature [2,3], ANN models could offer several advantages when applied as sparsely sampled k-space interpolators [6,7]. The methodology to extract prior knowledge by applying the ANN filters in MRI reconstruction is described in the following paragraphs.

Step1. We compile a set of R representative N X N MRI images with k-space matrices completely known, which comprise the training set of the SVM interpolators. Subsequently, we scan these matrices following the specific sampling schemes mentioned above and then, by randomly omitting trajectories the sparse k-spaces are produced, in order to simulate the real MR data acquisition process.

Step2. The original k-space matrix as well as its corresponding sparse k-space matrix associated with one N X N MRI training image, is raster scanned by a (2M+1) X (2M+1) sliding window containing the associated complex k-space values. The estimation of the complex number in the center of this window from the rest of the complex numbers comprising it is the goal of the proposed approximation procedure. For more details on this step see [6,7].

Step3. Each such pattern is then, normalized according to the following procedure. First, the absolute values of the complex numbers in the input pattern are calculated and

then, their average absolute value $|z_{\text{aver}}|$ is used to normalize all the complex numbers belonging both in the input and the desired output patterns. For more details on this step see [6,7].

Step4. The next step is the production of training patterns for the SVM interpolators and their training procedure. To this end, by randomly selecting sliding windows from the associated k-spaces of the R training images and producing the corresponding input and desired output training pairs of patterns, as previously defined, we construct the set of training patterns [6,7].

SVMs is a Neural Network (NN) methodology, introduced by Vapnik in 1992 [5]. The task of nonlinear regression using a Support Vector Machine could be defined as follows.

Let $f(\mathbf{X})$ be a multidimensional scalar valued function to be approximated, like the real/imaginary part of the sliding window central normalized complex number $z_{\text{centre}}^{\text{norm}}$ above defined in step 3. Then, a suitable regression model to be considered is: $D = f(\mathbf{X}) + n$,

where, \mathbf{X} is the input vector comprised of $(2M+1) \times (2M+1) - 1$ real/imaginary parts of the complex k-space normalized values associated with the window whose central normalized complex value is $z_{\text{centre}}^{\text{norm}}$, n is a random variable representing the noise and D denoting a random variable representing the outcome of the regression process. Given, also, the training sample set $\{(\mathbf{X}_i, D_i)\}$ ($i=1, \dots, N$) then, the SVM training can be formulated as the optimization problem next outlined:

Find the Lagrange Multipliers $\{\lambda_i\}$ ($i=1, \dots, N$) and $\{\lambda'_i\}$ ($i=1, \dots, N$) that maximize the objective function,

$$Q(\lambda_i, \lambda'_i) = \sum_{i=1..N} D_i (\lambda_i - \lambda'_i) - e \sum_{i=1..N} (\lambda_i + \lambda'_i) - 1/2 \sum_{i=1..N} \sum_{j=1..N} (\lambda_i - \lambda'_i) (\lambda_j - \lambda'_j) K(\mathbf{X}_i, \mathbf{X}_j)$$

subject to the constraints:

$\sum_{i=1..N} (\lambda_i - \lambda'_i) = 0$ and $0 \leq \lambda_i \leq C$, $0 \leq \lambda'_i \leq C$ for $i=1..N$, where C a user defined constant. In the above definition, $K(\mathbf{X}_i, \mathbf{X}_j)$ are the kernel functions. In the problem at hand we have employed the radial basis kernel

$$K(\mathbf{X}, \mathbf{X}_j) = \exp(-1/2\sigma^2 \|\mathbf{X} - \mathbf{X}_j\|^2)$$

Taking into account the previous definitions we can then, fully determine the approximating function as $F(\mathbf{X}) = \sum_{i=1..N} (\lambda_i - \lambda'_i) K(\mathbf{X}, \mathbf{X}_i)$,

which estimates the real and the imaginary part of the complex number $z_{\text{centre}}^{\text{norm}}$.

Namely, $F_{\text{real}}(\mathbf{X}_{\text{real}}) = \sum_{i=1..N} (\lambda_{i_{\text{real}}} - \lambda'_{i_{\text{real}}}) K_{\text{real}}(\mathbf{X}_{\text{real}}, \mathbf{X}_{i_{\text{real}}})$ and $F_{\text{imaginary}}(\mathbf{X}_{\text{imaginary}}) = \sum_{i=1..N} (\lambda_{i_{\text{imaginary}}} - \lambda'_{i_{\text{imaginary}}}) K_{\text{imaginary}}(\mathbf{X}_{\text{imaginary}}, \mathbf{X}_{i_{\text{imaginary}}})$ are the two corresponding SVMs. The former is applied to approximate the real part of $z_{\text{centre}}^{\text{norm}}$ while the latter for approximating its imaginary part.

Step5. After training phase completion, the SVM filter has been designed and can be applied to any similar test MRI image as follows. To this end, the $(2M+1) \times (2M+1)$ sliding window raster scans the sparse k-space matrix associated with this test image, starting from the center. Its central point position moves along the perimeter of rectangles covering completely the k-space, having as center of gravity the center of the k-space array and having distance from their two adjacent ones of 1 pixel. It can move

clockwise or counterclockwise or in both directions. For every position of the sliding window, the corresponding input pattern of $(2M+1) \times (2M+1) - 1$ complex numbers is derived following the above described normalization procedure. Subsequently, this normalized pattern feeds the SVM approximator. The wanted complex number corresponding to the sliding window center, is found as $z_{\text{centre}} = z_{\text{SVM}}^{\text{out}} * |z_{\text{aver}}|$, where $z_{\text{SVM}}^{\text{out}}$ is the SVM output and $|z_{\text{aver}}|$ the average absolute value of the complex numbers comprising the un-normalized input pattern. For each rectangle covering the k-space, the previously defined filling in process takes place so that it completely covers its perimeter, only once, in both clockwise and counterclockwise directions. The final missing complex values are estimated as the average of their clockwise and counterclockwise obtained counterparts. The outcome of the SVM filter application is the reconstructed test image, herein named SVM_Img (equation (2) below). Its difference from the image $I^{(t)}$ to be obtained during the current step of conjugate gradient optimization in the Bayesian reconstruction formula (1), provides the neural prior of time instance t to be added for the current optimization step.

In the proposed memory based framework for constructing efficient Bayesian priors based on ANN filters, however, SVM_Img(t-1), that is the reconstructed image in step t-1 by the associated SVM filter is, also, kept. Its difference from the image $I^{(t)}$ to be obtained during the current step of conjugate gradient optimization in the Bayesian reconstruction formulae (1)&(1)', provides the neural prior of time instance t-1 to be added in the set of priors for the current optimization step. Such a memory based framework in the Bayesian formalism provides the advantages of the momentum term well known in ANN and signal processing research [5] for smoothing the image to be reconstructed.

4 Integration of Neural Priors into the Bayesian Formalism in a Memory Based System Framework

Following the 5 steps above, we can formulate the incorporation of SVM priors to the Bayesian restoration process as follows.

- Design the SVM Neural filter at time instance t as previously defined
- Consider the Bayesian reconstruction formula (1)&(1)'. The image to be restored through the maximum likelihood optimization process of equations (1)&(1)' is I given the k-space S. The initial image in the process of conjugate gradient optimization is the zero-filled image. At each time step t of the process a different $I^{(t)}$ (the image at the t step, that is, the design variables of the problem) is the result. Based on figure 1 below, by applying the SVM filter on the original sparse k-space, but with the missing points initially filled by the FFT of $I^{(t)}$ (in order to derive the $I^{(t)}$ k-space)- and afterwards refined by the SVM interpolations, we could obtain the difference $I^{(t)} - \text{SVM_Img}^{(t)}$ as the Neural Prior of time instance t. Having, also, kept $\text{SVM_Img}^{(t-1)}$, the sum of differences $\Delta I^{(t)} - \Delta \text{SVM_Img}^{(t)}$ and $\Delta I^{(t)} - \Delta \text{SVM_Img}^{(t-1)}$ is considered as the memory based ANN prior at time instances t and t-1. By $\Delta F(t)$ we consider the first derivative of image F(t) or gradient of image F(t) with respect to x,y directions

- Therefore, the neural model prior form, based on SVM interpolation is:

$$\begin{aligned}
 & \text{SVM_Memory_Based_Priors_Sum=} \\
 & = \sum_{x,y} | \text{SVM_Img}^{(t)}(x,y) - I^{(t)}(x,y) | \quad + \quad (3) \\
 & + R \left(\sum_{x,y} \Delta \text{SVM_Img}^{(t)}(x,y) - \Delta I^{(t)}(x,y) \right) + \sum_{x,y} \Delta \text{SVM_Img}^{(t-1)}(x,y) - \Delta I^{(t-1)}(x,y)
 \end{aligned}$$

where, $\text{SVM_Img}^{(t)}(x,y)$ is the SVM estimated pixel intensity in image space (SVM reconstructed image: Inverse FFT of SVM completed k-space) at time step t and $I^{(t)}(x,y)$ is the image obtained at step t of the conjugate gradient optimization process in the Bayesian reconstruction. $\text{SVM_Img}^{(t-1)}(x,y)$ is the SVM estimated pixel intensity in image space at time instance $t-1$ and R is a user defined parameter between zero and one. It is actually the weight of the gradient memory based priors in the Bayesian priors sum. In our simulations $R=0.25$. In the proposed Memory based system of gradient Bayesian priors Framework we have involved only the previous time step $t-1$ prior. Of course more such gradient priors could be involved to smoothen the optimization process of Bayesian reconstruction but following the concepts of using

Integrating Memory Based SVM filters into the Bayesian prior formalism

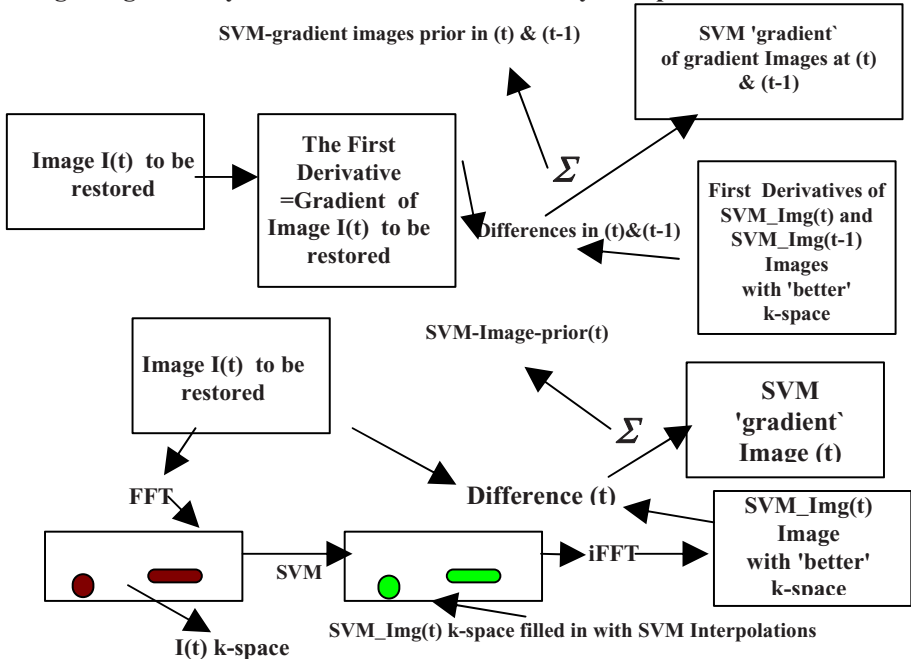


Fig. 1. The difference between the image to be restored $I(t)$ and the SVM filled in image $\text{SVM_Img}(t)$ constitutes a neural prior in time instance t . The differences between the gradient of the image to be restored $\Delta I(t)$ and the gradients of the SVM filled in images $\Delta \text{SVM_Img}(t)$ and $\Delta \text{SVM_Img}(t-1)$ constitute the memory based neural priors in time instances t and $t-1$.

the momentum term in signal processing and ANN research [5], we have adopted the one past image gradient prior approach.

- The proposed Prior in the Bayesian reconstruction is given as

$$\text{Final Prior} = \text{Lorentzian Bayesian Prior} + a * \text{SVM_Memory_Based_Priors_Sum} \quad (3)$$

• That is, the optimization process $I^{(t)}$ in (1) and (1)' is guided by the $\text{SVM_Img}^{(t)}$ as well as by the memory based system of gradients $\Delta \text{SVM_Img}^{(t)}$ of the images interpolated by the SVM at time (t) and (t-1). Therefore, the proposed algorithm attempts to find an optimum image $I(t)$ subject to the constraints of Lorentzian smoothing plus the constraints of that the reconstructed image is close to the neural filtered one as well as of that the gradient of the reconstructed image is close to the memory based gradients of the neural filtered MRI images.

5 Evaluation Study, Conclusions and Future Trends

An extensive experimental study has been conducted in order to evaluate the above defined novel Bayesian reconstruction methodology. All the methods involved have been applied to a large MRI image database downloaded from the Internet, namely, the Whole Brain Atlas <http://www.med.harvard.edu/AANLIB/home.html> (copyright © 1995-1999 Keith A. Johnson and J. Alex Becker). We have used 10 images randomly selected out of this collection for training the SVM approximation filters, and 20 images, again randomly selected for testing the proposed and the rival reconstruction methodologies. All images are 256 by 256. Their k-space matrices have been produced applying the 2D FFT to them. Radial trajectories have been used to scan the resulted 256 X 256 complex k-space arrays. 4 X 256 = 1024 radial trajectories are needed to completely cover such k-spaces. In order to apply the reconstruction techniques involved in this study, each k-space has been sparsely sampled using 128 only radial trajectories. Regarding the sliding window raster scanning the k-space, a 5 X 5 window was the best selection.

Concerning the SVM filter architecture, the 48-17-2 (number of inputs-number of support vectors-number of outputs) one was found after the SVM design stage (step 4, section 3). Actually, as explained in step 4 of section 3 this SVM filter is comprised of two different SVMs (this explains the number two of outputs). The first one is associated with approximating the real part of $z^{\text{norm}}_{\text{centre}}$ while the second one with approximating its imaginary part. This SVM approximation filter has been trained using 3600 training patterns. The compared reconstruction techniques involved in this study are: the proposed novel Bayesian mining approach, the traditional Bayesian reconstruction technique as well as the SVM filtering approximation technique. In addition, a Multilayer Perceptron (MLP) neural interpolator of 48-12-2 (number of inputs-hidden nodes-number of outputs) architecture (found to be the best one) has been involved in the comparisons. Moreover, the simplest "interpolation" approach, namely filling in the missing samples in k-space with zeroes and then, reconstructing the image, has been invoked. All these methods have been implemented using the MATLAB programming platform.

Concerning the measures involved to quantitatively compare reconstruction performance, we have employed the usually used Sum of Squared Errors (SSE) between the original MRI image pixel intensities and the corresponding pixel intensities of the

reconstructed image. Additionally, another quantitative measure has been used, which expresses performance differences in terms of the RMS error in dB [4].

The quantitative results obtained by the different reconstruction methods involved are outlined in table 1 (average SSE and RMS errors for the 20 test MRI images). Concerning reconstruction performance qualitative results, a sample is shown in figure 2. Both quantitative and qualitative results clearly demonstrate the superiority of the proposed Bayesian image mining methodology embedding SVM filtering based prior knowledge, in terms of MRI image restoration performance over the other rival methodologies (simple Bayesian restoration, SVM / MLP MRI filter and zero-filled

Table 1. The quantitative results with regards to reconstruction performance of the various methodologies

MRI Restoration Method	SSE (average in the 20 test MRI images)	dB (average in the 20 test MRI images)
Proposed Memory Based Framework of stepwise Bayesian MR Image Restoration with SVM Priors, including gradient priors	2.22 E3	19.78
Memory based Framework of traditional Bayesian MR Image Restoration with SVM Priors, and without gradient priors	2.31 E3	18.91
Bayesian MR Image Restoration with SVM Prior in a non-memory based priors framework	2.62 E3	17.51
Bayesian MR Image Restoration with MLP Prior in a non-memory based priors framework [7]	2.84 E3	16.68
Traditional Bayesian restoration [1]	3.40 E3	15.93
SVM restoration	3.27 E3	16.02
MLP restoration [6]	3.30 E3	15.98
Zero-filling restoration	3.71 E3	15.27

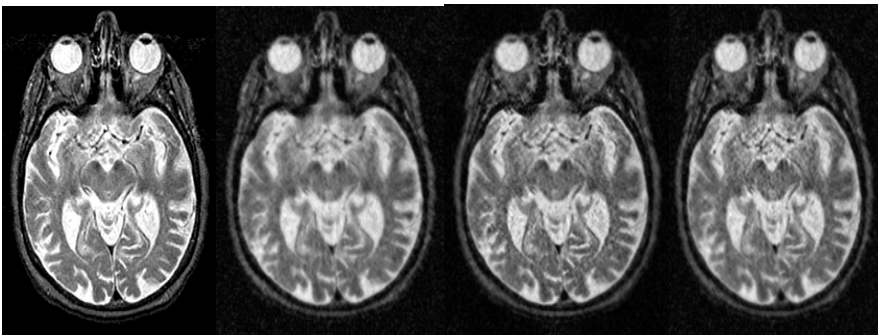


Fig. 2. Left to right: The proposed Bayesian MR Image Restoration with Memory Based SVM priors, the sparsely sampled k-space (nr=128)-zerofilled image reconstruction, the Bayesian MR Image Restoration with MLP priors [7] and the traditional Bayesian

reconstructions). Future trends of our research efforts include implementation of the 3-D Bayesian reconstruction with Neural Network priors for f-MRI as well as applications in MRI image segmentation for tumor detection.

References

1. Stijnman, G.H.L.A., Graveron-Demilly, D., Wajer, F.T.A.W., van Ormondt, D.: MR Image Estimation from Sparsely Sampled Radial Scans. In: Proc. ProRISC/IEEE Benelux workshop on CSSP, Mierlo, The Netherlands, pp. 603–611 (1997)
2. Smith, M.R., et al.: Application of Autoregressive Modelling in Magnetic Resonance Imaging to Remove Noise and Truncation Artifacts. *MRI journal* 4, 257 (1986)
3. Barone, P., Sebastiani, G.: A New Method of Magnetic Resonance Image Reconstruction with Short Acquisition Time and Truncation Artifact Reduction. *IEEE Trans. Med. Imaging* 11, 250 (1992)
4. Dologlou, I., et al.: Spiral MRI Scan-Time Reduction through Non-Uniform Angular Distribution of Interleaves and Multichannel SVD Interpolation. In: Proc. ISMRM, 4th Meeting, N.Y., p. 1489 (1996)
5. Haykin, S.: *Neural Networks: A comprehensive foundation*, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)
6. Reczko, M., Karras, D.A., Mertzios, B.G., Graveron-Demilly, D., van Ormondt, D.: Improved MR image reconstruction from sparsely sampled scans based on neural networks. *Journal pattern recognition letters* 22(1), 35–46 (2001)
7. Reczko, M., Karras, D.A., et al.: Improved MRI Reconstruction From Reduced Scans K-Space By Integrating Neural Priors In The Bayesian Restoration. In: 23rd EMBC 2001, Turkey (October 2001)

Traffic Data Preparation for a Hybrid Network IDS

Álvaro Herrero and Emilio Corchado

Department of Civil Engineering, University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
{ahcosio, escorchado}@ubu.es

Abstract. An increasing effort has been devoted to researching on the field of Intrusion Detection Systems (IDS's). A wide variety of artificial intelligence techniques and paradigms have been applied to this challenging task in order to identify anomalous situations taking place within a computer network. Among these techniques is the neural network approach whose models (or most of them) have some difficulties in processing traffic data "on the fly". The present work addresses this weakness, emphasizing the importance of an appropriate segmentation of raw traffic data for a successful network intrusion detection relying on unsupervised neural models. In this paper, the presented neural model is embedded in a hybrid artificial intelligence IDS which integrates the case based reasoning and multiagent paradigms.

Keywords: Computer Network Security, Network Intrusion Detection, Artificial Neural Networks, Unsupervised Learning, Projection Methods, Artificial Intelligence, Hybrid Artificial Intelligence Systems.

1 Introduction

Intrusion Detection Systems (IDS's) [1] are becoming more and more popular as network security tools, although firewalls are still the most widely used tool of this kind. A network IDS (NIDS) monitor the activity of a network with the purpose of identifying intrusive events and can take actions to abort these risky events.

Up to now, a wide range of Artificial Intelligence (AI) techniques have been used to build IDS's. On the one hand, there have been some previous attempts to take advantage of agents and Multiagent Systems (MAS) [2] in the field of Intrusion Detection (ID) [3], [4], [5]. It is worth mentioning the mobile-agents approach [6], [7]. On the other hand, some different machine learning models – including Data Mining techniques and Artificial Neural Networks (ANN) – have been successfully applied for ID [8], [9], [10], [11], [12]. As can be seen, there have been a huge number of attempts to apply ANN to the detection of intrusions.

Additionally, some other AI techniques have been combined (genetic algorithms and fuzzy logic [13], genetic algorithms and K-Nearest Neighbor (K-NN) [14] or K-NN and ANN [15] among others) in order to face ID from a hybrid point of view. In some cases they provide intelligence to MAS. This paper employs the Mobile Visualization Connectionist Agent-Based IDS (MOVICAB-IDS) [2], a dynamic multiagent architecture for network intrusion detection that includes deliberative

agents capable of learning and evolving with the environment. Some previous work has been devoted to the managing and processing of network traffic data [16], [17], [18], [19].

The remaining four sections of this paper are structured as follows: traffic data management is outlined in section 2, section 3 contains an overview of MOVICAB-IDS; some experimental results are described in section 4; finally, section 5 shows a comparative study and puts forward a number of conclusions.

2 Traffic Data Preparation

As in the case of traffic management [16], we can divide ID into three well defined tasks:

1. Data collecting: prior to getting the data from the network, some issues must be addressed; selecting the sources from where the information is going to be extracted, specifying the parameters (time length, filters, etc.) of the data gathering, defining the final format of the collected data, etc.
2. Data processing: once the data is gathered, it is processed for the identification of intrusions and attacks.
3. Intrusion abortion: some IDS's (lately referred as Intrusion Prevention Systems) incorporate any mechanism for stopping intrusions at the very moment they are identified.

This paper focuses on the Data collecting task, taking into account the connectionist techniques applied in the processing task. For the data collecting task, a 4 stage framework is proposed. These four stages are described in the following sections.

Stage 1. Data Capture

As this framework is proposed for a NIDS, a continual data flow must be managed. This data flow contains information about all the packets travelling along the network to be monitored. For this information to be captured, one of the network interfaces of a host within this network must be set up in promiscuous mode. Thus, this interface is able to "see" all the packets travelling along the network, whatever its destination host is.

Stage 2. Data Selection

NIDS's have to deal with the practical problem of high volumes of quite diverse data [17]. To deal with the problem of high diversity, we propose the splitting of the traffic into different groups, taking into account the protocol (either UDP, TCP, ICMP...) over IP as TCP and UDP packets are quite different. Once the captured data is classified by the protocol over IP, it can be processed in quite different ways. There have been several approaches dealing with traffic data summarized by TCP connections, such as the well-known KDD dataset [20]. On the contrary, we propose a packet-based approach, where each instance in the final datasets corresponds to a single packet. This packet-based approach has proven to be successful in the identification of some anomalies [2], [12].

Stage 3. Segmentation

The two first stages do not deal with the problem of continuity in network traffic data. As it is said before, most neural models can not process traffic data “on the fly”. Therefore, in order to overcome this shortcoming, we propose a way of creating limited datasets from this continuous data flow by segmenting it. Two kinds of segments are proposed:

- **Simple segments.** Each simple segment contains all the packets whose timestamps are between the initial and final time limits of the segment. There must be a time overlap between each pair of consecutive simple segments because anomalous situations could conceivably take place between simple segment S_x and S_{x+1} (where S_{x+1} is the next segment following S_x). In this case, it would be necessary to consider some packets twice in order to visualize the end of the anomalous situation and the evolution between simple segments.
- **Accumulated segments.** Each one of these segments contains several consecutive simple ones (removing the time overlap). The main considerations are, firstly, to present a long-term picture of the evolution of network traffic and, secondly, to allow the identification of attacks lasting longer than the length of a simple segment.

There are some key issues concerning segmentation that are quite important, such as the length (time duration) of the simple segments, the overlap time and the number of simple segments making up the accumulated segments among others.

Stage 4. Data Preprocessing

Finally, the different datasets (simple and accumulated segments) must be preprocessed before presenting them to the neural model. In this case of network traffic data, it is not needed the application of some techniques such as denoification, outlier detection, missing data managing and some others. Depending on the neural model to be applied, only data normalization is needed.

3 MOVICAB-IDS

For the experimental verification (see section 4) of the proposed data segmentation framework, MOVICAB-IDS, is proposed. It may be roughly defined as a hybrid NIDS formed of different software agents [21] that work in unison in order to detect anomalous situations by taking full advantage of an unsupervised connectionist model [22], [23]. The traffic data preparation is performed by MOVICAB-IDS in the following steps:

- 1st step.- Traffic Data Capture: packets travelling over the different network segments are captured.
- 2nd step.- Data Selection: the captured data is selected. A set of packets and features contained in the headers of the captured data is extracted from the raw network traffic.
- 3rd step.- Segmentation: the data stream is divided into simple and accumulated segments.

Once the data is ready, the detection of intrusions goes through:

- 4th step.- Data Analysis: after preprocessing the data, the Cooperative Maximum Likelihood Hebbian Learning (CMLHL) model [22], [23] is applied to analyse it (see section 3.1 for further details).
- 5th step.- Visualization: the projections of traffic data are presented to the network administrator for the supervision and monitoring (see section 5 for samples).

MOVICAB-IDS makes use of the hybrid approach to perform all these steps, combining the following paradigms:

- Multiagent systems (MAS) [2]: this NIDS employs deliberative agents capable of learning and evolving with the environment.
- Case-Based Reasoning (CBR) [24]: some of the agents contained in the previously described MAS are known as CBR-BDI agents [25] because they integrate the BDI (Believes, Desires and Intentions) [26] model and the CBR paradigm.
- Artificial Neural Networks: the connectionist approach fits the intrusion-detection problem mainly because it allows a system to learn empirically the input-output relationship between traffic data and its subsequent interpretation. The previously described CBR-BDI agents incorporate the CMLHL neural model [22], [23].

The combination of these paradigms let us take advantage of some of the properties of ANN (generalization that allows the identification of previously unseen attacks), CBR (learning from past experiences) and agents (reactivity, proactivity and sociability), making the ID task possible.

3.1 The Neural Model

The Data Analysis step previously mentioned is based on the use of the unsupervised neural model called Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [22], [23]. It is based on Maximum Likelihood Hebbian Learning (MLHL) [22], [27]. Consider an N-dimensional input vector, \mathbf{x} , and an M-dimensional output vector, \mathbf{y} , with W_{ij} being the weight linking input j to output i and let η be the learning rate. MLHL can be expressed as:

$$y_i = \sum_{j=1}^N W_{ij} x_j, \forall i. \quad (1)$$

The activation (e_j) is fed back through the same weights and subtracted from the input:

$$e_j = x_j - \sum_{i=1}^M W_{ij} y_i, \forall j. \quad (2)$$

Weight change:

$$\Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j) |e_j|^{p-1}. \quad (3)$$

Where: η is the learning rate and p is a parameter related to the energy function.

Lateral connections [28], [23] have been derived from the Rectified Gaussian Distribution [29] and applied to the MLHL model. The resultant net (CMLHL) can find the independent factors of a data set but do so in a way that captures some type of global ordering in the data set. So, the final CMLHL model is as follows:

Feed forward step: Equation (1)

$$\text{Lateral activation passing: } y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ . \quad (4)$$

Feed back step: Equation (2)

Weight change: Equation (3)

Where: τ is the “strength” of the lateral connections, b is the bias parameter and A is a symmetric matrix used to modify the response to the data. Its effect is based on the relation between the distances among the output neurons.

4 Experimental Study

Among all the implemented network protocols, there are several of them that can be considered potentially dangerous, such as the Simple Network Management Protocol (SNMP) [30], ICMP (Internet Control Message Protocol), TFTP (Trivial File Transfer Protocol) and so on. SNMP was identified as one of the top five most vulnerable services by CISCO [31], specially the two first versions of this protocol that are the most widely used at present time. An attack based on this protocol may severely compromise the security of the whole network [32]. Thus, this research line is focused on the identification of anomalous situations concerning SNMP.

Most of the security tools focus their attention on attacks coming from the internet but attacks are just as likely to come from inside the network as from the outside, however. MOVICAB-IDS helps network administrators to identify one of the most dangerous set of inside attacks: those related to the SNMP.

SNMP was oriented to manage nodes in the Internet community [30]. That is, it is used to control routers, bridges, and other network elements, reading and writing a wide variety of information about the devices, such as operating system, version, routing tables, default TTL (Time To Live) and so on. The Management Information Base (MIB) can be defined in broad terms as the database used by SNMP to store information about the elements that it controls.

Three main anomalous situations related to the SNMP are analyzed in this section, namely: network/port scans, SNMP community search and MIB information transfers. These SNMP-related anomalous situations are described in this section.

Network Scan

A network scan may be seen as series of messages sent to the same port number of different host to gain information on protocols/services activity status. These messages can be sent by an external agent attempting to access a host to find out more about the network services running. A network scan provides information on where to

probe for weaknesses, for which reason scanning generally precedes any further intrusive activity [33].

SNMP Community Search

The community string can be seen as the SNMP password for versions 1 and 2. This anomalous situation is characterized by the intruder sending SNMP queries to determine the SNMP community string. Once the community string has been obtained, all the sensitive information stored in the MIB is available for the intruder.

MIB Information Transfer

This situation is a transfer of some (or all the) information contained in the SNMP MIB. This kind of transfer is potentially quite a dangerous situation, but, on the other hand, the “normal” behaviour in the network includes queries to the MIB.

The segments analyzed in this section contain examples of all these anomalous situations. As SNMP is based on UDP, this section only deals with UDP traffic. Apart from the anomalous traffic previously described, information concerning normal traffic from a middle-size network is included as well. The MOVICAB-IDS visualization of the following four segments is depicted in Fig. 1:

- S_1 : this simple segment does only contain normal data. The segment length is 600 seconds.
- S_4 : this simple segment contains an MIB information transfer as well as normal data. The segment length is 600 seconds.
- A_3 : this accumulated segment contains several network scans and SNMP community searches as well as normal data. The segment length is 1560 seconds.
- A_{13} : this accumulated segment contains several network scans, SNMP community searches and two MIB information transfers as well as normal data. The segment length is 6360 seconds.

All the packets contained in the three first segments (S_1 , S_3 , A_3) are also contained in A_{13} .

As can be seen in Fig. 1.a, normal traffic is depicted by MOVICAB-IDS as parallel straight lines. That is, all the packets (associated by protocol) evolve in the same direction (labelled in Fig. 1.a as “Normal” direction). In the case of an MIB information transfer, it can be identified in Fig. 1.b (labelled in Fig. 1.b as Groups 1 and 2) due to the high concentration of packets and the evolution of these packets in directions non-parallel to the one for normal traffic. It is easy to identify anomalous situations within simple segments.

The other two anomalous situations, namely network scans and SNMP community searches are identified as anomalous in Fig. 1.c (Groups 1, 2 and 3). The non-parallel evolution of these groups enables us to label them as anomalous. It can be checked once again in Fig. 1.d where examples of all the three anomalous situations understudy can be identified as anomalous due to their non-parallel evolution and

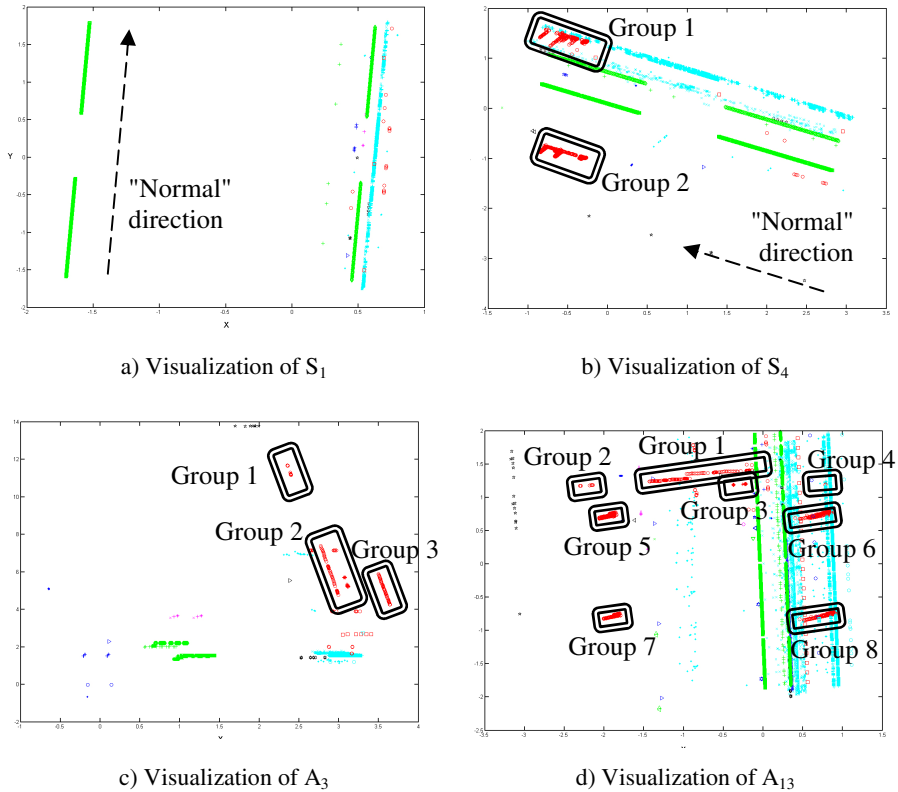


Fig. 1. Visualizations obtained by MOVICAB-IDS

high temporal concentration of packets: network scans (Group 1 in Fig. 1.d), SNMPcommunity search (Group 2, 3 and 4 in Fig. 1.d) and the MIB information transfers (Groups 5, 6, 7 and 8 in Fig. 1.d). In this case, as the analysed dataset is much bigger, the identification of some anomalous situations (specially the cases of the community search) is not as easy as in the other smaller segments.

5 Comparative Study and Conclusions

For comparison purposes, two well-known projection neural models, namely Principal Component Analysis (PCA) and Curvilinear Component Analysis (CCA) were applied to A_3 segment (Fig. 2). It also validates the proposed framework for traffic data preparation. As can be seen in Fig 2.a, PCA failed to detect the anomalous situations (network scans and SNMP community search) contained in A_3 . Both the network scans (Groups 1 and 2 in Fig. 2.a) and the SNMP community searches (Group 3 in Fig. 2.a) contained in A_3 are not identified as anomalous traffic as these packets evolve in parallel lines.

Fig. 2.b shows the projection of A_3 obtained by CCA using Euclidean distance. The anomalies are differentiated from normal traffic on the basis of parallel evolution as in the case of CMLHL. The network scans (Groups 1 and 2 in Fig. 2.b) together with the SNMP community searches (Groups 3 and 4 in Fig. 2.b) are depicted as lines non parallel to normal traffic. The main difference between the CMLHL projection and the one obtained by CCA is that CMLHL minimizes the overlapping between the different groups, getting clearer and more sparse projections.

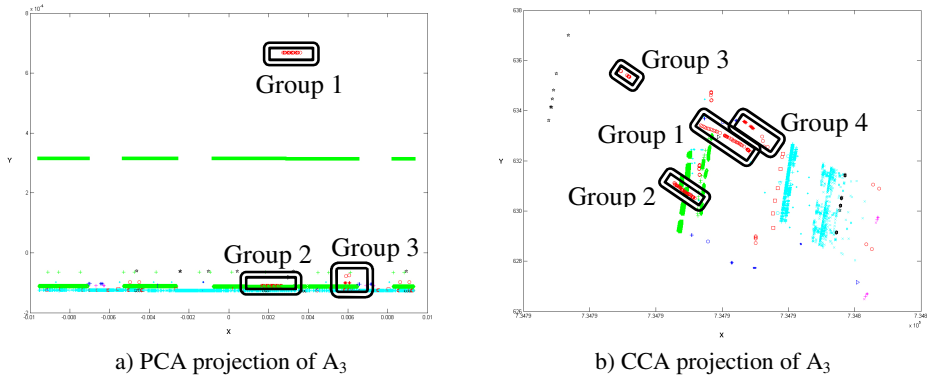


Fig. 2. Comparison of A_3 projections

We can conclude that a proper traffic data preparation enables a successful network intrusion detection relying on neural models. However, it does not ensure a successful identification of attacks as the applied neural model is decisive as well. On the other hand, there are some issues concerning segmentation whose importance is worth mentioning. That is the case of the length (time duration) of the simple segments, as the longer the segments are, the more likely an attack will be unnoticed.

Acknowledgments. This research has been partially supported by the project BU006A08 of the Junta de Castilla y León.

References

1. Anderson, J.P.: Computer Security Threat Monitoring and Surveillance. Technical Report, Washington, PA, James P. Anderson Co. (1980)
2. Herrero, Á., Corchado, E., Pellicer, M., Abraham, A.: Hybrid Multi Agent-Neural Network Intrusion Detection with Mobile Visualization. In: Innovations in Hybrid Intelligent Systems. Advances in Soft Computing, vol. 44, pp. 320–328. Springer, Heidelberg (2007)
3. Spafford, E.H., Zamboni, D.: Intrusion Detection Using Autonomous Agents. Computer Networks: The Int. Journal of Computer and Telecommunications Networking 34(4), 547–570 (2000)
4. Hegazy, I.M., Al-Arif, T., Fayed, Z.T., Faheem, H.M.: A Multi-agent Based System for Intrusion Detection. IEEE Potentials 22(4), 28–31 (2003)

5. Dasgupta, D., Gonzalez, F., Yallapu, K., Gomez, J., Yarramsetti, R.: CIDS: An agent-based intrusion detection system. *Computers & Security* 24(5), 387–398 (2005)
6. Wang, H.Q., Wang, Z.Q., Zhao, Q., Wang, G.F., Zheng, R.J., Liu, D.X.: Mobile Agents for Network Intrusion Resistance. In: Shen, H.T., Li, J., Li, M., Ni, J., Wang, W. (eds.) *APWeb Workshops 2006*. LNCS, vol. 3842, pp. 965–970. Springer, Heidelberg (2006)
7. Deeter, K., Singh, K., Wilson, S., Filipozzi, L., Vuong, S.: APHIDS: A Mobile Agent-Based Programmable Hybrid Intrusion Detection System. In: Karmouch, A., Korba, L., Madeira, E.R.M. (eds.) *MATA 2004*. LNCS, vol. 3284, pp. 244–253. Springer, Heidelberg (2004)
8. Laskov, P., Dussel, P., Schafer, C., Rieck, K.: Learning Intrusion Detection: Supervised or Unsupervised? In: Roli, F., Vitulano, S. (eds.) *ICIAP 2005*. LNCS, vol. 3617, pp. 50–57. Springer, Heidelberg (2005)
9. Liao, Y.H., Vemuri, V.R.: Use of K-Nearest Neighbor Classifier for Intrusion Detection. *Computers & Security* 21(5), 439–448 (2002)
10. Sarasamma, S.T., Zhu, Q.M.A., Huff, J.: Hierarchical Kohonen Net for Anomaly Detection in Network Security. *IEEE Transactions on Systems Man and Cybernetics, Part B* 35(2), 302–312 (2005)
11. Zanero, S., Savaresi, S.: Unsupervised Learning Techniques for an Intrusion Detection System. In: *Proc. of the ACM Symposium on Applied Computing*, pp. 412–419 (2004)
12. Corchado, E., Herrero, A., Sáiz, J.M.: Detecting Compounded Anomalous SNMP Situations Using Cooperative Unsupervised Pattern Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) *ICANN 2005*. LNCS, vol. 3697, pp. 905–910. Springer, Heidelberg (2005)
13. Sindhu, S.S.S., Ramasubramanian, P., Kannan, A.: Intelligent Multi-agent Based Genetic Fuzzy Ensemble Network Intrusion Detection. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) *ICONIP 2004*. LNCS, vol. 3316, pp. 983–988. Springer, Heidelberg (2004)
14. Middlemiss, M., Dick, G.: Feature Selection of Intrusion Detection Data Using a Hybrid Genetic Algorithm/KNN Approach. In: *Design and application of hybrid intelligent systems*, pp. 519–527. IOS Press, Amsterdam (2003)
15. Kholfi, S., Habib, M., Aljahdali, S.: Best Hybrid Classifiers for Intrusion Detection. *Journal of Computational Methods in Science and Engineering* 6(2), 299–307 (2006)
16. Babu, S., Subramanian, L., Widom, J.: A Data Stream Management System for Network Traffic Management. In: *Proc. of Workshop on Network-Related Data Management (NRDM 2001)* (2001)
17. Dreger, H., Feldmann, A., Paxson, V., Sommer, R.: Operational Experiences with High-volume Network Intrusion Detection. In: *Proc. of the 11th ACM Conf. on Computer and Communications Security*, pp. 2–11. ACM Press, New York (2004)
18. Hall, M., Wiley, K.: Capacity Verification for High Speed Network Intrusion Detection Systems. In: Wespi, A., Vigna, G., Deri, L. (eds.) *RAID 2002*. LNCS, vol. 2516, pp. 239–251. Springer, Heidelberg (2002)
19. Lee, W., Stolfo, S.J., Mok, K.W.: Mining in a Data-Flow Environment: Experience in Network Intrusion Detection. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, San Diego (1999)
20. Elkan, M.: Results of the KDD 1999 Classifier Learning Contest (1999), <http://www-cse.ucsd.edu/users/elkan/clresults.html>
21. Wooldridge, M., Jennings, N.R.: Agent theories, architectures, and languages: A survey. *Intelligent Agents* (1995)

22. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. *Data Mining and Knowledge Discovery* 8(3), 203–225 (2004)
23. Corchado, E., Fyfe, C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. *Int. Journal of Pattern Recognition and Artificial Intelligence* 17(8), 1447–1466 (2003)
24. Aamodt, A., Plaza, E.: Case-Based Reasoning - Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7(1), 39–59 (1994)
25. Carrascosa, C., Bajo, J., Julián, V., Corchado, J.M., Botti, V.: Hybrid Multi-agent Architecture as a Real-Time Problem-Solving Model. *Expert Systems with Applications: An International Journal* 34(1), 2–17 (2008)
26. Bratman, M.E.: *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge (1987)
27. Fyfe, C., Corchado, E.: Maximum Likelihood Hebbian Rules. In: *Proc. of the 10th European Symposium on Artificial Neural Networks (ESANN 2002)*, pp. 143–148 (2002)
28. Corchado, E., Han, Y., Fyfe, C.: Structuring Global Responses of Local Filters Using Lateral Connections. *Journal of Experimental & Theoretical Artificial Intelligence* 15(4), 473–487 (2003)
29. Seung, H.S., Socoli, N.D., Lee, D.: The Rectified Gaussian Distribution. *Advances in Neural Information Processing Systems* 10, 350–356 (1998)
30. Case, J., Fedor, M.S., Schoffstall, M.L., Davin, C.: Simple Network Management Protocol (SNMP). RFC-1157 (1990)
31. Cisco Secure Consulting. *Vulnerability Statistics Report* (2000)
32. Myerson, J.M.: Identifying Enterprise Network Vulnerabilities. *Int. Journal of Network Management* 12(3), 135–144 (2002)
33. Stephen, L.: The Spinning Cube of Potential Doom. *Commun. ACM* 47(6), 25–26 (2004)

Comparing Hybrid Versions of SS and DE to Solve a Realistic FAP Problem

José M. Chaves-González¹, Marisa da Silva Maximiano²,
Miguel A. Vega-Rodríguez¹, Juan A. Gómez-Pulido¹, and Juan M. Sánchez-Pérez¹

¹ Univ. Extremadura. Dept. Technologies of Computers and Communications,
Escuela Politécnica. Campus Universitario s/n. 10071. Cáceres, Spain
{jm, mavega, jangomez, sanperez}@unex.es

² Polytechnic Institute of Leiria. School of Technology and Management. Leiria, Portugal
marisa.maximiano@estg.ipleiria.pt

Abstract. In this work we study and compare hybrid versions of two different metaheuristics to solve a real-world frequency assignment problem (FAP) in GSM networks. We have used a precise mathematical formulation, developed in published work, in which the frequency plans are evaluated using accurate interference information coming from a real GSM network. Therefore, we focus on solving FAP problem for a realistic-sized, real-world GSM network using a hybrid version of Scatter Search and Differential Evolution algorithms. We have analyzed and fixed both approaches to the FAP problem and, after a detailed statistical study, the obtained results prove that our hybrid approaches compute accurate frequency plans for real-world instances in an optimum way. In fact, our results surpass all the results previously published in the literature.

Keywords: FAP, Frequency Planning, SS, DE, real-world GSM network.

1 Introduction

In this paper we study and compare two different hybrid metaheuristics which solve in an optimum way a realistic-sized real-world frequency assignment problem (FAP). FAP is an NP-hard problem, so its resolution using metaheuristics [1] has proved to be particularly effective. We have evaluated hybrid versions of the algorithms DE (Differential Evolution) and SS (Scatter Search) on a real-world GSM (Global System for Mobile communications) network with 2612 transceivers. Moreover both used metaheuristics have proved to be very effective in the resolution of optimization problems (such as the FAP is) and consequently its usage has been significantly increased in the last few years [2, 3]. It is important to point that GSM is the most successful mobile communication technology nowadays. In fact, by mid 2006 GSM services are in use by more than 1.8 billion subscribers [4] across 210 countries, representing approximately 77% of the world's cellular market. Besides, FAP is one of the most relevant and significant problems that it can be found in the GSM technology. To deal with the realistic problem we work with, we use a complex formulation, proposed in [5], which takes in consideration the requirements of real-world GSM. The two most significant elements we find in the frequency planning problem are the transceivers

(TRXs) which give support to the communication and the frequencies which make possible the communication. A mobile communication antenna includes several TRXs placed in several sectors. The problem is that there are not enough frequencies (there are not more than a few dozens) to give support to each transceiver (there are usually thousands of them) without causing interferences. It is completely necessary to repeat frequencies in different TRXs, so, a good planning to minimize the number of interferences is highly required.

The rest of the paper is structured as follows: In section 2 we present the fundamentals of the FAP solved and the mathematical formulation we use for its resolution. Section 3 describes the hybridization performed in the metaheuristics used. The algorithms compared in this study are explained in sections 4 (SS) and 5 (DE). In section 6 we explain the experiments and the results obtained. Finally, the conclusions and future work of the research are discussed in the last section.

2 Frequency Planning Problem in GSM Networks

The two most relevant components which refer to frequency planning in GSM systems are the *antennas* or, as they are more known, base transceiver stations (BTSs) and the *transceivers* (or TRXs). The TRXs of a network are installed in the BTSs where they are grouped in sectors, oriented to different points to cover different areas. The instance we use is quite large (it covers the city of Denver, with more than 500,000 inhabitants) and the GSM network includes 2612 TRXs, grouped in 711 sectors, distributed in 334 BTSs. We are not going to extend the explanation of the GSM network architecture but the reader interested on it can consult reference [6].

FAP lies in the assignment of a channel (or a frequency) to every TRX [6] in the network. This is one of the latest stages in a GSM network design. The optimization problem arises because the usable radio spectrum is very scarce and frequencies have to be reused for many TRXs in the network. However, the multiple use of a same frequency may cause interferences that can reduce the quality of service (QoS) down to unsatisfactory levels. Although there are several ways of quantifying the interferences produced in a telecommunication network, the most extended one is by using what is called the *interference matrix* [7], denoted by M . Each element $M(i,j)$ of this matrix contains two types of interferences: the *co-channel interference*, which represents the degradation of the network quality if the cells i and j operate on the same frequency; and the *adjacent-channel interference*, which occurs when two TRXs operate on adjacent channels (e.g., one TRX operates on channel f and the other on channel $f+1$ or $f-1$). An accurate interference matrix is an essential requirement for frequency planning because the final goal of any frequency assignment algorithm will be to minimize the sum of all the interferences. In addition to the requirements described above, frequency planning includes more complicating factors which occur in real life situations (see [6] for a detailed explanation).

Finally, in the following subsection we give a brief description of the mathematical model we use. This model was proposed in a previous work, so the reader who wants a deeper explanation can consult the reference [5].

2.1 Mathematical Formulation

Let $T = \{t_1, t_2, \dots, t_n\}$ be a set of n transceivers (TRXs), and let $F_i = \{f_{i1}, \dots, f_{ik}\} \subset N$ be the set of valid frequencies that can be assigned to a transceiver $t_i \in T, i = 1, \dots, n$. Note that k , which is the cardinality of F_i , is not necessarily the same for all the transceivers. Furthermore, let $S = \{s_1, s_2, \dots, s_m\}$ be a set of given sectors (or cells) of cardinality m . Each transceiver $t_i \in T$ is installed in exactly one of the m sectors. Moreover, we denote the sector in which a transceiver t_i is installed by $s(t_i) \in S$. Finally, the interference matrix, M , is defined as: $M = \{(\mu_{ij}, \sigma_{ij})\}_{m \times m}$, where the two elements μ_{ij} and σ_{ij} of a matrix entry $M(i, j) = (\mu_{ij}, \sigma_{ij})$ are numerical values greater or equal than zero, and they represent the mean (μ_{ij}) and the standard deviation (σ_{ij}) of a Gaussian probability distribution describing the carrier-to-interference ratio (C/I) when sectors i and j operate on a same frequency. The higher the mean value is, the lower the interference will be, and thus the better the communication quality.

A solution to the problem is obtained by assigning to each TRX $t_i \in T$ one of the frequencies from F_i . We will denote a solution (or frequency plan) by $p \in F_1 \times F_2 \times \dots \times F_n$, where $p(t_i) \in F_i$ is the frequency assigned to the TRX t_i . Therefore, the plan solution will be to find a solution p that minimizes the following cost function:

$$C(p) = \sum_{t \in T} \sum_{u \in T, u \neq t} C_{sig}(p, t, u) \tag{1}$$

In order to define the function $C_{sig}(p, t, u)$, let s_t and s_u be the sectors in which the transceivers t and u are installed, which are $s_t = s(t)$ and $s_u = s(u)$ respectively. Moreover, let μ_{s_t, s_u} and σ_{s_t, s_u} be the two elements of the corresponding matrix entry $M(s_t, s_u)$ of the interference matrix with respect to sectors s_t and s_u . Then, $C_{sig}(p, t, u) =$

$$\begin{cases} K & \text{if } s_t = s_u, |p(t) - p(u)| < 2 \\ C_{co}(\mu_{s_t, s_u}, \sigma_{s_t, s_u}) & \text{if } s_t \neq s_u, \mu_{s_t, s_u} > 0, |p(t) - p(u)| = 0 \\ C_{adj}(\mu_{s_t, s_u}, \sigma_{s_t, s_u}) & \text{if } s_t \neq s_u, \mu_{s_t, s_u} > 0, |p(t) - p(u)| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$K \gg 0$ is a very large constant defined by the network designer to make undesirable allocating the same or adjacent frequencies to TRX serving the same area (e.g., placed in the same sector). $C_{co}(\mu, \sigma)$ is the cost due to *co-channel interferences*, whereas $C_{adj}(\mu, \sigma)$ represents the cost in the case of *adjacent-channel interferences* [5, 8].

3 Hybridizations Performed in the Metaheuristics

In this study we use two metaheuristics (SS and DE), and both of them have been hybridized with two extra methods (the same for both) in order to optimize the results (the frequency plans) that the evolutionary algorithms can obtain only by themselves.

The first method we hybridize was explained in [9], and it basically consists in a local search method adapted to our FAP problem (used to optimize the assignment of frequencies to every TRX in each sector of the solution). This method is used to improve every single individual created in each iteration of both algorithms, such as

Algorithm 1

```

1: Solution = {}
2: while ( current_sector <= last_sector ) do
3:   for ( all TRX ∈ current_sector ) do
4:     freq_TRX = random (freq_range)
5:     if ( cost (current_sector, TRX, freq_TRX) < K_Cost ) then
6:       Solution = insert (Solution, TRX, freq_TRX )
7:     else
8:       freq_TRX = search_freq_without_K_cost (current_sector, freq_TRX)
9:       Solution = insert ( Solution, TRX, freq_TRX )
10:    end if
11:  end for
12: end while

```

Fig. 1. Pseudocode for efficient population generator method

we can see in figures 2 and 3 (*localSearch* method). To get a detailed explanation about this method, please, consult reference [9].

On the other hand, the other method we hybridize with the metaheuristics is an efficient algorithm we use every time that a new individual is generated (lines 1 and 8 in figure 2 and line 1 in figure 3). This heuristic method is very important, since it is used when the initial population is generated (and we can state that the better the solutions are from the beginning, the best results will be obtained at the end). The pseudo-code for this method can be observed in figure 1. Basically, this method avoids the highest-cost interferences (with K cost, see equation 2). This is checked in line 5 and case that it is necessary, a better frequency for a sector will be searched (line 8) to make sure that the final result of the method will provide a good solution (choosing the frequencies in a sector so that they do not produce high interferences) to the FAP. Therefore, the metaheuristics used work with a good population of solutions.

4 Scatter Search

Scatter Search (SS) [10, 11] is a metaheuristic used to solve optimization problems (such as the FAP is). The algorithm and its fine adaptation (before the hybridization presented here) to the FAP problem have been explained in detail in a previous work [12]. We can summarize here that the values found for the main parameters of the algorithm after a complete set of experiments are: 40 for the *population size* and 9 for the *RefSet size*. Please, consult references [9, 12] for a detailed explanation of the algorithm adaptation and the experiments performed to fix the parameters of SS to solve the FAP problem. What we want to emphasize here is the hybridization of the algorithm with the methods explained in section 3. The pseudocode of the final hybrid version of SS with the methods explained in section 3 can be observed in figure 2.

As we can see in figure 2, we have followed the typical structure of SS but with some variations to improve it when solving the problem we deal with. Firstly we create a population and improve it using the two methods explained in section 3 (lines 1 and 2, PopulationSize = 40). Using these methods we make possible that the algorithm works with a quite better set of solutions, which makes that the final results are better as well than using only the SS metaheuristic.

Algorithm 2

```

1: population = efficientPopulationGenerator (PopulationSize)
2: population = localSearch (population)
3: RefSet = generateFrom (population)
4: while (not time-limit) do
5:   subSets = subsetGenerationMethod (RefSet)
6:   subSets = combinationMethod (subSets)
7:   RefSet = localSearch (SubSets)
8:   population = efficientPopulationGenerator (PopulationSize)
9:   population = localSearch (population)
10:  RefSet = generateFrom (RefSet, population)
11: end while

```

Fig. 2. Pseudocode for Scatter Search

From the initial population we create the RefSet (RefSetSize = 9) and the algorithm works with its solutions (lines 5-7) until no more new solutions are generated. In that moment the population is regenerated again (lines 8, 9, using again the methods described in section 3) to create a new RefSet from it and the algorithm restarts until the time limit is expired (line 4).

5 Differential Evolution

Differential Evolution (DE) [3, 13] is a metaheuristic which is also very used for function optimization. In a similar way to SS, the scheme of DE has been modified and improved to adapt its general structure to the resolution of the FAP problem. A detailed description of a previous (before the hybridization presented here) DE adaptation to the FAP can be consulted in [14]. To sum up this previous study, we will say that, after a complete set of experiments, the main parameters values are: PopulationSize = 10, F = 0.1, CR = 0.2 and Strategy used = Rand2Bin [14]. Again, what we want to underline in this section is our last improvement developed over the algorithm, which basically consists in the hybridization described in section 3. The pseudocode of this final hybrid version of DE can be seen in figure 3.

As we can observe in figure 3, the general structure of the algorithm is kept, but we have included the methods that we have described in section 3 to improve the algorithm results when solving the FAP problem. In line 1 we can see that we use the

Algorithm 3

```

1: population = efficientPopulationGenerator (PopulationSize)
2: population = evaluatePopulation (population)
3: while (not time-limit) do
4:   for (all individuals  $\in$  population) do
5:     newIndiv = createTrialIndividual (population)
6:     newIndiv = localSearch (newIndiv)
7:     newIndiv = evaluateIndividual (newIndiv)
8:     population = updatePopulation (newIndiv)
9:   end for
10: end while

```

Fig. 3. Pseudocode for Differential Evolution

efficient population generator (PopulationSize = 10 –as we can see here and in section 4, populations for solving FAP problem should not be very big–) and from that population the algorithm evolves making crossover and mutation operations (such as it is described in [14]). However, we make the difference in the evolution in this version because every new individual generated (line 5, using the scheme known as *DE/Rand2Bin* to obtain the trial individual) is improved by using the local search method (line 6) described in section 3. Finally, in line 8, it is checked if the new individual is better than its parent to update the population nicely.

6 Empirical Results

All the experiments performed with the used algorithms have been developed over the real-world instance described in section 2 (2612 TRXs, 711 sectors and only 18 frequencies to give support to all the TRXs). The objective of the experiments was to adjust both algorithms and to show that hybrid metaheuristics can obtain optimum frequency plannings when solving a real-world FAP problem.

In order to provide the results with statistical confidence and check the behaviour of the algorithms within short and long periods of time, we have considered 30 independent runs for each experiment taking in consideration the results every two minutes (such as we can see in the graph of figure 4, where the mean results of the solutions obtained in the 30 executions, in function of cost units, are represented).

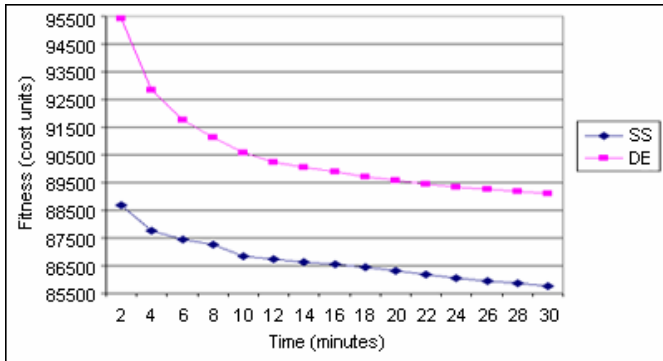


Fig. 4. Results evolution from 2 to 30 min. for the two metaheuristics used

Besides, according to the mathematical formulation of the problem (section 2.1), the results are given in function of the cost that a frequency plan is able to obtain. The smaller the value of the cost is, the better the frequency plan (and the result) will be. Our best results obtained before the hybrid versions of the algorithms described in this paper were the results published in [9] and [12]. These references described how we obtained good frequency plannings using the SS algorithm. The best result obtained here was: 93820.4 cost units on average (after 30 executions too). From that point we began to work to improve the results obtained there. Therefore, from that state we have developed several improvements and fine adjustments over the SS algorithm, mainly the hybridizations explained in this paper, until we have obtained optimum results with the algorithm when solving a realistic FAP problem. On the other hand, a

parallel line of work has laid in the development of an optimal hybrid version with DE to analyze the results obtained using the same hybridization with a second metaheuristic. Figure 4 summarizes the results obtained in our study. As we can see, the evolution and final results for both algorithms are very positive. In fact, to our best knowledge, we have obtained the best results tackling this problem, if we compare our results with all the ones found in the bibliography (using very different metaheuristics, as seen in table 2) which work with the same problem instance (table 2. For more details, please, see the references: [5], [8], [9], and [12]).

In table 1 we can observe the precise cost results in three specific periods of time (2, 10 and 30 minutes). Although SS clearly beats DE, both algorithms have very good results. The best frequency plan obtained by DE has a cost of: 87845.9 units, which clearly improves the results obtained in [5], [8], [12] and it is a very similar result to the obtained in [9]. On the other hand, the best frequency plan obtained with SS has a cost of 84234.5 units, which is to our best knowledge, the best result obtained solving the FAP problem with a real-world instance: the previous best result was 85607.3 interference units, obtained by an (1+2) EA -Evolutionary Algorithm- in 1800 seconds [9]. It is important to highlight that, based on our experience, obtaining differences of 1372.8 units in the long term (1800 seconds) is very difficult.

Table 1. Empirical results (in cost units) of the metaheuristics used for 3 different time limits. We present the best, the average and the standard deviation of 30 independent executions.

	120 seconds			600 seconds			1800 seconds		
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.
DE	92145.8	95414.2	1080.4	89386.4	90587.2	682.3	87845.9	89116.8	563.8
SS	86169.4	88692.7	1124.9	84570.6	86843.8	950.5	84234.5	85767.6	686.3

Table 2. Comparison among the best and mean results (in cost units, after 30 executions) obtained by different metaheuristics when solving the same FAP problem in 30 minutes

	SS	ssGA	LSHR	(1+2) EA	DE	ACO	PBIL
Best	84234.5	85720.3	87743.0	85607.3	87845.9	89305.9	162810.2
Avg.	85767.6	86908.9	88550.3	88574.3	89116.8	90649.9	163394.1

7 Conclusions and Future Work

In this paper we show that the usage of hybrid metaheuristics obtains optimal frequency plans when solving a real-world FAP problem in a GSM network composed of 2612 transceivers (and only 18 frequencies available). But even more, we have improved the results obtained in all the other works found in the bibliography ([5], [8], [9], and [12]) which tackle the FAP problem with the same real-world instance. Besides, we have compared two different approaches, such as we can see in table 1, and results show us that SS obtains better frequency plans than DE (possibly thanks to its better combination method), but in any case, both of them obtain very good results (both have a similar evolution curve, the main difference is the starting point, SS obtains the lowest interference values because it managed to descend faster during the first 120 seconds, see figure 4) thanks to the adjustment and the hybridization performed in each algorithm.

Future work includes the study of other evolutionary algorithms to make deeper analysis with more different metaheuristics. Furthermore, we are working now to obtain more real-world instances, in order to evaluate the algorithms using different instances. In other lines we are also going to use cluster and grid computing in order to speedup all our experiments.

Acknowledgments. This work was partially funded by the Spanish Ministry of Education and Science and FEDER under contract TIN2005-08818-C04-03 (the OPLINK project). José M. Chaves-González is supported by the research grant PRE06003 from Junta de Extremadura (Spain). Thanks also to the Polytechnic Institute of Leiria, for the economic support offered to Marisa Maximiano.

References

1. Blum, C., Roli, A.: Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys* 35, 268–308 (2003)
2. Glover, F., Laguna, M., Martí, R.: Scatter search. *Advances in Evolutionary Computing: theory and Applications*. In: Ghosh, A., Tsutsui, S. (eds.) *Natural Computing Series*, pp. 519–537. Springer, New York (2003)
3. Price, K., Storn, R.: Differential Evolution: A Simple Evolution Strategy for Fast Optimization. *Dr. Dobb's Journal* 22(4), 18–24 (1997)
4. *Wireless Intelligence* (2006), <http://www.wirelessintelligence.com>
5. Luna, F., Blum, C., Alba, E., Nebro, A.J.: ACO vs EAs for Solving a Real-World Frequency Assignment Problem in GSM Networks. In: *GECCO 2007*, London, UK, pp. 94–101 (2007)
6. Eisenblätter, A.: Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds. PhD thesis, Technische Universität Berlin (2001)
7. Kuurne, A.M.J.: On GSM mobile measurement based interference matrix generation. In: *IEEE 55th Vehicular Technology Conference, VTC Spring 2002*, pp. 1965–1969 (2002)
8. Domínguez-González, D., et al.: Using PBIL for Solving a Real-World Frequency Assignment Problem in GSM Networks. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) *EPIA 2007*. LNCS (LNAI), vol. 4874, pp. 207–218. Springer, Heidelberg (2007)
9. Luna, F., et al.: Metaheuristics for Solving a Real-World Frequency Assignment Problem in GSM Networks. In: *GECCO 2008*, Atlanta, GE, USA (2008) (pending of publication)
10. Martí, R., Laguna, M., Glover, F.: Principles of Scatter Search. *European Journal of Operational Research* 169, 359–372 (2006)
11. Laguna, M., Hossell, K.P., Martí, R.: *Scatter Search: Methodology and Implementation*. Kluwer Academic Publishers, Norwell (2002)
12. Chaves-González, J.M., Vega-Rodríguez, M.A., et al.: SS vs PBIL to Solve a Real-World Frequency Assignment Problem in GSM Networks. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, A.Ş., Yang, S. (eds.) *EvoWorkshops 2008*. LNCS, vol. 4974, pp. 21–30. Springer, Heidelberg (2008)
13. Price, K., Storn, R.: DE (2008), <http://www.ICSI.Berkeley.edu/~storn/code.html>
14. Maximiano, M., Vega-Rodríguez, M.A., et al.: Solving the Frequency Assignment Problem with Differential Evolution. In: *SoftCOM 2007*, Split-Dubrovnik, Croatia, pp. 1–5 (2007)

PSO for Selecting Cutting Tools Geometry

Orlando Duran, Nivaldo Rodriguez, and Luiz Airton Consalter

Pontificia Universidad Catolica de Valparaiso
Av. Brasil 2241, Chile

FEAR Universidade de Passo Fundo, P. Fundo, RS, Brasil
{nivaldo.rodriguez, orlando.duran}@ucv.cl, lac@upf.br

Abstract. Today an extensive variety of cutting tool is available to suit different cutting situations and needs. Cutting tool selection is a very important subtask involved in process planning activities. This paper proposes the application of the particle swarm optimization (PSO) process for the definition of the optimal cutting tool geometry. This optimization model was developed considering an approach to the macro-level optimization of tool geometry in machining model satisfying technological constraints. To determine the most appropriate configuration of the algorithm and its parameters a set of tests were carried out. The different parameters settings were tested against data obtained from the literature. A variety of simulations were carried out to validate the performance of the system and to show the usefulness of the applied approach. The definition of proposed system also has presented the following main conclusion: through the utilization of the PSO approach, the selection of the appropriate cutting tool geometry is possible in real world environments.

Keywords: PSO, Cutting Tools, Computer Aided Manufacturing.

1 Introduction

Cutting tool geometry selection is a very important and very complex subtask involved in process planning systems. Figure 1 shows a description of the main angles of a generic cutting edge in turning operations. One of the main factors that made this subtask so difficult is the large number of available tools and to the wide variety of considerations that are involved in this domain. An extensive variety of tool geometries is available today to suit different situations. As a consequence, the process planner has to work with voluminous machining data handbooks before deciding on the best tool geometry for the machining task at hand. In addition, the rules for selection are often vague and contradictory; hence, it is almost impossible for engineers or machinists to check all available cutters to find the most appropriate one. Up to now, no article has considered technological factors when selecting effective cutting tools and their geometries. There are a number of papers in the literature that suggest the use of soft computing and evolutionary approaches to represent the knowledge of manufacturing operations. An example was presented by Knapp and Wang [1] who used an Artificial Neural Networks (ANN) in process planning. Cutting tool selection using ANN was introduced by Dini

[2]. Monostori, Egresits, Hornyk and Viharos [3] proposed a set of models to estimate and classify tool wear using neural networks based techniques. Liao and Chen [4] have used back-propagation neural networks for modelling and optimising the creep feed grinding of aluminium with diamond wheels. Recently, there have been others attempts to determine in an automated way the optimal tool for a given machining operation. Zhao, Ridgway and Al-Ahmari [5] proposed a knowledge based system that performs the selection of cutting tools and conditions for turning operations. Arezoo, Ridgway and Al-Ahmari [6] developed a knowledge based system for selection of cutting tools and conditions in turning operations. The proposed system selects the toolholder, insert and cutting conditions. Mookherjee and Bhattacharyya [7] presented an expert system, which automatically selects the turning tool insert or milling insert, the material and the geometry, based on the requirement of the users. Toussaint and Cheng [8] propose a web-based engineering approach to developing a design support system using case-based reasoning (CBR) technology for helping in the decision-making process when choosing cutting tools. Dereli and Filiz [9] have proposed an approach using genetic algorithms (GAs), which mathematically determines from the 3D features of a mechanical part the best tooling arrangement to optimize the tooling set. McMullen, Clark, Albritton and Bell [10] have focused on optimizing production sequences by limiting the number of tooling replacements. Their approach is concerned with finding relevant tooling sequences while trying to maintain tool-wear uniformity through correlation and heuristic methods. The tool selection procedures developed thus far use expert systems approaches based on productivity or accessibility factors and others different than the impact that the cutting edge geometry may cause in machinability, surface roughness, chip breaking and cutting temperature. Here we propose the application of a biological inspired algorithm for selecting cutting tool regarding technological aspects and defining the cutting tool geometry using an empirically obtained relationship between cutting tool angles and the cutting performance as the main objective function. Kaldor and Venuvinod [11] proposed a macro level optimization of cutting tool geometry. They defined a model to describe the interactions among the tool material, workpiece material and the tool geometry. The authors proposed a geometric number called σ_r , that characterizes the ideal set of tool angles at which a given tool material functions best (in terms of the tool life) in the context of machining a given work material (equation 1):

$$\sigma_r = \cos^2 \lambda_s \frac{\sin\left(\sigma_n + \frac{\beta_n}{2}\right) \sin\left(\frac{\beta_n}{2}\right) \left(\frac{\beta_n}{2} + \frac{1}{2} \sin\beta_n\right) - \cos\left(\sigma_n + \frac{\beta_n}{2}\right) \cos\left(\frac{\beta_n}{2}\right) \left(\frac{\beta_n}{2} - \frac{1}{2} \sin\beta_n\right)}{\left(\frac{\beta_n}{2}\right)^2 - \left(\frac{1}{2} \sin\beta_n\right)^2} \quad (1)$$

Where σ_r corresponds to a nondimensional stress factor that characterizes a cutting tool geometry; α_n represents the tool normal clearance angle, β_n represents normal wedge angle, γ_n represents tool normal rake angle and λ_n represents tool cutting edge inclination angle. (Figure 1) shows the geometric specification of single point tools based on ISO 3002/1. Figure 2 shows the relationships among σ_r and different cutting tool angles. Kaldor and Venuvinod [11] examined over 80 published sources listing machinability data and relevant experimental results encompassing a wide range of cutting operations, tool and work materials and cutting conditions. They found that

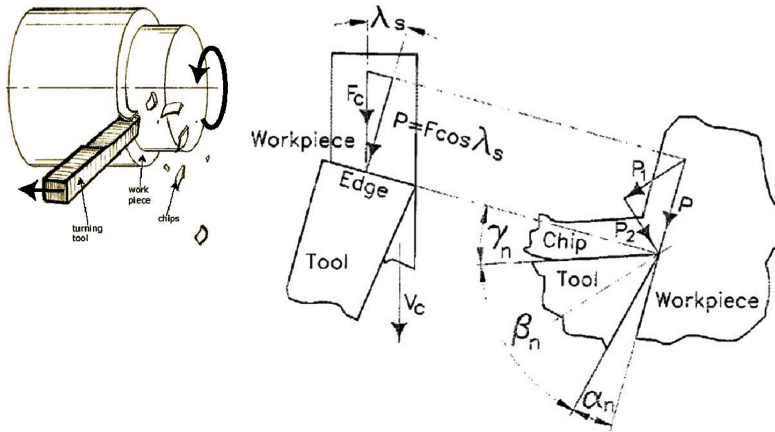


Fig. 1. Cutting geometry Model

the following quantitative semiempirical expression relating tool and work material properties can be assumed with a confidence greater than 95%(eq.2):

$$G = \log\{Sb/\{EU\}^2\} + 4.73 \tag{2}$$

Where G number represents the ideal relationship between tool material and work-piece material providing maximum tool life; Sb represents the bending strength of the tool material; E is the modulus of elasticity of the tool material and U corresponds to the specific cutting energy of the work material, all expressed in GPa. In consequence, the ideal tool geometry for a given cutting situation can be obtained through the minimization of the difference between G and σ_r .

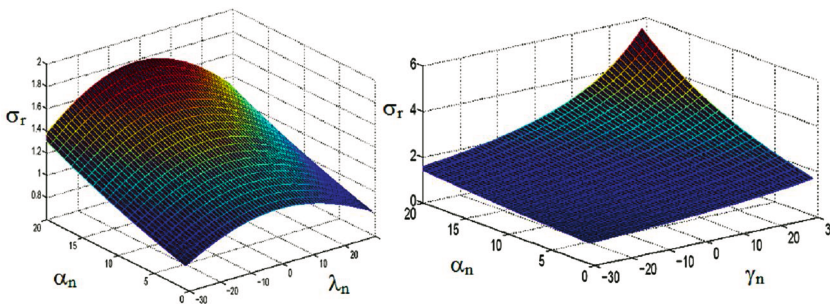


Fig. 2. Relationship among α_n , γ_n , λ_n and σ_r

As it was mentioned, the data provided by the aforementioned authors is based on a collection of results from laboratory experiments and specific cutting operations. However, there are insufficient machining cases available until now. This fact has become the bottleneck for the utilization of the proposed model. Therefore, more research is needed in this field to provide a basis for practical use of the model.

2 Particle Swarm Optimization

The Particle Swarm Algorithm (PSO) is an adaptive algorithm based on a social-psychological metaphor [12]. A population of individuals adapt by returning stochastically towards previously successful regions in the search space, and are influenced by the successes of their topological neighbors. Most particle swarms are based on two sociometric principles. Particles fly through the solution space and are influenced by both the best particle in the particle population and the best solution that a current particle has discovered so far. The best particle in the population is typically denoted by (global best), while the best position that has been visited by the current particle is denoted by (local best). The (global best) individual conceptually connects all members of the population to one another. That is, each particle is influenced by the very best performance of any member in the entire population. The (local best) individual is conceptually seen as the ability for particles to remember past personal successes. The dimension of the search space is D . Therefore, the current position of a particle in this space at the moment t is given by a vector $x(t)$, with D components. Its current velocity is $v(t)$. The best position found up to now by this particle is given by a vector $p(t)$. Lastly, the best position found by informants of the particle is indicated by a vector $g(t)$. In general, we will write simply x , v , p , and g . The d th component of one of these vectors is indicated by the index d , for example x_d . With these notations, the equations of motion of a particle are, for each dimension d :

$$v_d = c_1 v_d + c_2 (p_d - x_d) + c_3 (g_d - x_d) \quad (3)$$

$$x_{d+1} = x_d + v_d \quad (4)$$

The confidence coefficients are defined in the following way:

- c_1 is constant (confidence in its own movement);
- c_2 and c_3 (respectively confidence in its best performance and that of its best informant) are randomly selected with each step of time according to a uniform distribution in a given interval $[0, c_{\max}]$.

In the rest of this paper we refer c_1 as inertia and c_{\max} as correction factor.

3 Proposed Approach

In essence, the proposed approach points to the use of particle swarm optimization for selecting adequate cutting tool geometry in a given combination of material workpiece and cutting tool material. An optimization model that uses a fitness function that embodies the difference between a given (as a function of a given set of tool angles) and G (for a given combination of tool material and workpiece material) is presented. The PSO algorithm performs the evaluation of a selected numbers of individuals (that represent a set of feasible tool angles) until a given termination a criterion is satisfied.

Therefore, once the set of tool angles is obtained through the PSO algorithm, an expert system can recommend the appropriate cutting angles for a given situation. In this experiment the following objective function is selected:

$$Minf(\lambda, \alpha, \gamma) = ABS(\sigma_r - G) \tag{5}$$

Subject to:

$$\begin{aligned} -20^\circ &\leq \lambda_n \leq 20^\circ \\ 0^\circ &\leq \alpha_n \leq 20^\circ \\ -20^\circ &\leq \gamma_n \leq 20^\circ \\ \beta_n &= 90^\circ - (\alpha_n + \gamma_n) \end{aligned}$$

The three angles have to be within predefined ranges. The limits of those ranges are based upon technological and operational factors.

In the model of Kaldor and Venuvinod, any combination of angles within the specified range are permissible and may be selected so as to provide optimum cutting tool geometry. This research has applied a PSO algorithm for tool angles definition, considering Kaldor and Venuvinod model while satisfying angle constraints. It starts with some valid solutions generated randomly, then makes a random change to them and accepts the ones whose fitness function is reduced. The process is repeated until no changes for fitness function reduction can be made or until other finalization condition is reached. To determine the most appropriate configuration of the algorithm and its parameters a set of 20 tests were carried out. The different algorithm settings were tested against data obtained empirically from Kaldor and Venuvinod [11]. A sub set of these data is shown in Table 1. The values of the control parameters tested and compared are reported in Table 2. The average results of 20 runs for each set of parameters are shown in Table 3. As it can be observed, test number 7 corresponds to the minimum average error. Therefore, we selected the following parameters: correction factor equal to 0.4 and inertia equal to 1.0. The swarm size is equal to 40 elements.

Table 1. Subset of empirical data used to set the PSO algorithm parameters

α_n	γ_n	λ_s	σ_r	α_n	γ_n	λ_s	σ_r	α_n	γ_n	λ_s	σ_r
10,0	-4,0	-8	1,82	10,0	16,0	0,0	2,82	13,0	5,0	0,0	2,42
18,0	-18,0	-16,0	1,82	13,0	15,0	0,0	2,96	8,0	24,0	4,0	2,79
6,2	-16,0	-20,0	1,25	5,0	8,0	0,0	1,83	8,0	8,0	0,0	2,07
0,0	-3,0	0,0	1,32	4,8	9,6	-2,6	1,85	8,9	18,0	-1,3	2,34
1,0	0,0	0,0	1,4	4,8	9,7	-2,6	1,86	5,0	6,0	0,0	1,78
0,0	3,0	0,0	1,41	4,0	15,0	0,0	1,94	8,0	15,0	0,0	2,34
5,0	-7,5	0,0	1,5	4,8	14,0	6,6	1,96	7,0	-0,7	-4,0	1,74
4,8	-6,1	-3,5	1,51	4,8	14,0	6,6	1,98	4,8	4,8	-1,3	1,73
4,8	-6,1	-3,5	1,51	6,0	14,0	6,0	2,08	4,8	4,8	-1,3	1,73
6,0	-6,0	6,0	1,56	8,0	9,8	5,7	2,11	6,0	0,0	0,0	1,7
4,0	0,0	0,0	1,58	7,9	10,4	4,1	2,14	9,8	-12,0	-7,0	1,63
3,5	3,5	3,5	1,61	5,8	15,5	4,1	2,18	10,0	-10,0	-13,0	1,62

Table 2. Inertia and Correction parameters tested

Test Num.	Inertia weight	Co-rrect. Factor	Test Num.	Inertia weight	Co-rrect. Factor	Test Num.	Inertia weight	Co-rrect. Factor	Test Num.	Inertia weight	Co-rrect. Factor
1	0,2	0,5	6	0,2	1,0	11	0,2	1,5	16	0,2	2,0
2	0,4	0,5	7	0,4	1,0	12	0,4	1,5	17	0,4	2,0
3	0,6	0,5	8	0,6	1,0	13	0,6	1,5	18	0,6	2,0
4	0,8	0,5	9	0,8	1,0	14	0,8	1,5	19	0,8	2,0
5	1,0	0,5	10	1,0	1,0	15	1,0	1,5	20	1,0	2,0

Table 3. Average results obtained by the set of tested parameters

Test Number	Average Error	Test Number	Average Error	Test Number	Average Error	Test Number	Average Error
1	0,0642210	6	0,0491310	11	0,0026906	16	0,0027164
2	0,0016095	7	0,0014614	12	0,0052967	17	0,0796200
3	0,0014934	8	0,0068968	13	0,0770290	18	0,1447400
4	0,0025370	9	0,0207910	14	0,1450600	19	0,2424900
5	0,0031807	10	0,0752900	15	0,1920500	20	0,3804800

4 Application Example and Results

An expert system was developed for cutting tool selection from CAM databases. Each CAM system has a cutting-tool database and the appropriate cutting tool may be selected from it. Based on the answer of the developed expert system, one or a set of cutting tools are selected from database for a given combination of tool material and workpiece material. Since no cutting tool in the database is likely to meet 100 percent of the σ_r , a sort of Euclidean distance algorithm was implemented for the automatic search of the database (i.e. nearest neighbours algorithm). This search algorithm produces objects (cutting tools) that are geometrically close to the optimal one. Therefore, it can find one or more cutting tools in the database that meet the optimal geometric properties recommended by the G value or at least, a near approximation to it.

To implement such an interaction a series of Matlab functions calls were written using the ODBC protocol. The system is composed of a graphical user interface, a PSO-based expert system, an interconnected machining tool database and an interface for a CAMsoftware package. Figure 3, shows the dialog box where the user inputs the information about the workpiece material and identifies the type of tool material. The results of the searching algorithm are presented to the end user, which is in charge of the final selection of the specific tool among the candidates isolated by the expert system in coordination with the searching algorithm (Figure 4). The system returns all the tools that meet the cutting angles recommended by the genetic algorithm. Several other parameters can be extracted or computed by the system. The cutting speed and feed, are calculated by an intelligent module that were implemented using neural networks approach. The maximum depth of cut is obtained by computing the effective cutting edge for the selected tool. This values are used for calculate the Maximum

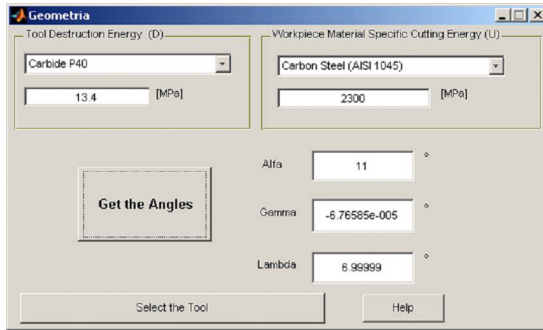


Fig. 3. Graphical user Interface - Input dialog box

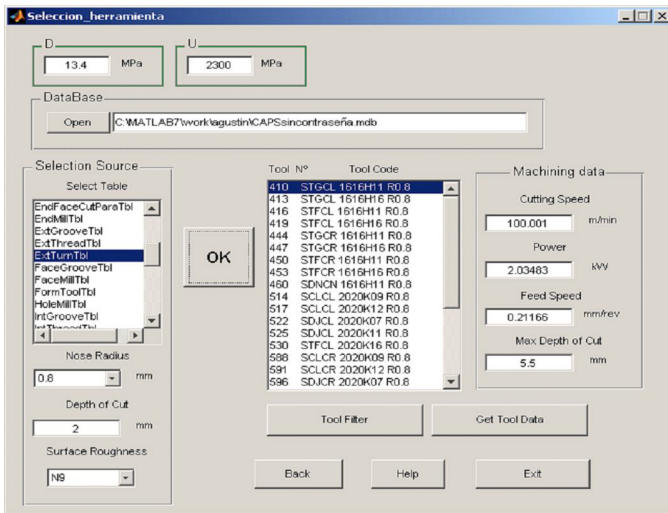


Fig. 4. Window with the result of the searching algorithm

Power consumption. All this information is presented to the user in a dialog box. Finally, the user can select or define the appropriate tool from the CAM database.

5 Conclusions

A PSO approach for selecting the optimal cutting tool geometry was suggested. The proposal points to the definition of an optimal set of cutting tool angles for a given machining situation. A variety of simulations were carried out to validate the performance of the algorithm and to show the usefulness of the suggested approach. Further evolution of the approach is possible. Future work aims at the integration of this algorithm with other commercial CAM systems for identification of a list of cutting tools stored in a database to aid the CAM operator to select the more appropriate cutting tool in a given operation.

References

1. Knapp, G.M., Wang, H.p.: Acquiring, storing and utilizing process planning knowledge using neural networks. *J. of Int. Manufacturing* 3, 333–344 (1992)
2. Dini, G.: A neural approach to the automated selection of tools in turning. In: *Proc. of the 2nd AITEM Conf., Padova, September 18-20*, pp. 1–10 (1995)
3. Monostori, L., Egresits, C.S., Hornyk, J., Viharos, Z.S.J.: Soft computing and Irbid AI approaches to intelligent manufacturing. In: *Proc. of 11th Int. Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Castellon*, pp. 763–774 (1998)
4. Liao, T.W., Chen, L.J.: A Neural Network Approach for Grinding Processes Modeling and Optimization. *Int. J. of Machine Tools and Manufacture* 34(7), 919–937 (1994)
5. Zhao, Y., Ridgway, K., Al-Ahmari, A.M.A.: Integration of CAD and a cutting tool selection system. *Computers and Industrial Engineering Archive* 42(1), 17–34 (2002)
6. Arezoo, B., Ridgway, K., Al-Ahmari, A.M.A.: Selection of cutting tools and conditions of machining operations using an expert system. *Computers in Industry* 42(1), 43–58 (2000)
7. Mookherjee, R., Bhattacharyya, B.: Development of an expert system for turning and rotating tool selection in a dynamic environment. *Journal of materials processing technology* 113(1-3), 306–311 (2001)
8. Toussaint, J., Cheng, K.: Web-based CBR (case-based reasoning) as a tool with the application to tooling selection. *Int. J. Adv. Manuf. Technol.* 29, 24–34 (2006)
9. Dereli, T.R., Filiz, I.H.S.: Optimisation of process planning functions by genetic algorithms. *Int. J. Comput. Ind. Eng.* 36(2), 281–308 (1999)
10. McMullen, P.R., Clark, M., Albritton, D., Bell, J.: A correlation and heuristic approach for obtaining production sequences requiring a minimum of tool replacements. *Int. J. Comput. Oper. Res.* 30, 443–462 (2003)
11. Kaldor, S., Venuvinod, P.K.: Macro level Optimization of Cutting Tool Geometry. *Journal of Manufacturing Science and Engineering* 119, 1–9 (1997)
12. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ*, pp. 1942–1948 (1995)

A Hybrid Ant-Based System for Gate Assignment Problem

Camelia-M. Pinte¹, Petrica C. Pop², Camelia Chira¹, and D. Dumitrescu¹

¹ Babes-Bolyai University, Kogalniceanu 400084, Cluj-Napoca, Romania

² North University, V.Babes 430083, Baia-Mare, Romania

{cmpinte¹, cchira, ddumitr}@cs.ubbcluj.ro, pop_petrica@yahoo.com

Abstract. This paper presents an ant system coupled with a local search applied to an over-constrained airport gate assignment problem (*AGAP*). In the airport gate assignment problem we are interested in selecting and allocating aircrafts to the gates such that the total passenger connection time is minimized. Our algorithm uses pheromone trail information to perform modifications on *AGAP* solutions, unlike traditional ant systems that use pheromone trail information to construct complete solutions. The algorithm is analyzed and compared with tabu search heuristic and Ant Colony System metaheuristic.

Keywords: ant algorithms, assignment problems, constraints, t-test.

1 Introduction

An airport is a transfer point between travel domains in which two unlike carriers, motor vehicles and airplanes, must interact to exchange people and luggage. The main purpose of flight-to-gate assignments is to assign aircrafts to suitable gates so that passengers can conveniently embark and disembark. The distance a passenger has to walk in any airport to reach various key areas including departure gates, baggage belts and connecting flights provide for an important performance measure of the quality of any airport.

The *Airport Gate Assignment Problem (AGAP)* seeks optimal flight-to-gate assignments to minimize the total passenger connection times. *AGAP* can be viewed as an instance of the Quadratic Assignment Problem (*QAP*).

Braksm^a and Shortreed [2] have described one of the first attempts to use quantitative means to minimize intra-terminal travel into a design process. Babic et al. [1] modeled the *AGAP* as a 0-1 integer program and used branch-and-bound algorithm to find solutions where transfer passengers were not considered. Yan and Chang [11] proposed a network model that was formulated as a multi-commodity flow problem. Network-based simulation of aircrafts at gates in airport terminals were considered by Cheng [5,6]. Wang et al. [9] presented an optimization algorithm for solving *AGAP* and as an application, they achieved in optimizing the dispatch of 334 airlines by using of 18 gates in the case of Beijing International Airport.

The most notable heuristic methods developed for *AGAP* are the following: tabu searches Xu and Bailey [10], Ding et al. [7], a hybrid simulated annealing Ding et al.

[7] and a genetic algorithm Gu and Chung [8]. Ding et al. in [7] focused on the AGAP with the objective of minimizing distance costs of the over constrained gate assignment problem, minimizing the number of ungated aircraft and the total walking distances.

This paper presents a hybrid ant-local search system for the airport gate assignment problem, denoted *HAS-AGAP*, which works better than known heuristics (presented in the next section). The paper is organized as follows. The definition, a quadratic programming formulation for the *AGAP* and a greedy method for minimizing ungated flights are presented in Section 2. Our hybrid ant-local search is presented in Section 3. Then, the performance of our algorithm is analyzed by comparing with Ant Colony System. Finally, in the last section some conclusions are drawn.

2 The Airport Gate Assignment Problem

The *Airport Gate Assignment Problem (AGAP)* seeks feasible flight-to-gate assignments so that the total passenger walking distances and consequently connection times are minimized. Distances in airports that are taken into account are those from check-in to gates in the case of embarking or originating passengers, from gates to check-out in the case of disembarking or destination passengers and from gate to gate in the case of transfer or connecting passengers.

In the over-constrained case the number of aircrafts exceeds the number of available gates and the distance from the apron area to the terminal for aircraft assigned to these areas is considered. Let us consider the following notations:

- N is the set of flights arriving at (and/or departing from) the airport;
- M is the set of gates available at the airport;
- n is the total number of flights;
- m is the total number of gates;
- a_i is the arrival time of flight i ;
- d_i is the departure time of flight i ;
- w_{kl} is the walking distance for passengers from gate k to gate l ;
- f_{ij} is the number of passengers transferring from flight i to flight j .

Two dummy gates are used. Gate 0 represents the entrance or exit of the airport and gate $m + 1$ represents the apron where flights arrive at when no gates are available. In order to model the *AGAP* as a quadratic program there were considered the binary variable y_{ik} denoting that flight i is assigned to gate k if $y_{ik} = 1$ and $y_{ik} = 0$ otherwise, where ($0 < k < m + 1$).

The walking distance between gate k and the airport entrance or exit is w_{k0} , f_{0i} represents the number of originating departure passengers of flight i and f_{i0} is the number of the disembarking arrival passengers of flight i . The walking distance between the apron and gate k is w_{m+1k} . The objectives refer to minimizing the number of flights assigned to the apron and the total walking distance (the walking distance of transfer passengers, disembarking arrival passengers and originating departure passengers). *AGAP* may be formulated as follows, see [7].

Minimize the number of flights assigned to the apron:

$$\text{minimize } \sum_{i=1}^n y_{im+1}$$

Minimize the total walking distance:

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{m+1} \sum_{l=1}^{m+1} f_{ij} w_{kl} y_{ik} y_{jl} + \sum_{i=1}^n \sum_{l=1}^{m+1} f_{oi} w_{ol} + \sum_{i=1}^n \sum_{l=1}^{m+1} f_{io} w_{lo} \tag{1}$$

$$\text{s.t. } \sum_{k=1}^{m+1} y_{ik} = 1, \forall i, 1 \leq i \leq n$$

$$a_i < d_i, \forall i, 1 \leq i \leq n \tag{2}$$

$$(d_i - a_j)(d_j - a_i)y_{ik}y_{jk} \leq 0, \forall i, j, 1 \leq i, j \leq n, k \neq m + 1 \tag{3}$$

$$y_{ik} \in \{0, 1\}, \forall i, 1 \leq i \leq n, \forall k, 1 \leq k \leq m + 1. \tag{4}$$

Constraint (1) ensures that every flight must be assigned to one and only one gate or assigned to the apron. Constraint (2) specifies that each flights departure time is later than its arrival time. Constraint (3) says that two flights schedule cannot overlap if they are assigned to the same gate. Condition (4) disallows any two flights to be scheduled to the same gate simultaneously except if they are scheduled to the apron.

The above mentioned model is a 0-1 integer programming model with a quadratic objective function. As in [7], the first step for the over-constrained AGAP is to minimize the number of flights that need be assigned to the apron. The minimal number of flights can be calculated by greedy algorithm described in the following. After sorting all the flights by the departure time, flights are assigned one by one to the gates. Any flight is assigned to an available gate with latest departure time. If there are no gates available, the flight will be assigned to the apron.

Correctness of the greedy algorithm is proved in [7]. The greedy solution gives the optimal number of flights that can be scheduled in gates and also helps us to get a feasible initial solution (L^+), which is used in the proposed metaheuristic.

3 Solving AGAP with an Hybrid Ant System

The idea of imitating the behaviour of the ants for finding good solutions to the combinatorial optimization problems was initiated by Dorigo, Maniezzo and Colorni [4]. The principle of these methods is based on the way ants search for food and find their way back to the nest.

In the best ant system to date (Dorigo and Gambardella [3]), pheromone trails are not only modified *locally* by the artificial ants during or just after the construction of a new solution, but also *globally*, considering the best solution generated by all the ants at a given iteration or even the best solution ever constructed.

The introduced *Hybrid Ant System (HAS-AGAP)* uses the pheromone trails in a non-standard way: in proceeding applications of ant systems pheromone trails were exploited to build a completely new solution. In our paper pheromone trails are used

to modify an existing solution, in the spirit of a neighbourhood search. After an artificial agent has modified a solution, taking into account only the information given by the pheromone trail intensity, an improvement phase that consists in performing a fast local search that takes into consideration just the objective function is applied.

Adding local search to ant systems has also been identified as very promising by other researchers such as Dorigo and Gambardella [3] in the case of the TSP, where they design an ant system almost as efficient as the best implementations of the Lin and Kernighan heuristic by adding a simple *3-opt* phase after the construction phase. In *HAS-AGAP* each ant is associated with a problem solution that is first modified using pheromone trail and later is improved using a local search mechanism.

3.1 Neighbourhood Search

A neighbourhood search move is the operation that maps one solution π to another solution π_1 . The attractiveness of a move from π to π_1 can be examined by calculating the cost (objective function value) improvement of π_1 compared to that of π . In our neighbourhood search we used three types of neighbourhoods, which prevailed in AGAP applications based on concepts from Xu et al. [10] and Ding et al. [7]:

- The Insert Move: move a single flight to a gate other than the one it currently assigns.
- The interval Exchange Move: Exchange two flights intervals in the current assignment, where a flight interval consists of one or more consecutive flights in one gate.
- The Apron Exchange Move: Exchange one flight which has been assigned to the apron with a flight that is assigned to a gate currently.

The elements of the algorithm based on ant colony system coupled with a local search for solving the AGAP, are:

- Graph representation: the nodes of the graph are locations, and the goal of the ants is to visit all the locations and match an activity to it, forming a bijective mapping from activities to locations. At time t each edge (i,j) is labelled by a trail intensity $\tau_{ij}(t)$.
- Initially the ants are randomly placed in the nodes of the graph.
- At each iteration an ant moves to a new node. When an ant decides which node is the next move it does so with a probability based on the distance to that node and the amount of trail intensity on the connecting edge. The inverse of distance to the next node, is known as the visibility η_{ij} .
- At each moment evaporation takes place to stop the intensity trails increasing unbounded. The values of rate evaporation ρ , are in the unit interval $[0,1]$.
- A tabu list: ants memorizing locations that they have visited, so that they never visit them again.
- A tabu list: ants memorizing the activities that have been mapped to the visited locations.
- In order to favour the selection of an edge that has a high pheromone value τ and high visibility value η , a probability is considered. If we denote by J_i^k the unvisited neighbours of node i by ant k and $u \in J_i^k$, then p_{iu}^k is defined as the probability of choosing $j = u$ as the next node if $q > q_0$ (the current node is i) and is given by the rule (see [4]):

$$P_u^k = \frac{[\tau_{ij}][\eta_{ij}]^\beta}{\sum_{o \in J_i^k} [\tau_{io}(t)][\eta_{io}(t)]^\beta} \tag{5}$$

where β is a parameter used for tuning the relative importance of edge length in selecting the next node. Consider a random variable q uniformly distributed over $[0, 1]$ and q_0 with $0 \leq q_0 \leq 1$. If $q \leq q_0$ the next node j is chosen as:

$$j = \arg \max_{u \in J_i^k} \{ \tau_{iu}(t) [\eta_{iu}(t)]^\beta \} \tag{6}$$

- Manipulation of solutions: local search

Local search consists in applying the above mentioned neighbourhood search based on three types of neighbourhoods with first improving strategy to a solution. We denote by S the solution space, V the neighbourhood structure containing the three types of neighbourhoods and by c the cost function. The local search procedure is:

```

Neighbourhood search procedure
Select a starting solution s[0] from S
Repeat
    Select s such that c(s) < c(s[0])
    Replace s[0] by s
Until c(s) > c(s[0]) for all s from V(s[0])
s[0] is the approximation of the optimal solution.
    
```

This procedure examines all the possible swaps according to the considered neighbours and immediately performs an improving swap, if one is found. The update of the pheromone trails is done differently from the standard ant system where all the ants update the pheromone trails with the result of their computations. The pheromone trails are updated by taking into account only the best solution produced by the local search. First, all the pheromone trails are weakened by setting

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} \tag{7}$$

A parameter ρ ($0 < \rho < 1$) that controls the evaporation of the pheromone trail, close to 0 implies that the pheromone trails remain active a long time, while a value of ρ close to 1 implies a high degree of evaporation. Then, the pheromone trails are reinforced by considering only the best solution s^* generated by the system so far and setting:

$$\tau_{ij^*} = \tau_{ij^*} + \frac{\alpha}{c(s^*)} \tag{8}$$

4 Numerical Results

HAS-AGAP is compared with tabu search heuristic and with Ant Colony System [3]. For the comparison, a large set of problem instances is considered (see Tables 1-2), with sizes between 100×16 ($m \times n$) and 460×34 and used in the experiments [7]. For small test data, (e.g. less than 25×5 as in [7]) when all the flights can be assigned to

the gates, or using the apron for Tabu Search [9], ACS and HAS-AGAP found for all problems optimal solutions, like using the Brute Force Method.

We considered as in [7] an airport with two parallel sets of terminals, where gates are symmetrically located in the two terminals. The distance between the check-in / check-out point to gate 1 (and gate 2) is 5 units. The distance between two adjacent gates in one terminal (e.g., gate 1 and gate 3) is 1 unit and the distance between two parallel gates in different terminals (e.g., gate 1 and gate 2) is 3 units.

It is assumed that the passengers can only walk horizontally or vertically. It is used the Manhattan Distance. If one passenger wants to transfer from gate 3 to gate 2, the walking distance is 4 units.

The arrival time a_i of flight i is randomly generated in the interval $[10i, 10i+7]$ and the departure time d_i is generated in the interval $[a_i+60, a_i+69]$. There is considered, as in real life, the total number of passengers in a flight within an interval $[300,400]$. The number of transfer passengers will increase if flight schedules are close, but not too close. For each flight i , the number of disembarking passengers whose destination is this airport f_{i0} and the number of embarking passengers whose origin is this airport f_{0i} are generated within the interval $[1, 100]$.

The parameters chosen for the considered ant-based algorithms are: $\alpha=1, \beta=5, \rho=0.01, \tau_0 = 0.01$ and $q_0=0.9$. The number of ants is considered the same with the number of gates (m). The number of iterations is considered $2m \cdot 2500$. In our implementations we considered two test cases:

- Randomized large input: the test cases generated are categorized into 10 different sizes with 10 cases in each size;
- Fully packed input: all the flights are fully packed (i.e. one flight is followed by another and there are no gaps between the consecutive flights);

The computational results obtained using our algorithm HAS-AGAP, compared with the Tabu Search (TS) [7] and Ant Colony System [3], are presented in the next tables.

Table 1. Test Set: randomized large input

Size	TS-cost	TS-time	ACS-cost	ACS-time	HAS-cost	HAS-time
100×16	101456	26.12	100927	5564.48	100782	5343.53
160×20	160345	64.23	160332	9247.50	158934	10762.48
220×24	215342	80.67	213987	11422.25	213243	16956.20
280×28	325412	86.24	324647	12965.20	324825	17043.87
340×32	365521	88.64	365866	13245.57	365784	18564.13
400×36	481256	125.36	480867	17453.15	480432	21130.52
460×40	498688	238.66	496334	17988.68	495487	17122.33
520×44	566325	425.66	562887	18966.22	563147	18425.40
580×48	674889	504.30	671225	20501.50	671322	19688.57
640×52	716455	702.36	710288	25457.32	709058	25036.28

We find that among these groups of test cases, HAS-AGAP performs better than TS and ACS for AGAP and having almost the same running time in seconds as ACS, but longer running time than TS (see the results on Tables 1-3). The computer used for numerical tests: AMD 2600, 1.9 GHz, 512GB RAM. It is possible to obtain better values, using a computer with higher performances or/and using better value for the parameters.

Table 2. Test Set: fully packed input

Size	TS-cost	TS-time	ACS-cost	ACS-time	HAS-cost	HAS-time
100×16	221485	24.80	219432	4214.42	218865	4205.26
140×18	350426	32.96	347885	6210.24	346299	5833.45
180×20	522830	44.25	519887	14335.70	519758	13742.48
220×22	647239	71.12	648335	17567.80	647232	16956.20
260×24	871246	86.44	870843	18499.25	870916	18823.15
300×26	1015284	89.20	1013244	20430.18	1013428	20104.28
340×28	1143906	105.15	11344952	27211.26	1144823	26521.30
380×30	1434850	143.42	1434695	31218.36	1434507	31126.22
420×32	1572625	175.03	1572439	38846.52	1572483	36228.18
460×34	1951522	239.11	1949927	42185.63	1949912	40815.25

Table 3. Results of *t-test*

<i>t-test</i> results	Test Set: randomized large input			Test Set: fully packed input		
	ACS-HAS	TS-ACS	TS-HAS	ACS-HAS	TS-ACS	TS-HAS
<i>t</i>	2.31	2.81	3.23	1.87	2.32	2.73
<i>probability</i>	0.046	0.020	0.010	0.095	0.045	0.023

The *t-test* is performed [11]. The results of a paired *t-test* performed are shown in Table 3. The degrees of freedom are 9 in all considered cases. The probability, assuming the null hypothesis, is less than 0.05, proving the significant statistic difference between all considered algorithms from Table 1, test set. The difference between ACS and HAS for test set - fully packed input is not significant.

For test set - fully packed input, the probability, assuming the null hypothesis, is less than 0.05, proving the significant statistic difference between TS-ACS and TS-HAS.

5 Conclusions

Nowadays results in the application of metaheuristic algorithms for providing good solutions to difficult combinatorial optimization problems seem to indicate that combining local optimization with good metaheuristics can give powerful optimization algorithms. In this paper we present an ant colony system coupled with a simple local search applied to the over-constrained airport gate assignment problem. Comparison with tabu search heuristic and ant colony system is showing good results for the hybrid algorithm.

References

1. Babic, O., Teodorovic, D., Tomic, V.: Aircraft stand assignment to minimize walking. *Journal of Transportation Engineering* 110, 55–66 (1984)
2. Braaksma, J., Shortreed, J.: Improving airport gate usage with critical path method. *Transportation Engineering J. of ASCE* 97, 187–203 (1971)

3. Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evol. Comp.* 1, 53–66 (1997)
4. Dorigo, M., Maniezzo, V., Colomi, A.: The ant system: Optimization by a colony of cooperative agents. *IEEE Trans. on Systems, Man, and Cybernetics- Part B* 26, 29–41 (1991)
5. Cheng, Y.: Network-based simulation of aircraft at gates in airport terminals. *J. of Transportation Engineering*, 188–196 (1998)
6. Cheng, Y.: A rule-based reactive model for the simulation of aircraft on airport gates. *Knowledge-based Systems* 10, 225–236 (1998)
7. Ding, H., Lim, A., Rodrigues, B., Zhu, Y.: The over-constrained airport gate assignment problem. *Computers and Oper. Research* 32, 1867–1880 (2005)
8. Gu, Y., Chung, C.A.: Genetic algorithm approach to aircraft gate reassignment problem. *J. of Transportation Engineering* 125(5), 384–389 (1999)
9. Wang, Z., Shang, H., Ning, X.: Assignment algorithm for airport gate and its application. *J. of Nanjing University of Aeronautics and Astronautics* 39(6), 819–823 (2007)
10. Xu, J., Baile, G.: The airport gate assignment problem: Mathematical model and a tabu search algorithm. In: 34th Annual Hawaii Int. Conf. on System Sciences, vol. 3 (2001)
11. Yan, S., Chang, C.-M.: A network model for gate assignment. *J. of Advanced Transportation* 32(2), 176–189 (1998)
12. <http://www.physics.csbsju.edu/stats/t-test.html>

Extracting Multi-knowledge from fMRI Data through Swarm-Based Rough Set Reduction

Hongbo Liu¹, Ajith Abraham^{2,*}, and Hong Ye¹

¹ School of Computer Science, Dalian Maritime University,
Dalian 116026, China

² Centre for Quantifiable Quality of Service in Communication Systems,
Norwegian University of Science and Technology, Trondheim, Norway
lhb@dlut.edu.cn, ajith.abraham@ieee.org

Abstract. With the development of brain science, various kinds of new methods and techniques are applied. A mass of Functional Magnetic Resonance Imaging (fMRI) data is collected ceaselessly, which implicates very important information. The useful information need be extracted and translated to intelligible knowledge, which is exigent to develop some methods to analyze them effectively and objectively. In this paper, we attempt to extract multi-knowledge from fMRI Data using rough set approach. We introduce the data pre-processing workflow and methods. A rough set reduction approach is presented based on particle swarm optimization algorithm, which discover the feature combinations in an efficient way to observe the change of positive region as the particles proceed through the search space. The approach supports multi-knowledge extraction. We also illustrate some results using our approach, which is helpful for cognition research.

Keywords: Rough set, fMRI, Particle swarm optimization, Multi-knowledge.

1 Introduction

Neuroinformatics is a research field that encompasses the development of neuroscience data and application of computational models and analytical tools. These areas of research are important for the integration and analysis of increasingly fine grain experimental data and for improving existing theories about nervous system and brain function [1]. Functional Magnetic Resonance Imaging (fMRI) is one of the most important tools for Neuroinformatics, which combines neuroscience and informatics science and computational science to develop approaches needed to understand human brain. Recently most of the researches mainly arrange the activation features on region of interest (ROI) through statistical analysis, and neuroscientists or psychologists give out some explanation for the experimental results, which depends strongly on on their accumulative experience and subjective tendency. Rough set theory provides a novel approach to reduce the fMRI data and extracting knowledge. It helps us to derive rules from data represented in a given information system. The derivation of rules serves two main purposes: Firstly, the rules may be used in the classification of

database objects, that is, predict the outcomes of unseen objects. Secondly, the rules may be used to develop a model of the domain under study, that is present knowledge in a format that can be understood by a human [2]. The rough set approach consists of several steps leading towards the final goal of generating rules from information/decision systems. The main steps of the rough set approach are: (1) mapping of the information from the original database into the decision system format; (2) completion of data; (3) discretization of data; (4) computation of reducts from data; (5) derivation of rules from reducts; (6) filtering of rules. One of most important task is the data reduction process.

Conventional approaches always try to find a good reduct or to select a set of features [3]. In the knowledge discovery applications, only the good reduct can be applied to represent knowledge, which is called a single body of knowledge. In fact, many information systems in the real world have multiple reducts, and each reduct can be applied to generate a single body of knowledge. Therefore, multi-knowledge based on multiple reducts has the potential to improve knowledge representation and decision accuracy [4]. However, it would be exceedingly time-consuming to find multiple reducts in an instance information system with larger numbers of attributes and instances. In most of strategies, different reducts are obtained by changing the order of condition attributes and calculating the significance of different condition attribute combinations against decision attribute(s). It is a complex multi-restart processing about condition attribute increasing or decreasing in quantity. Swarm-based search approaches are of great benefits in the multiple reduction problems, because different individual trends to be encoded to different reduct. So it is attractive to find multiple reducts in the decision systems.

Particle swarm algorithm is inspired by social behavior patterns of organisms that live and interact within large groups [5]. The swarm intelligent model helps to find optimal regions of complex search spaces through interaction of individuals in a population of particles [6]. The particle swarm algorithm is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal feature subset. Additionally, it can be the case that particles discover the best feature combinations as they proceed throughout the search space. This paper introduces particle swarm optimization algorithm to the rough set reduction processing and its application on fMRI data analysis.

2 Data Pre-processing

A typical normalized image contains more than 500,000 voxels, so it is impossible that feature vector can contain so immense voxels. We transform datasets from MNI template to Talairach coordinate system. Then we can use the region information in Talairach as features to reduce the dimensionality of the images. We used a SPM99 software package¹ and in-house programs for image processing, including corrections for head motion, normalization and global fMRI signal shift [7]. A simplified workflow is illustrated in Figure 1. Feature selection and extraction approach for fMRI data is composed of: (1) Find out the most active voxels in several regions of brain under the *t*-test

¹ <http://www.fil.ion.ucl.ac.uk/spm/>

of basic models in SPM99 and save their coordinates; (2) Scan fMRI image and search the voxels according to the coordinates saved; (3) Respectively average all voxels in the spherical region whose center is corresponding saved voxel and whose radius is a pre-defined constant. These results of a single image are formed one feature vector; (4) If the image is not the last one, go to Step 2, otherwise, end. The user interface for feature selection & extraction is illustrated in Figure 2.

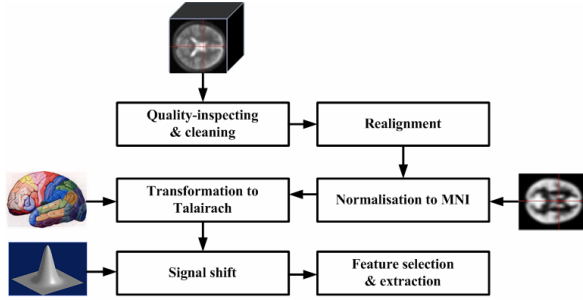


Fig. 1. Pre-processing workflow for fMRI data

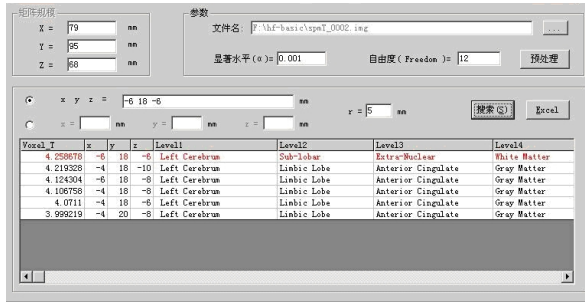


Fig. 2. Location for feature selection & extraction

3 Rough Set Reduction Algorithm

3.1 Reduction Criteria

The basic concepts of rough set theory and its philosophy are presented in [3], [8-10]. Here, we explain only the denotements relevant to our reduction method.

In rough set theory, an information system is denoted in 4-tuple by $S = (U, A, V, f)$, where U is the universe of discourse, a non-empty finite set of U objects $\{x_1, x_2, \dots, x_{|U|}\}$. A is a non-empty finite set of attributes such that $a : U \rightarrow V_a$ for every $a \in A$, where V_a is the value set of the attribute a and $V = \cup_{a \in A} V_a$. $f : U \times A \rightarrow V$ is the total decision function (also called the information function) such that $f(x, a) \in V_a$ for every $a \in A, x \in U$. The information system can also be defined as a decision table by $S = (U, C, D, V, f)$. For the decision table, C and D are condition and decision attributes, respectively.

Lower Approximation: Let $R \subseteq C \cup D$, $X \subseteq U$ and $U/R = \{R_1, R_2, \dots, R_i, \dots\}$. The R -lower approximation set of X is the set of all elements of U which can be with certainty classified as elements of X , assuming knowledge R . It can be presented formally as

$$APR_R^-(X) = \bigcup \{ R_i \mid R_i \in U/R, R_i \subseteq X \} \tag{1}$$

Positive Region: Let $B \subseteq C$, $U/D = \{D_1, D_2, \dots, D_i, \dots\}$. The B -positive region of D is the set of all objects from the universe U which can be classified with certainty to classes of U/D employing features from B , i.e.,

$$POS_B(D) = \bigcup_{D_i \in U/D} APR_B^-(D_i) \tag{2}$$

Reduct: The attribute $a \in B \subseteq C$ is D -dispensable in B , if $POS_B(D) = POS_{B-a}(D)$; otherwise the attribute a is D -indispensable in B . If all attributes $a \in B$ are D -indispensable in B , then B will be called D -independent. A subset of attributes $B \subseteq C$ is a D -reduct of C , iff $POS_B(D) = POS_C(D)$ and B is D -independent. Usually, there are many reducts in an instance information system. Let $2^{|A|}$ represent all possible attribute subsets $\{ \{a_1\}, \dots, \{a_{|A|}\}, \{a_1, a_2\}, \dots, \{a_1, \dots, a_{|A|}\} \}$. Let RED represent the set of reducts, i.e., $RED = \{ B \mid POS_B(D) = POS_C(D), POS_{(B-a)}(D) < POS_B(D) \}$.

Multi-knowledge: Let φ is a mapping from the condition space to the decision space. Then multi-knowledge can be defined as $\Psi = \{ \varphi_B \mid B \in RED \}$.

Reduced Positive Universe: Let $U/C = \{ [u_1]_C, [u_2]_C, \dots, [u_m]_C \}$. Reduced Positive Universe U' can be written as:

$$U' = \{ u'_1, u'_2, \dots, u'_m \} \tag{3}$$

and

$$POS'_c(D) = [u'_1]_c \cup [u'_2]_c \cup \dots \cup [u'_i]_c. \tag{4}$$

Where $\forall u'_s \in U'$ and $[u'_s]_{C/D} = 1 (s = 1, 2, \dots, t)$.

Reduced Positive Region: Reduced positive universe can be written as:

$$U'_{pos} = \{ u'_1, u'_2, \dots, u'_i \} \tag{5}$$

and $\forall B \subseteq C$, reduced positive region

$$POS'_B(D) = \bigcup_{X \in U' \wedge X \subseteq U'_{pos} \wedge |X/D|=1} X \tag{6}$$

where $|X/D|$ represents the cardinality of the set X/D . $\forall B \subseteq C$, $POS_B(D) = POS_C(D)$ if $POS'_B(D) = U'_{pos}$ [10]. It is to be noted that U' is the reduced universe, which usually would reduce significantly the scale of datasets. It provides a more efficient method to observe the change of positive region when we search the reducts. We only calculate U/C , U' , U'_{pos} , POS'_B and then compare POS'_B with U'_{pos} .

3.2 Particle Swarm Approach for Reduction

The particle swarm model consists of a swarm of particles, which are initialized with a population of random candidate solutions. They move iteratively through the d -dimension problem space to search the new solutions, where the fitness f can be measured by calculating the number of condition attributes in the potential reduction solution. Each particle has a position represented by a position-vector p_i (i is the index of the particle), and a velocity represented by a velocity-vector v_i . Each particle remembers its own best position so far in a vector $p_i^\#$, and its j -th dimensional value is $p_{ij}^\#$. The best position-vector among the swarm so far is then stored in a vector p^* , and its j -th dimensional value is p_j^* . The particle moves in a state space restricted to zero and one on each dimension. At each time step, each particle updates its velocity and moves to a new position according to Eqs.(7) and (8):

$$v_{ij}(t) = wv_{ij}(t-1) + \phi_1 r_1 (p_{ij}^\#(t-1) - p_{ij}(t-1)) + \phi_2 r_2 (p_j^*(t-1) - p_{ij}(t-1)) \quad (7)$$

$$p_{ij}(t) = \begin{cases} 1 & \text{if } \rho < \text{sig}(v_{ij}(t)); \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Where ϕ_1 and ϕ_2 are positive constants, r_1 and r_2 are the random numbers in the interval $[0,1]$. w is called as the inertia factor, ρ is random number in the closed interval $[0,1]$.

Given a decision table $T = (U, C, D, V, f)$, the set of condition attributes, C , consist of m attributes. We set up a search space of m dimension for the reduction problem. Accordingly each particle's position is represented as a binary bit string of length m . Each dimension of the particle's position maps one condition attribute. The domain for each dimension is limited to 0 or 1. The value '1' means the corresponding attribute is selected while '0' not selected. During the search procedure, each individual is evaluated using the fitness. According to the definition of rough set reduct, the reduction solution must ensure the decision ability is the same as the primary decision table and the number of attributes in the feasible solution is kept as low as possible. In our algorithm, we first evaluate whether the potential reduction solution satisfies $POS'_E = U'_{pos}$ or not (E is the subset of attributes represented by the potential reduction solution). If it is a feasible solution, we calculate the number of '1' in it. The solution with the lowest number of '1' would be selected. The pseudo-code for the particle swarm search method is illustrated in Algorithm 1.

Algorithm 1. A Rough Set Reduct Algorithm Based on Particle Swarm

```

begin
  Calculate  $U'$ ,  $U'_{pos}$  using Eqs.(3) and (5).
  Initialize the particle swarm, and other parameters.
  While (the end criterion is not met) do
     $t = t + 1$ 
    Calculate the fitness value of each particle,
      if  $POS'_E \neq U'_{pos}$ , the fitness is punished as the
        total number of the condition attributes,
      else the fitness is the number of 1.

```



```


$$p^* = \arg \min_{i=1}^n (f(p^*(t-1)), f(p_2(t)), \dots, f(p_n(t)));$$

For  $i = 1$  to  $n$ ;
    
$$p_i^\#(t) = \arg \min_{i=1}^n (f(p_i^\#(t-1)), f(p_i(t)));$$

    For  $j = 1$  to  $d$ 
        Update the  $j$ -th dimension value of  $p_i$  and  $v_i$ 
        according to Eqs.(7) and (8);
    Next  $j$ 
Next  $i$ 
End While.
End.

```

3.3 Experiments Using Benchmarks

To illustrate the effectiveness and performance of the considered algorithms, we illustrate the rough set reduct process and results through two benchmark problems. In this experiment, Genetic algorithm (GA) was used to compare the performance with PSO. In GA, the probability of crossover is set to 0.8 and the probability of mutation is set to 0.08. In PSO, ϕ_1 and ϕ_2 both are 1.49, and the inertia weight w is decreasing linearly from 0.9 to 0.1. The size of the population in GA and the swarm size in PSO both are set to $(\text{even})(10 + 2 * \text{sqrt}(D))$, where D is the dimension of the position, i.e., the number of condition attributes. In each trial, the maximum number of iterations is $(\text{int})(0.1 * \text{recnum} + 10 * (\text{nfields} - 1))$, where recnum is the number of records/rows and nfields is the number of condition attributes. Each experiment (for each algorithm) was repeated 3 times with different random seeds. We consider two benchmark datasets, such as lung-cancer and lymphography from AFS².

Figure 3 illustrates the performance of the algorithms for lung-cancer and lymphography datasets, respectively. For lung-cancer dataset, the results (the number of reduced attributes) for 3 GA runs were 10: {1, 3, 9, 12, 33, 41, 44, 47, 54, 56} (The number before the colon is the number of condition attributes, the numbers in brackets are attribute index, which represents a reduction solution). The results of 3

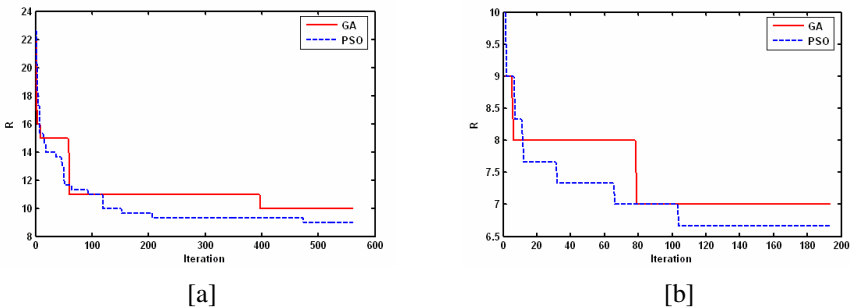


Fig. 3. Performance of rough set reduction for [a] lung-cancer dataset and [b] lymphography dataset

² <http://sra.itc.it/research/afs/>

PSO runs were 9: {3, 8, 9, 12, 15, 35, 47, 54, 55}, 10: {2, 3, 12, 19, 25, 27, 30, 32, 40, 56}, 8: {11, 14, 24, 30, 42, 44, 45, 50}. For lymphography datasets, the results of 3 GA runs all were 7: {2, 6, 10, 13, 14, 17, 18}, the results of 3 PSO runs were 6: {2, 13, 14, 15, 16, 18}, 7: {1, 2, 13, 14, 15, 17, 18}, 7: {2, 10, 12, 13, 14, 15, 18}. PSO usually obtained a better result than GA.

4 Extracting Knowledge from fMRI Data

We analyze the fMRI data from three cognition experiments: Tongue movement experiment, Associating Chinese verb experiment, and Looking at or silent reading Chinese word experiment. They are involved in 9 tasks: 0 - Control task; 1 - Tongue movement; 2 - Associating verb from single noun; 3 - Associating verb from single non-noun; 4 - Making verb before single word; 5 - Looking at number; 6 - Silent reading Number; 7 - Looking at Chinese word; 8 - Silent reading Chinese word. We select 13 ROIs and rank their actives as 0, 1, 2 and 3. The dataset includes 1 record for control task and 2580 records for other 8 cognitive tasks. In other words, the information system consists of 2581 rows and 14 columns. 2581 objects with 13 conditions and 1 decision attributes could be classified to 9 classes. The swarm size in PSO is set to 18. Each experiment (for each algorithm) was repeated 3 times with different random seeds. Other parameter settings are set to as same as ones for the benchmarks in last section. When our algorithm put out the reduction results, we check each particle for the solution. If the solution of the particle satisfies the criterion $POS'_E = U'_{pos}$, we accept the solution. It provided 18×3 opportunities to extract the rules, although the length of these reduction results are not always "best", a minimum length. Some of rules are illustrated as follows:

- Rule1: if M1=2, SMA=2, Broca=2 then Task=1;
- Rule2: if BAs { 7,19,20,40,44,45 } =3, BSC=2 then Task=2;
- Rule3: if BAs { 10,11,13,44,45 } =3, BSC=1 then Task=3;
- Rule4: if BAs { 7,19,40 } =3, BSC=3 then Task=4;
- Rule5: if SMA=2, Broca=3 then Task=6;
- Rule6: if SMA=2, Broca=2, Wernike=3 then Task=8.

5 Conclusions

In this paper, we investigated the rough set approach to extract knowledge from fMRI Data. The data preprocessing workflow and methods for the rough set approach were first discussed. A rough set reduction approach was presented based on particle swarm optimization algorithm, which discover the best feature combinations in an efficient way to observe the change of positive region as the particles proceed through the search space. We also evaluated the performance of particle swarm optimization algorithm with genetic algorithm based on two benchmark datasets. Empirical results indicated that PSO usually required shorter time to obtain better results than GA, although its stability needs to be improved in further research. The swarm-based search approach provides some good benefits in the multiple reduction problems for multi-knowledge extraction. We illustrated some results using our approach for fMRI

data analysis. Although the correctness of the rules need neuroscientists to analyze and verify further, the approach is helpful for cognition research.

Acknowledgments. The first author would like to thank Ye Ji for his scientific collaboration in this research work. This work is supported partly by MOST (2100CCA00700), NSFC (60373095), DLMU (DLMU-ZL-200709).

References

1. Arbib, M.A., Grethe, J.S. (eds.): *Computing the Brain: A Guide to Neuroinformatics*. Academic Press, London (2001)
2. Roed, G.: *Knowledge Extraction from Process Data: A rough Set Approach to Data Mining on Time Series*. Knowledge Extraction from Process data. Thesis, Norwegian University of Science and Technology (1999)
3. Wu, Q., Bell, D.: *Multi-knowledge Extraction and Application*. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) *RSFDGrC 2003*. LNCS (LNAI), vol. 2639, pp. 574–575. Springer, Heidelberg (2003)
4. Wu, Q.: *Multiknowledge and Computing Network Model for Decision Making and Localisation of Robots*. Thesis, University of Ulster (2005)
5. Kennedy, J., Eberhart, R.: *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco (2001)
6. Clerc, M.: *Particle Swarm Optimization*. ISTE Publishing Company, London (2006)
7. Ji, Y., Liu, H., Wang, X., Tang, Y.: *Cognitive States Classification from fMRI Data Using Support Vector Machines*. In: *Proceedings of The Third International Conference on Machine Learning and Cybernetics*, pp. 2920–2923. IEEE Computer Society Press, Los Alamitos (2004)
8. Pawlak, Z.: *Rough Sets and Intelligent Data Analysis*. *Information Sciences* 147, 1–12 (2002)
9. Wang, G.: *Rough Reduction in Algebra View and Information View*. *International Journal of Intelligent Systems* 18, 679–688 (2003)
10. Xu, Z., Liu, Z., Yang, B., Song, W.: *A Quick Attribute Reduction Algorithm with Complexity of $\text{Max}(O(|C||U|), O(|C|^2|U|/C))$* . *Chinese Journal of Computers* 29, 391–399 (2006)

Estimation Using Differential Evolution for Optimal Crop Plan

Millie Pant¹, Radha Thangaraj¹, Deepti Rani², Ajith Abraham³,
and Dinesh Kumar Srivastava⁴

¹ Department of Paper Technology, IIT Roorkee, India

² University of Évora, Núcleo da Mitra, 7002-054, Évora 7000 Évora, Portugal

³ Q2S, Norwegian University of Science and Technology, Norway

⁴ Department of Hydrology, IIT Roorkee, India

millifpt@iitr.ernet.in, t.radha@ieee.org,

deeptinatyan@yahoo.com, ajith.abraham@ieee.org,

dkshyfhy@iitr.ernet.in

Abstract. This paper presents an application of Differential Evolution (DE) to determine optimal crop plan for command area of Pamba-Achankovil-Vaippar (PAV) link project, so as to maximize the net irrigation benefit. The mathematical model of the problem is linear in nature subject to various constraints due to availability of total land area, water, fertilizers, seeds and manure, etc. Numerical results show that DE gives a better performance in comparison to the usual software tools used for solving such problems.

Keywords: Differential Evolution, Reservoir, Crop Plan, Optimization.

1 Introduction

DE has emerged as one of the most promising contemporary optimization technique in the past few years. Some of the reasons for the popularity of DE include easy implementation, little parameter tuning and fast convergence. It has been successfully applied to solve a wide range of optimization problems such as clustering [1], unsupervised image classification [2], digital filter design [3], optimization of non-linear functions [4], chemical engineering processes [5] and multi-objective optimization [6] etc. In the present study we discuss the performance of DE to obtain optimal crop plans under adequate, normal and limited irrigation water availability for irrigation area under the PAV link project. A linear programming based optimization model is used for this.

The proposed Pamba-Achankovil-Vaippar Link project has three storage reservoirs, two tunnels, necessary canal system and a few power generating units [7]. The Punnameedu reservoir (reservoir-2) is located at a higher elevation on river Pamba Kal Ar in the Pamba basin of Karala state in India, which serves a part/full of its downstream mandatory requirements and supplies surplus water to reservoir-1 by intra-basin export of surplus water (diversion) through tunnel-2. The Achankovil Kal Ar reservoir (reservoir-1) located on Achankovil Kal Ar river in Achankovil river

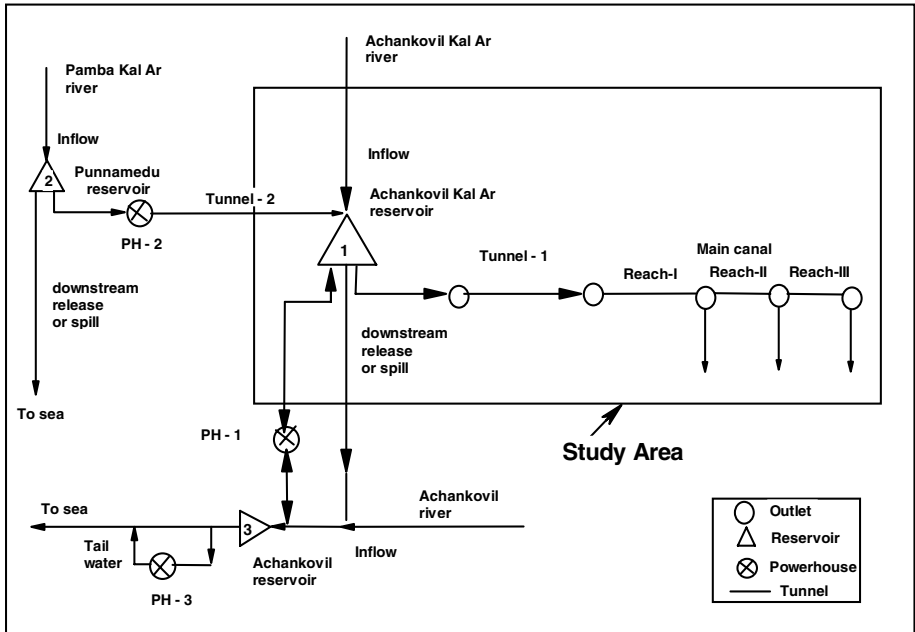


Fig. 1. Schematic Diagram for PAV Link Diversion System

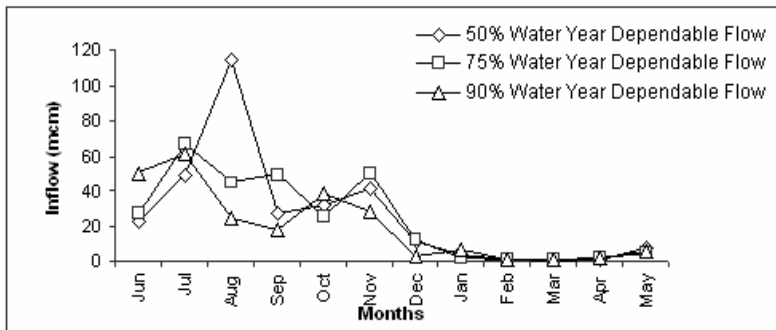


Fig. 2. Monthly inflows at Reservoir-I

basin of Kerala state, supplies water for irrigation purposes to the state of Tamilnadu, through tunnel-1 to the main canal. The water from main canal is then distributed to the command area of Vaippar basin in Tamilnadu state. The reservoir is proposed as a within-the-year storage scheme. Figure 1 shows a schematic representation of the PAV link diversion system. Besides this, reservoir-1 releases water for power generation. The Achankovil reservoir (reservoir-3), which is located on Achankovil river in the Achankovil river basin of Kerala state, besides acting as a pumped storage scheme accommodating the water drawn from the upstream reservoir-1, also serves the purpose of releasing water to downstream to meet its downstream mandatory demands.

Also, if there is deficit at reservoir-1 the surplus water of reservoir-3 can be pumped back to reservoir-1. The monthly inflow at reservoir-1 for the 50 percent, 75 percent and 90 percent water year dependable flows are shown in Figure 2.

The GCA (gross command area) potential and CCA (culturable command area) potential of the project would be 145,573 and 101,555 ha, respectively. The proposal is to irrigate 91,400 ha (CCA actually considered) of area per annum with an irrigation intensity of 90 percent. The proposed cropping pattern was formulated by the Tamilnadu State Agriculture Department exclusively for this project. Crop areas given in the report were determined on the basis of the food requirements of the population likely to be benefited from project. The suggested cropping pattern consists of 8 crops namely; Paddy, Oilseed, Jowar, Vegetables (Brinjal, Ladyfinger and Beans), Pulses, Bajra, Cotton and Chillies; the proposed corresponding area allocation for each crop is 15234, 7109, 12187, 15233, 6093, 15233, 12187 and 8124 ha, respectively, and crop yields from these crops under irrigation area; 5.39, 1.51, 2.56, 3.0, 0.741, 2.56, 1.66 and 1.51 metric tones (M.T) per unit cropped area, respectively. The gross irrigation requirement for the command area as per project is 635 mcm including transmission losses.

The total production from proposed irrigation, cost of produce and expenses on cultivation of various crops under irrigation conditions, i.e., total cost on seeds, fertilizers, manure, irrigation charges and labour per unit area of each crop is available from project report [8]. The water release from reservoir-1 for irrigation is obtained through joint operation of the reservoir system with the help of successive approximation dynamic programming model [9]. The water available at the main canal is further distributed among the areas under Reaches-I, II and III. In this study, however, optimal cropping patterns are presented for the total CCA for water available during 50 percent, 75 percent and 90 percent water year dependable flows. The remaining of the paper is organized as follows; in section 2, we briefly describe the working of DE. In section 3, we discuss the mathematical model used in the present study. Section 4 deals with experimental settings. In section 5, we give numerical results and the paper finally concludes with section 6.

2 Differential Evolution

Differential Evolution was proposed by Storn and Price [10]. It is a population based algorithm like genetic algorithms using the similar operators; crossover, mutation and selection. The main difference in constructing better solutions is that genetic algorithms rely on crossover while DE relies on mutation operator [11]. DE works as follows: First, all individuals are initialized with uniformly distributed random numbers and evaluated using the fitness function provided. Then the following will be executed until maximum number of generation has been reached. For a D-dimensional search space, each target vector $x_{i,g}$, a mutant vector is generated by

$$v_{i,g+1} = x_{r_1,g} + F * (x_{r_2,g} - x_{r_3,g}) \tag{1}$$

where $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$ are randomly chosen integers, must be different from each other and also different from the running index i . $F (>0)$ is a scaling factor which

controls the amplification of the differential evolution $(x_{r_2,g} - x_{r_3,g})$. In order to increase the diversity of the perturbed parameter vectors, crossover is introduced [12]. The parent vector is mixed with the mutated vector to produce a trial vector $u_{ji,g+1}$,

$$u_{ji,g+1} = \begin{cases} v_{ji,g+1} & \text{if } (rand_j \leq CR) \text{ or } (j = j_{rand}) \\ x_{ji,g} & \text{if } (rand_j > CR) \text{ and } (j \neq j_{rand}) \end{cases} \quad (2)$$

where $j = 1, 2, \dots, D$; $rand_j \in [0,1]$; CR is the crossover constant takes values in the range $[0, 1]$ and $j_{rand} \in (1,2,\dots,D)$ is the randomly chosen index.

Selection is the step to choose the vector between the target vector and the trial vector with the aim of creating an individual for the next generation.

3 Mathematical Model

A linear programming based optimization model is used for crop planning. The model maximizes net returns from crops and yields optimal crop plan and monthly releases required from reservoir-1. Surface water, land availability, fertilizers (N, P, and K), seeds and manure requirements are considered as constraints in the model. For the purpose of modeling, the crops have been segregated as food grains, cash crops and others. Paddy, Jowar and Bajra are clubbed together as these falls under the category of food grains.

Crop Planning Model

Objective function:

$$\text{Max } Z = G_T - P_T \quad (3)$$

$$G_T = \sum_{i=1}^N y_i b_i A_i \quad \text{and} \quad P_T = \sum_{i=1}^N (CS_i + CM_i + CF_i + CI_i + CLH_i) A_i$$

In the above equation Z = annual net return from irrigated agriculture, G_T = total annual gross returns from crops, P_T = total annual net expenses on cultivating crops, N = total number of crops, $i = 1, 2, \dots, 8$ for Paddy, Oilseeds, Jowar, Vegetables, Pulses, Bajra, Cotton and Chillies, respectively, A_i = area under i th crop, CS_i = expenses on seeds for i th crop per unit area, CM_i = expenses on manure for i th crop per unit area, CF_i = expenses on fertilizers for i th crop per unit area, CLH_i = expenses on labor and machinery for i th crop per unit area, CI_i = expenses on irrigation water charges for i th crop per unit area, y_i = crop yield in weight units from i th crop per unit area, and b_i = value of crop produce from i th crop per unit yield.

Constraints

(i) Surface Water Availability Constraint

The volumes of water releases should be less than or equal to the surface water available, i.e.,

$$\sum_{i=1}^N (W_{i,t}) A_i \leq R_t \quad \text{for all } t \quad (4)$$

where $W_{i,t}$ = gross irrigation requirement of i th crop during time period t in terms of depth, A_i = area under i th crop and R_t = irrigation water released from reservoir in time period t .

(ii) Land Availability Constraints:

The net area allocated to the given crops should be less than or equal to the total area available, i.e.,

$$\sum_{i=1}^N (\lambda_{i,t}) A_i \leq A_T, \quad \text{for all } t \quad \text{and} \quad \sum_{i=1}^N A_i \leq A_T \quad (5)$$

where $\lambda_{i,t}$ = land use coefficient of the i th crop during time period t and A_T = total area under irrigation per annum.

(iii) Yield Requirement Constraint:

Yield produced from the crops should be greater than or equal to the proposed yield requirement, i.e.,

$$\sum_i y_i A_i \geq \sum_i y_T^i \quad \text{for } i=1,3 \text{ and } 6 \quad \text{and} \quad y_i A_i \geq y_T^i \quad \text{for all } i \quad (6)$$

where y_T^i = total yield required from i^{th} crop.

(iv) Fertilizers Availability Constraints:

Quantity of fertilizer of type f required will be less than or equal to the quantity of fertilizer type f available, i.e.,

$$\sum_{i=1}^N F_{f,i} A_i \leq F_{f,T} \quad \text{for all } f \quad (7)$$

where $F_{f,i}$ = quantity of fertilizer type f required per unit area for i th crop and $F_{f,T}$ = total available quantity of fertilizer of type f . Three types of fertilizers have been considered in the application of model, i.e., Nitrogen, Phosphorus and Potassium (N, P, K).

(v) Manure Availability Constraint:

Total quantity of manure required will be less than or equal to the total quantity of manure available, i.e.,

$$\sum_{i=1}^N M_i A_i \leq M_T \quad (8)$$

where M_i = quantity of manure required per unit area for i th crop and M_T = total available quantity of manure.

(vi) Seeds Availability Constraint:

$$S_i A_i \leq S_T \quad \text{for all } i \quad (9)$$

where S_i = quantity of seeds required per unit area for i th crop and $S_{i,T}$ = total available quantity of seeds for i th crop.

(vii) Bounds on Areas under Various Crops:

$$A_{i,\min} \leq A_i \leq A_{i,\max} \quad (10)$$

where $A_{i,\min}$ = lower limit on the area under i th crop and $A_{i,\max}$ = upper limit on the area under i th crop.

A minimum crop area constraint has been specified on each crop so as to see that area occupied by crops does not fall below area under rain-fed cultivation. It has also been specified that area proposed under cotton and chillies should not be more than 18 percent and 17 percent of annual irrigation. This condition is justified because their yields have high revenues and optimally higher area allocation to these crops may cause reduction in food grain output, which is socially undesirable. It has been considered essential that total food grain production should not be less than 101, 995 M.T.

4 Experimental Settings

In this section we give the data used for the mathematical model used in section 3 and the parameter settings for DE. From information available [13] estimates of average values of quantities/ha required for each crop for resource inputs, i.e., seeds, manure and fertilizers are obtained. Total requirements of these resources are obtained from these values and crop area allocation as per project report. Initially it was assumed that total quantity available for each resource is equal to total quantity required for the resource. The crop planning model is solved using LINGO package. The first trial run is made of the model assuming that the amount of each resource available is equal to the required amount, and from results it was seen that out of the total CCA, i.e., 91400 ha only 88818.64 ha is allocated to the crops, i.e., with this trial the total CCA was not allocated to various crops (Table 1). Further model runs were made by varying quantity of resource availability in some percent of required amount the area allocations for these trials are given in Table 1.

Finally it was assumed that 120 percent of the total quantity initially estimated for each resource may be considered as the extent of quantity available as input, for which almost all the area proposed has been allocated (Table 2).

Parameter settings for DE

As discussed in the earlier section, DE requires very few parameter settings in comparison to other Evolutionary algorithms. Only three parameters; Population size, crossover constant and scaling parameter are needed for DE.

Population size: 30; crossover constant CR= 0.5; Scaling parameter F = 0.5.
Maximum number of generation: 1000; Number of runs: 30;

Table 1. Optimal area allocations with variable resource inputs available

Resource Inputs	Optimal Area Allocations (ha)
80%	73120.00
90%	74384.31
100%	88818.64
110%	89754.08
120%	91399.99
130%	91400.00

Table 2. Extents of resource available

Resource		Extent Availability ⁺
Fertilizers (kg)	N	5774510
	P	4911240
	K	2303256
Manure (M.T.)		1431135
Seeds (kg)	Paddy	1188252
	Oil seeds	2102154
	Jowar	93838.8
	Vegetables	178403.04
	Pulses	329049
	Bajra	255906
	Cotton	319893
Chillies	329049	

Table 3. Performance Comparison of DE with LINGO for 50%, 75% and 90% water year dependable flow

ITEM	50% water year dependable flow		75% water year dependable flow		90% water year dependable flow	
	DE	LINGO	DE	LINGO	DE	LINGO
A1	2818.71	2818.639	0.00161	0	0.00019	0
A2	15233	15233	15233	15233	13125.4	13125.35
A3	7109	7109	5034.2	5033.97	0.00044	0
A4	8124	8124	8124	8124	8124	8124
A5	12186	12186.99	12186	12186.99	0.00023	0
A6	4433.27	4433.273	5193.36	5193.36	927.386	927.3857
A7	15233	15233	16000	15876.04	16000	15876.04
A8	14273.1	14273.09	14624.4	14592.21	14624.4	14592.21
Objective value	16513.1	16513.1	16503.3	16449.01	15319.1	15264.74

5 Numerical Results

The performance of DE is compared with LINGO, which is another popular software tool for solving Linear Programming Problems. From the numerical results given in Table 3, it can be seen that for 50% water year dependable flow, both DE and LINGO gave exact results. However for 75% and 90% water year dependable flow performance of DE is superior to LINGO in terms of net profit. For 75% water year

dependable flow, the net profit as obtained by DE is 16503.3 and by LINGO is 16449.01, which shows an improvement of 0.33 %. For 90 % water year dependable flow DE gave a profit of Rs 15319.1, whereas profit obtained by LINGO is Rs 15264.7, which shows an improvement of 0.36 %. Also we get to know the optimal area allocation for different crops while using DE and LINGO. The time taken by DE and LINGO are same for all the three cases.

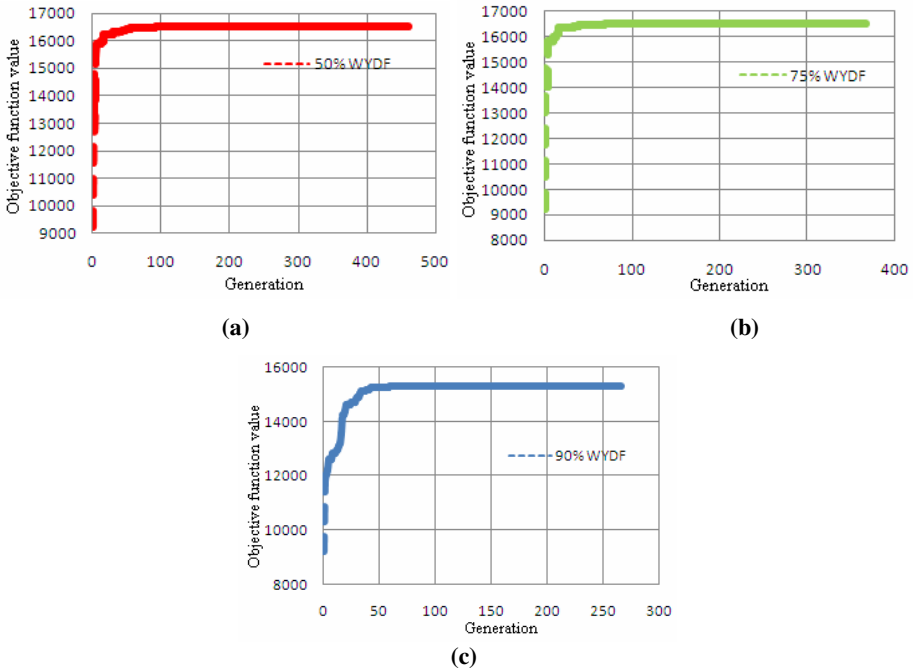


Fig. 3. (a), (b), (c) represent convergence graphs for 50%, 75% and 90% water year dependable flow

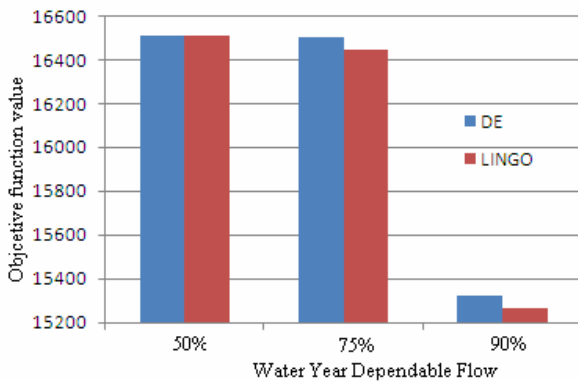


Fig. 4. Objective function value obtained by DE and LINGO

6 Conclusion

In the present study an application of DE is shown for determining the optimal cropping pattern so as to maximize the net profit under the given conditions. Its performance is compared vis-à-vis LINGO, which is a very popular tool for solving Linear Programming Problems. Numerical results show that DE gives a superior performance in comparison to LINGO for 75% and 90% water year dependable flow. Thus DE may be considered as an alternative for solving such type of problems. In future we shall be comparing the performance of DE with other techniques like PSO for solving similar types of problems.

References

1. Paterlini, S., Krink, T.: High performance clustering with differential evolution. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, pp. 2004–2011 (2004)
2. Omran, M., Engelbrecht, A., Salman, A.: Differential evolution methods for unsupervised image classification. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, pp. 966–973 (2005a)
3. Storn, R.: Differential evolution design for an IIR-filter with requirements for magnitude and group delay. Technical Report TR-95-026, International Computer Science Institute, Berkeley, CA (1995)
4. Babu, B., Angira, R.: Optimization of non-linear functions using evolutionary computation. In: Proceedings of the 12th ISME International Conference on Mechanical Engineering, India, pp. 153–157 (2001)
5. Angira, R., Babu, B.: Evolutionary computation for global optimization of non-linear chemical engineering processes. In: Proceedings of International Symposium on Process Systems Engineering and Control, Mumbai, pp. 87–91 (2003)
6. Abbass, H.: A memetic pareto evolutionary approach to artificial neural networks. In: Stumptner, M., Corbett, D.R., Brooks, M. (eds.) Canadian AI 2001. LNCS (LNAI), vol. 2256, pp. 1–12. Springer, Heidelberg (2001a)
7. Ramakrishnan, M.: Systems Analysis of Multireservoirs. M.E. Dissertation, Dept. of Hydrology, Univ. of Roorkee, Roorkee (1999)
8. Feasibility Report of Pamba-Achankovil-Vaippar Link Project, NWDA, Ministry of Water Resources, New Delhi, India (1995)
9. Rani, D.: Multilevel Optimization of a Water Resources System, Ph.D. Thesis, Dept. of Mathematics, Indian Inst. of Tech., Roorkee, India (2004)
10. Storn, R., Price, K.: Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report, International Computer Science Institute, Berkeley (1995)
11. Karaboga, D., Okdem, S.: A simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm. Turk J. Elec. Engin. 12(1), 53–60 (2004)
12. Storn, R., Price, K.: Differential Evolution – a simple and efficient Heuristic for global optimization over continuous spaces. Journal Global Optimization 11, 341–359 (1997)
13. Handbook on Agriculture, ICAR, Ministry of Agriculture, New Delhi, India (1997)

Hybrid Metaheuristics for Global Optimization: A Comparative Study

Antoniya Georgieva¹ and Ivan Jordanov²

¹ Nuffield Department of Obstetrics and Gynaecology, University of Oxford,
Oxford OX3 9DU, UK

² School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK
Antoniya.Georgieva@obs-gyn.ox.ac.uk, Ivan.Jordanov@port.ac.uk

Abstract. Several metaheuristic approaches for global optimization (GO) are investigated and their performances compared in this paper. We critically review and analyze recently proposed Stochastic Genetic Algorithm (StGA) for GO and compare it with our GO hybrid metaheuristic called *Genetic LP _{τ}* and *Simplex Search (GLP _{τ}* S), which combines the effectiveness of Genetic Algorithms during the early stages of the search with the advantages of Low-Discrepancy sequences and the local improvement abilities of Simplex search. For comparison purposes we also use Fast Evolutionary Programming (FEP) and Differential Evolution (DE) methods. In parallel to our method, FEP and DE, we investigate further, re-run and test the StGA implementation on a number of multimodal mathematical functions. The obtained StGA results demonstrate inferior performance (compared with our *GLP _{τ}* S and DE methods), producing much worse than the reported in [1] results (with the only exception for the two-dimensional cases). We argue that the published in [1] accuracy and convergence speed results (given as number of function evaluations) are incorrect for most of the testing functions and investigate why the method is failing in those cases.

Keywords: Hybrid methods, Metaheuristics, Evolutionary computation, Global optimization.

1 Introduction

The GO techniques could be broadly classified into three groups – deterministic, heuristic (or stochastic, including metaheuristic approaches), and hybrid methods. The objective functions of the deterministic methods are usually subjected to certain assumptions and involve calculation of the derivatives or *Lipschitz* constants, which provide analytical rules and mechanisms for reaching a global optimum in a finite number of steps. The methods from the second group use heuristic and stochastic rules to guide the search, with meta-heuristic ones having even *adaptive* rules that tailor the search to the particular landscape of the objective function. The algorithms from these two groups have specific advantages and drawbacks and in order to improve their performances, authors often hybridize them, in search for faster and more

robust techniques [1, 2, 3]. In this study, we critically analyse and compare performances of several GO approaches with our hybrid GO method that uses novel metaheuristic rules and a local search [4].

2 GO Heuristic Methods Used in the Comparative Study

In this section we briefly introduce our hybrid metaheuristic method, called *Genetic LP_τ and Nelder-Mead Simplex Search (GLP_τS)* [4], which combines the effectiveness of GA during the early stages of the search with the advantages of the *LP_τO* method [2], and the local improvement abilities of Simplex search. In [2] we reported promising results for the *LP_τO* global optimisation technique over a set of multimodal test functions of moderate dimensions (up to 20). However, in order to be able to handle efficiently problems of higher than 20 dimensions, we improved the hybrid method by incorporating evolutionary strategy and genetic operators for finding promising regions in the early exploration of the searched hyperspace. Also, our initial population points are not random (as in a conventional GA), but uniformly distributed *LP_τ* points (detailed description of the method can be found in [4]).

Generally, the technique could be described as follows:

- a) Generate a number I of initial *LP_τ* points;
- b) Select G points, ($G \in I$), that correspond to the best function values. Let this be the initial population $p(G)$ of the GA;
- c) Perform GA until a halting condition is satisfied;
- d) From the population $p(G)$ of the last GA generation, choose N points ($1 \leq N \leq G/2$);
- e) Initialize the *LP_τO* search in the neighbourhood of each chosen point;
- f) After the stopping conditions of the *LP_τO* searches are satisfied, initialize a local Simplex Search in the best point found by all *LP_τO* searches.

In this work we also use Fast Evolutionary Programming (FEP), [3], for a comparison, and implement and report results for Differential Evolution (DE) method. This is a well known technique and four different implementations of DE are applied for optimisation of a number of 2 to 30 dimensional test functions in [5].

The other metaheuristic approach, which is the main subject of the critical review and investigation of this work, is Stochastic Genetic Algorithm (StGA). The novelty claimed by the authors in [1] is the employed *stochastic coding strategy*. However, this approach is very similar to a class of Genetic Algorithms that use local improvement procedures as part of the evaluation of individuals [6, 7], also known as *Memetic Algorithms* [8], which, in general need more function evaluations to converge to a global solution. The reported in [1] optimization results from tests on mathematical functions were considerably superior and so dominant (reducing the computational effort sometimes more than 100 times, e.g., compared with FEP, in the case of function f_8 , Table 1), that we decided to further investigate this method, implement it and execute all tests again (we also used Matlab for this purpose).

The proposed in [1] StGA method can be described briefly as follows:

- a) Initialize the population with size NP ;
- b) Perform *local selection* for the current population;
- c) Perform *global selection* for the current population;
- d) Perform *recombination* with the selected individuals;
- e) Perform *mutation* with probability P_m and compute the fitness function for the new individuals;
- f) If the predefined number of iterations is not reached, repeat steps 2 to 6 for the new population.

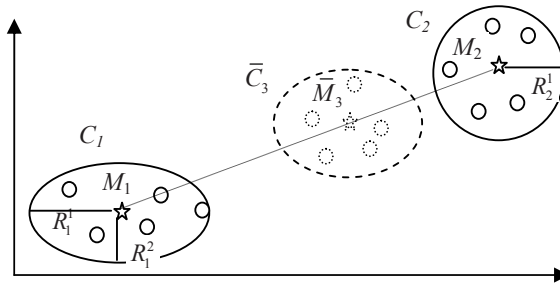


Fig. 1. Illustration of StGA crossover. C_3 is the child chromosome, result of crossover of the parents C_1 and C_2 . The chromosomes represent stochastic regions around the mean points M_i (each determined by five randomly chosen asexual children).

The novelty of StGA is the *Stochastic Coding Mechanism* – individuals are not coded as points with n coordinates in the space, but as regions with mean and variance in each direction of space. For example, one chromosome is given as

$C_i = [(M_{i1}, V_{i1}), (M_{i2}, V_{i2}), \dots, (M_{in}, V_{in})]$, where $i = 1, \dots, NP$, n is the problem dimensionality, M_{ij} is the mean value in each direction and V_{ij} – the corresponding variance. This coding plays a role only during the *local selection* stage. All other stages of StGA are trivial, in terms of being used in most evolutionary algorithms.

One specific characteristic of StGA is that the authors use chromosome binary representation with flexible number of bits for each test function.

3 Discussion and Critical Comparison

For demonstrative purposes, let us consider Fig. 1, where two chromosomes from a parent generation are denoted with C_1 and C_2 .

During the *local selection*, five points ($N = 5$, *asexual children*) are generated randomly in *stochastic regions* around points M_1 and M_2 . The size of the regions corresponds individually to each chromosome C_i and is different for all n dimensions ($n=2$ in Fig. 1). If any of the *asexually* produced children has better fitness value than the corresponding mean M_i , it takes the place of M_i (and the stochastic region shrinks *a bit*), otherwise, the stochastic region expands *a bit*. After the local selection phase, there is a global tournament selection, where the chromosomes are selected for the *mating pool*.

Let C_1 and C_2 be two such points expected to undergo crossover to produce a new chromosome \bar{C}_3 . As described in [1], and further clarified in our correspondence with the authors, the crossover in the mating pool has the following steps:

- a) The mean points M_1 and M_2 (that correspond to the chromosomes C_1 and C_2) are coded as *integer* binary strings;
- b) After one-point crossover of M_1 and M_2 and taking the average of the corresponding radii of the stochastic regions, a new chromosome \bar{C}_3 is produced with a mean \bar{M}_3 and stochastic region with $R_3^1 = (R_1^1 + R_2^1)/2$ and $R_3^2 = (R_1^2 + R_2^2)/2$.

Although M_1 and M_2 are points with real coordinates, when coded, their coordinates are transformed into integers. The authors claimed that their coding mechanism allows “a somewhat coarse division of the variable space without compromising the accuracy of the final solution”, because “those points that are not covered by the binary string could also be approached by StGA through numerical sampling of the stochastic regions” (provided by the local selection phase). However, we argue that such an *integer* binary coding can lead to misinterpreting the actual search and can produce misleading results, mainly due to:

- a) Let the chromosomes C_1 and C_2 , that were chosen for the *mating pool* during the tournament selection, have fitness values $F_1=F(M_1)$ and $F_2=F(M_2)$ respectively. After coding M_1 and M_2 as *integer* binary strings, two new points M_1^* and M_2^* (with integer coordinates) that most likely have two different fitness values – F_1^* and F_2^* will be obtained. These values will not necessarily be as *good* as the values of F_1 and F_2 . Or, at least it can be said that the fitness values used in the selection procedure, do not correspond to the actual coded individuals. Therefore, the new solution \bar{M}_3 (obtained by one-point crossover of M_1^* and M_2^*) is most likely not to correspond and not to lead to an improved fitness value, which in turn puts question mark about the method convergence to a global minimum.
- b) It is obvious that \bar{M}_3 will have coordinates with integer values (generated as an integer binary string). In the next generation, during the local selection stage, $N = 5$ points will be generated in the stochastic region of point \bar{M}_3 . However, even if a better point \hat{M}_3 , ($F(\hat{M}_3) < F(\bar{M}_3)$) is encountered during this stage, and is selected to be a *representative* of the chromosome \bar{C}_3 , when chosen for the *mating pool*, its coordinates will be transformed into integers as well. Therefore, it is likely (especially if the stochastic region is *small*) that the new point \hat{M}_3 will coincide with \bar{M}_3 after the coding. As stated by the authors in our correspondence: “the integer determined by each binary string corresponds to a real number in the physical space of the variable being represented”. If we consider the cases of functions f_7 , f_9 and f_{20} , where 8-bit binary representation is used, the

number of represented values (chromosomes) will be $2(2^7-1) + 1$ (assuming one bit is used for the sign). It can be seen that not too many numbers (chromosomes) can be presented in this way and that the numerical grid is rough enough to miss vital optimization points in the searched space.

Considering the last proposition, one can conclude that the size of a stochastic region plays very important role during the local selection (the size is given by the radii $R_i^j, j = 1, \dots, n$ and $i = 1, \dots, K$, where K is the population size). Accordingly to [1], radii R_i^j take random values in the region

$$R_i^V = (1/120 \sim 1/80) (B_{ui} - B_{li}), \tag{2}$$

and can decrease or increase during the optimization with an adaptation step $\delta_i = (2/100 \sim 5/100) V_i^R$, where V_i^R takes random value within R_i^V .

To demonstrate our second proposition, let us consider the *Rastrigin function* (f_9), used as a test function in [1]. The global minimum $f_{\min}(0, \dots, 0) = 0$ is in the origin and the function is investigated in the interval $[-5.12, 5.12]$, $n = 30$. From (2) follows that $R_i^V = [0.853, 0.128]$ and the maximal possible value for $V_i^R = 0.128$. Therefore, the maximal possible interval for $\delta_i = (0.00256, 0.0064)$. After k iterations, providing the stochastic region is always *expanding*, the maximal possible value for the radii R_i^j will be $R_i^j = 0.128 + 0.128 * k * 0.0064$. Let us assume $k = 200$ (the authors used 190 iterations in total), then the maximal possible value for any radius of any stochastic region will be $R_i^j = 0.292$. Following our previous discussion, the coordinates of all *asexually produced* children, when coded in a region with this radius will coincide with the closest integer. In the case of *Rastrigin function*, we argue that these values are ‘very small’ and the authors’ coding makes the local selection practically useless, since all *asexual* children will be too close to M_i .

3.1 Counting the Number of Function Evaluations

Mean number of function evaluations (MNFE) is commonly used metrics for assessing convergence properties of many GO methods. This metric is also used in [1], but we believe that the MNFE needed for StGA is not correctly calculated in [1].

Accordingly to the results shown in Table II of [1], $MNFE = (NP * NS) * NG$, (where NP – population size, NS – number of children produced during local selection in each region and NG – number of generations), which would give the number of function evaluations only for the local selection stage, but not for the computation of each new generation and also excludes the number of initial points. The actual process count, in our view, should be: $NP(I + NS + I + NS + I + \dots + NS + I)$. Therefore, the total number of function evaluations should be: $NP + NG * (NP * NS + NP)$.

For function f_1 this would give 39030 number of function evaluations in total, instead of the 30000 reported in [1]. However, even if the counting was correctly done, there is still significant difference between the performance of StGA and the other Evolutionary heuristics used in this comparative study.

4 Simulation of StGA and Comparison of Results

We simulated in Matlab the proposed by Tu and Lu StGA method and tested it on the same test functions and used the same parameter values as given by the authors (in Table II and Table VII of [1]). The obtained results are given in Table 1.

Table 1. Comparison of StGA [1] with StGA* (our implementation), FEP and GLPrS (without function tuning) in terms of number of function evaluations. All functions are 30 dimensional.

Fun	StGA	StGA*	GLPrS	FEP
f_1	30 000	111 000	117 355	150 000
f_2	17 600	174 000	105 673	200 000
f_8	1 500	265 000	96 600	900 000
f_9	28 500	250 000	132 183	500 000
f_{10}	10 000	118 000	103 229	150 000
f_{11}	52 500	165 000	101 495	200 000
f_{12}	8 000	125 000	111 884	150 000
f_{13}	16 000	135 000	122 835	150 000

We also performed parameter tuning individually for each testing function when implementing our GLPrS method [4], and our simulation of DE method [5] (Table 2).

As expected there was a large discrepancy between the reported in [1] results (second column of Table 1 and Table 2) and our implementation StGA* of the method (column 3 of the tables). It can be seen from Table 1 and Table 2 that the results of StGA* are inferior in most of the cases (with the exception for FEP).

Table 2. Comparison of StGA [1] with StGA* (our implementation), FEP, GLPrS and DE (last two methods with individual tuning for each function). All functions are 30 dimensional.

Fun	StGA	StGA*	FEP	GLPrS	DE
f_5	45 000	1×10^6	2×10^6	43 186	115 137
f_8	1 500	265 000	900 000	14 333	20 691
f_9	28 500	250 000	500 000	66 729	118 936
f_{10}	10 000	118 000	150 000	18 741	46 157
f_{11}	52 500	165 000	200 000	14 446	34 210

Results from additional experiments and simulations are also given in Table 3. Here, in the second column, we show the accuracy (mean value and standard deviation given in parentheses) for each function reported in [1]; in the third column - the analytic minimum that we want to achieve; in the fourth column - the accuracy results (mean and standard deviation) achieved by our simulation of StGA; in the fifth column - the number of successful runs out of 20; and in the last two columns we give the best and worst optimization values achieved in those 20 runs. We used as a stopping criterion the number of function evaluations given in Table II of [1], as we expected the method to converge to a global minimum for that expense (as reported by

Table 3. Comparison of the StGA accuracy results reported in [1] and our implementation StGA*. The same parameter values are used in both cases.

	StGA in [1]	Analytic Min	StGA* our simulation	Succ. Runs	Min	Max
f_1	2.45_x10^{-15} (5.25_x10^{-16})	0	2.54_x10^{+3} (741.58)	0/20	505.31	3.9_x10^{+3}
f_2	2.03_x10^{-7} (2.95_x10^{-8})	0	17.6 (18.67)	0/20	6	93.875
f_3	9.98_x10^{-29} (6.9_x10^{-29})	0	6.8_x10^{+5} (1.86_x10^{+5})	0/20	4.5_x10^{+5}	1.2_x10^{+6}
f_4	2.01_x10^{-8} (3.42_x10^{-9})	0	25.47 (3.76)	0/20	17.97	32.84
f_5	0.04435 (0.0)	0	563.37 (686.37)	0/20	54.05	2.3_x10^{+3}
f_6	0.0 (0.0)	0	3.58_x10^{+3} (465.8)	0/20	2.65_x10^{+3}	4.6_x10^{+3}
f_7	8.4_x10^{-4} (1.0_x10^{-3})	0	2.31_x10^{+4} (2.1_x10^{+4})	20/20	6.4_x10^{-5}	9.5_x10^{-4}
f_9	4.42_x10^{-13} (1.14_x10^{-13})	0	16.155 (23.95)	9/20	0	101.7
f_{10}	3.52_x10^{-8} (3.51_x10^{-9})	0	16.33 (2.8)	0/20	8.593	18.85
f_{11}	2.44_x10^{-17} (4.54_x10^{-17})	0	145.29 (28.99)	0/20	98.65	214.25
f_{12}	8.03_x10^{-7} (1.96_x10^{-14})	0	173.34 (34.97)	0/20	121.28	275.38
f_{13}	2.01_x10^{-8} (3.42_x10^{-9})	0	3.8_x10^{+3} (1.34_x10^{+4})	0/20	266.45	6_x10^{+4}
f_{14}	1.0 (0.0)	1	0.998 (7_x10^{-8})	20/20	0.998	0.998
f_{15}	3.179_x10^{-4} (4.726_x10^{-6})	3.08_x10^{-4}	5.49_x10^{+4} (2.8_x10^{+4})	20/20	3.2_x10^{-4}	0.0012
f_{16}	-1.03034 (1.0_x10^{-3})	-1.0316	-1.0316 (4.53_x10^{-6})	20/20	-1.0316	-1.0315
f_{17}	0.3986 (6.0_x10^{-4})	0.398	0.3979 (2.54_x10^{-5})	20/20	0.3979	0.3980
f_{18}	-9.828 (0.287)	-10.1532	-4.28 (2.71)	3/20	-10.1532	-2.63
f_{19}	-10.40 (0.0)	-10.4029	-6.01 (3.71)	8/20	-10.403	-1.8371
f_{20}	-10.45 (0.037)	-10.5363	-5.21 (3.21)	5/20	-10.5363	-1.8589
f_5^{100}	1.01 (1.1959)	0	6.8_x10^{+6} (3_x10^{+6})	0/20	2.65_x10^{+6}	1.22_x10^{+7}
$g(x)$	-78.29368 (0.03)	-78.3322	-56.6687 (1.57)	0/20	-60.23	-53.74

the authors). As it can be seen from Table 3, the StGA method failed to converge in most of the cases (with the exception for functions f_{14} to f_{17}).

We also experimented leaving the method to run until a global solution (or close enough global solution) is found and in all cases (with the only exception of the 2 dimensional test functions), the computational expense was considerably higher than the one reported in [1]. For example, for function f_1 we needed more than 110 000 function evaluations (compared to the reported 30 000), in the case of function f_2 - more than 170 000 (compared to the reported 45 000), etc.

The results from our StGA implementation can be summarized as follows:

- a) For the 100 dimensional cases (f_5^{100} and $g(x)$) the StGA simulation (using the authors' parameter values) produced much worse than the reported in [1] results and we were not able to achieve even a single successful run;
- b) For the 30 dimensional test functions ($f_1 - f_{13}$), we achieved successful runs for two functions only (about 50% success rate for function f_9 and 100% success rate for f_7). For all other functions the method simply failed to converge for the specified number of function evaluations. The obtained results were not even close to the ones reported in [1]. Our hypothesis is that the accidental success for functions f_7 and f_9 is due to the small number of binary bits used for them – 8, as well as the position of the minima – in the origin (0, ..., 0). Also, the small search intervals for those functions and the way binary coding works helped StGA to converge accidentally to a global minimum in both of those cases;

- c) For the cases of 4 dimensional test functions ($f_{18} - f_{20}$), the results of StGA simulation looked a bit better, with an average of 25% percent successful runs, but were still far from the ones reported in [1]. For one of the four-dimensional cases – function f_{15} , our results were comparable with those given in [1], (but with worse accuracy);
- d) For all two-dimensional cases (functions f_{14} , f_{16} and f_{17}) we received very close to the published in [1] results (for f_{16} and f_{17} they were even slightly better).

5 Conclusion

Several GO heuristic evolutionary approaches (StGA, FEP, DE and GLP τ S) have been briefly investigated, reviewed, tested on a number of multimodal mathematical functions of high dimensionality and critically compared based on their convergence speed and accuracy.

The results of our simulation of StGA method lead to a conclusion that the accuracy and convergence characteristics (such as number of function evaluations for reaching a global minimum) differ significantly from those reported in [1]. For most of the test cases (with the only exception for the two-dimensional test functions) the method simply failed to reach a global solution for the specified number of function evaluations. The authors rightly claim their stochastic coding strategy as a novelty (although it might be considered as a memetic algorithm), but in our view, their implementation of chromosomes integer binary coding limits significantly the power of their search approach. The results from our implementation show that the method either accidentally hits a global solution (mainly, because most of the test functions are with global minimum in the origin), or their software implementation contains a bug that produced these exceptionally superior results.

For all of the test functions StGA produced inferior results comparing to DE and our GLP τ S method.

References

1. Tu, Z., Lu, Y.: A Robust Stochastic Genetic Algorithm (StGA) for Global Numerical Optimization. *IEEE Trans. on Evolutionary Computation* 8(5), 456–470 (2004)
2. Jordanov, I., Georgieva, A.: Neural Network Learning with Global Heuristic Search. *IEEE Trans. on Neural Networks*, 937–942 (2007)
3. Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. *IEEE Trans. on evolutionary Computation* 3(2), 82–102 (1999)
4. Georgieva, A., Jordanov, I.: Global Optimization Based on Novel Heuristics, Low-discrepancy Sequences and Genetic Algorithms. *European J. of Oper. Research* (to appear, 2008)
5. Price, K., Storn, R., Lampinen, J.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer, New York (2005)
6. Joines, J., Kay, M.: Utilizing Hybrid Genetic Algorithms. In: Sarker, R., Mohammadian, M., Yao, X. (eds.) *Evolutionary Optimization*, pp. 199–227. Kluwer, Boston (2002)
7. Hedar, A., Fukushima, M.: Minimizing Multimodal Functions by Simplex Coding Genetic Algorithm. *Optimization Methods and Software* 18, 265–282 (2003)
8. Hart, W., Krasnogor, N., Smith, J.: Recent Advances in Memetic Algorithms. In: *Studies in Fuzziness and Soft Computing*, vol. 166 (2005)

Generating Routes with Bio-inspired Algorithms under Uncertainty

Maria Belén Vaquerizo García

Languages and Systems Area
Burgos University, Burgos
belvagar@ubu.es

Abstract. The planning of an optimal design of routes is a complex problem of optimization and belongs to the type of NP-Hard problems. In this case, to find an exact solution is nonviable, and, therefore, it needs methods that assure the optimal management of the real resources to the design of the new routes under the best criteria about times and costs.

This paper proposes the use of heuristic algorithms bio-inspired for the optimization in the design of the routes under diverse restrictions in the collective urban public transport in a town. This is because there are many applications in the transport field where this type of heuristic have proved to be very efficient. Moreover, among the variables that have greatest impact in developing this work, is the passenger demand, seen as uncertain data. For his treatment, it is suggested the use of the Fuzzy Sets Theory.

Therefore, the purpose of this study is to establish a model for solving a complex and uncertain problem.

Keywords: Collective Urban Public Transport, Routes Generator, Combinatorial Complexity, Ant Colony Optimization, Fuzzy Demand, Triangular Fuzzy Numbers.

1 Introduction

The present study examines the problem of generating routes in a system of collective urban public transport for any mid-size city. It requires having a model for the suitable planning of such lines [1]. The problem has to consider a large number of decision variables and with uncertain data, which is the passenger demand to cover. Therefore, the problem is considered as a complex and uncertain problem of combinatorial optimization.

A difficulty of this work, to get the service previously established, is because of the impossibility of knowing the exact number of passengers going up and down at every stop in each line and in each time period considered within the timeframe set. However, it may be possible to estimate to these values using mathematical techniques to work with estimated data. Thus, it is suggested, as a possible alternative, the use of the Fuzzy Sets Theory [12], to keep the information on the terms under which has been expressed.

Moreover, it is an optimization problem in terms of its resolution, so it must be polynomial programmable [13]. So, it needs procedures to provide a solution to the problem through a number of operations depending on the size of the problem. Among these procedures are those that take the behavior of nature [3] as a reference point for its development, also known as Bio-inspired Algorithms. In this area, Ant Colony Optimization algorithms, from now on is referred as ACO, are relatively recent and have proved to be efficient in these types of problems. They are based on the imitation of how ants of a colony find the shortest path between the nest and the food.

Therefore, to solve this problem, in Section 2 this paper analyze the social problem and the main features to be considered to solve the problem about collective urban public transport. And, with this information, in Section 3 it has been proposed the following: The construction of routes through an algorithm based on the Ant Colony, and the application of the Fuzzy Sets Theory to be able to operate with fuzzy demand. This paper finalize with the conclusions and evaluation of the strategy used.

2 Social Problem and Main Features in Urban Public Transport

Currently, designing routes in collective urban public transport has a great complexity in their operations, caused by the need to provide a good and effective service to the passengers who are in the stops in the different routes. To provide a quality service, we must build a set of optimal routes to cover all passengers demand with the minimum number of buses, since the number of lines to build depends on the number of buses to take. To do so, it must find a system of routes for each pair origin-destination, with the best results in economic terms, but always getting a fixed level of service previously established.

Overall, maximizing the service quality for the passengers is considered as minimizing travel times and waiting times, and maximizing the service quality for transport companies is considered as reducing the number of buses. In any case, both of them depend on the number of passengers who are in the different stops on the topographical network studied. Therefore, as the demand is not a concrete date, it requires the use of techniques to work with estimated data.

Moreover, it should be considered that changes on routes without the proper tools can cause an excessive increase in their time of travel and that passenger demand could not be covered. Therefore, it is necessary to consider the main features of the problem and the algorithm to use.

2.1 Combinatorial Complexity

This problem is considered a complex problem of combinatorial optimization, which belongs to the class of problems known as NP-hard, those whose exact resolution requires specific models and algorithms with great difficulty and much processing time to obtain the optimal solution.

Otherwise, because of the size of this type of problem, it is considered that could be solved in a reasonable computer time through heuristics techniques or efficient algorithms, to provide a “good” solution, which is relatively close to the optimum by examining only a small subset of the total number of solutions. For this reason, the

problem is developed through the use of heuristics techniques, concretely algorithms based on the nature, and in a specific way, through an ACO [4]. Complex systems are composed of a large set of individual components that interact with each other and they can change their internal states as a result of these interactions. The process of interactions can generate collective and global behaviour.

This paper uses one of the more simply algorithms based on the Ant Colony, namely as Ant Colony System [9], in order to construct a routes Generator to this problem, where the elements that interact among themselves, called agents, establish their relations, the type of communication that may have, when they can do it, and the type of information they can share [2].

In its application to a real problem it is necessary that the same can be represented as a graph. Thus, the following Figure represents the search of a path between two stops in any network or graph.

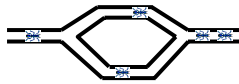


Fig. 1. Ant Colony between two stops

2.2 Fuzzy Demand

The Fuzzy Sets Theory is a branch of mathematics which considers the subjective and the uncertain. If in a problem, the available information is vague or uncertain, is better use realistic representations but imprecise than only supposedly accurate models. However, the probability and possibility can be linked into a single model, but having in mind that they are different concepts.

Therefore, in this paper, by the consideration that the demand is seen as uncertain data, not fixed, it was decided to work with Fuzzy Numbers, concretely Triangular Fuzzy Numbers. So, it is possible to represent value of the number of passengers between two stops, generating other two values, representing with it the minimum, the average and the maximum demand that can occur between these two stops, trying in this way the uncertainty in demand.

This work uses Fuzzy Numbers and Fuzzy Sets, and they are different concepts. Next, the following figures represents both of these terms, where in the Figure 2 you have a Fuzzy Set between 5 and 8 to represent a range of demand, namely as A, and in the Figure 3 there is a Fuzzy Number around 4, to represent a concrete value of demand, namely as B.

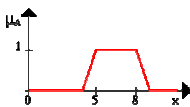


Fig. 2. A Fuzzy Set for the demand between 5 and 8, namely as A

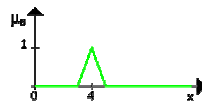
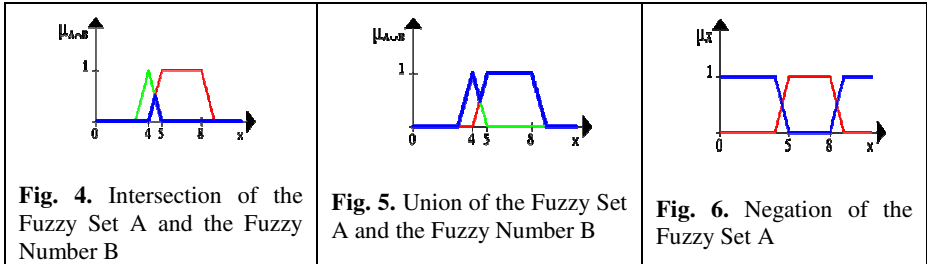


Fig. 3. A Fuzzy Number for the demand around 4, namely as B

Next, the following Figures, considering the ones before, show some operations that can be done with these data about the demand. Figure 4 shows the intersection operation “AND” of the Fuzzy Set A and the Fuzzy Number B. Figure 5 shows the union operation “OR” of the Fuzzy Set A and the Fuzzy Number B. And, finally, Figure 6 shows the negation operation “NOR” of the Fuzzy Set A.



The description of the previous operations has been done for finite demand, it is easy to generalize to infinite values of the demand.

3 Routes Generator with ACO Algorithms under Uncertainty

Therefore, as discussed before, the problem here studied is the urban network of any medium-sized city, composed of a set of nodes that will be the stops and a set of edges that connect pairs of nodes or stops. Thus, it may be assumed that the problem of itineraries to resolve must be to visit nodes or edges of a graph, in terms of finding the shortest paths between each two stops previously considered as origin and destination, and satisfying a number of additional restrictions. The system, based on input data and some restrictions previously established, must determine, in an effective way, an optimal design of routes, which provide a fixed level of service and cover the demands required respecting the number of buses available. It also provides information that needs to be relevant, flexible and adaptable to the changing conditions on the environment on which applies.

3.1 ACO Algorithms

Currently there are many algorithms based on ACO, such as: Ant System [10], Ant Colony System [9], Max-Min Ant System [14], Elitist Ant System, Rank-Based Ant System [5] and Best-Worst Ant System [7,8]. Moreover, in any of them, this problem can be represented as a graph with weights, in which each arc contains two different types of information and with different functions, namely:

- Heuristic Information: which represents a measure of the cost of the arc, depending on the specific case, which is calculated before starting the algorithm and it is not changed during the performance of the algorithm.
- Stored Information: which contributes a measure of the “desirability” of the arc, represented by the amount of pheromone placed on it and changed during the implementation of the algorithm (based on the number of ants which went over the same one previously).

Thus, in this study, an ant has been defined as an artificial agent that:

- Remember all the nodes which it has visited, which are stored in a list.
- At the end of each iteration, this list contains the solution built by the ant.
- At each step, from the current node, it chooses a node to move between the possible nodes that have not been visited yet, as a rule probabilistic transition.
- Once built its solution, it leaves a trail of pheromone (in an amount that depends on the kindness of this solution) on each arc through has gone by and empty the list referred to in the first point.
- Alternatively, it can also deposit pheromone on each arc that runs while building the solution.

According to this, is necessary to determine how to establish the rules of transition states and of updating the pheromone.

3.2 Triangular Fuzzy Numbers to Handle Fuzzy Demand

In this work, it is used a particular case of Fuzzy Numbers to represent the value of the demand, namely as Triangular Fuzzy Numbers. And, as its name suggests, it has triangular shape perfectly defined by three real numbers, representing with it the range of the demand. The meaning of these three real numbers that define a Triangular Fuzzy Number are: an initial amount below which it will not descend (a_1), other over which is impossible to reach (a_3), and a third (a_2) which represent the highest level of presumption.

Thus, a value of the demand is represented by a Triangular Fuzzy Number that can be represented in the two next ways:

1. Through the list of real numbers that describes it:

$$A = (a_1, a_2, a_3) \quad a_1, a_2, a_3 \in \mathbb{R} \quad a_1 \leq a_2 \leq a_3$$

2. Through its role as membership:

$$\mu_{\mu_A}(x) = \begin{cases} 0 & x \leq a_1 \\ \frac{x - a_1}{a_2 - a_1} & a_1 < x \leq a_2 \\ \frac{a_3 - x}{a_3 - a_2} & a_2 \leq x < a_3 \\ 0 & a_3 \leq x \end{cases}$$

3.3 Construction of a Routes Generator

Once chosen any algorithm based on ACO and with the application of Fuzzy Sets Theory [6], by the justifications before, the next is to establish, in a more precisely way, the data to construct the routes, as well as the rules to insert the stops in the routes.

For collecting and processing the necessary information, it requires a previous and accurately analysis of the actual situation of the urban transport in the city studied, to allow analyzing the characteristics of this problem. This information will be:

1. Geographic information:
 - 1.1. Network routes: Current routes and their frequencies

- 1.2. Available resources: Buses (number, types, capacities, etc)
- 1.3. Any other relevant information: Restrictions on routes, facilities for disabled people, etc
2. Reports of demand: Estimations of passengers going up and down at every stop in each line, and each time period.

Therefore, from this study of the actual network that represents a town, it's also necessary to get the following input data:

- Number of stops or nodes in the network
- Minimum distance between every two stops
- Ordinary demands of passengers between those stops per hour
- Special demand among pairs of stops and in what time or time interval considered it occurs.
- Stop or stops representing the city centre
- Stop or stops that represent extremes of the city
- Maximum number of routes allowed to be constructed
- Knowledge about stops with have some type of restriction
- Percentage of demand allowed not to be covered

Thus, these data are part of the information of the network considered. Moreover, we also have to consider the next decision variables:

- On how obtaining their lines and routes
- On the insertion of stops in the paths of the routes

Therefore, the identification of all these data (input data and decision variables) will allow subsequently to formalize the problem to solve. Finally, it should be noted that the evaluation of the solution implies calculating the objective function of the model, and that, moreover, it implies to choose a resolution method, in which there must be a commitment between efficiency and quality of the solution obtained.

On the other hand, the mathematical model, that represents the problem studied, can be seen as a multiple objective function (maximize the level of service and minimize the number of buses required) [11] or as a single objective function (maximize the level of service, considering the number of buses available as a restriction). In both cases, it should be noted that the approach to maximize the level of service is considered as minimizing time travel and waiting time of passengers at stops.

For this reason, when a problem routes is discussed, it is often addressed through the formulation of models whose decisions can be obtained through algorithms to solve instances of suitable size in a optimal way and in a reduced time.

Finally, based on all the previous considerations, in the next example, it has been implemented an Ant Colony System with Triangular Fuzzy Numbers to work with the demand. Programs are designed to get the feasible routes that can be formed from a series of stops taken as initial data. Previously, it is necessary, for each possible line to be constructed, to have established their origin and destination stops.

Thus, considering the previous variables and for the following network considered, the system would provide the next designs of routes shown in the next Figures.

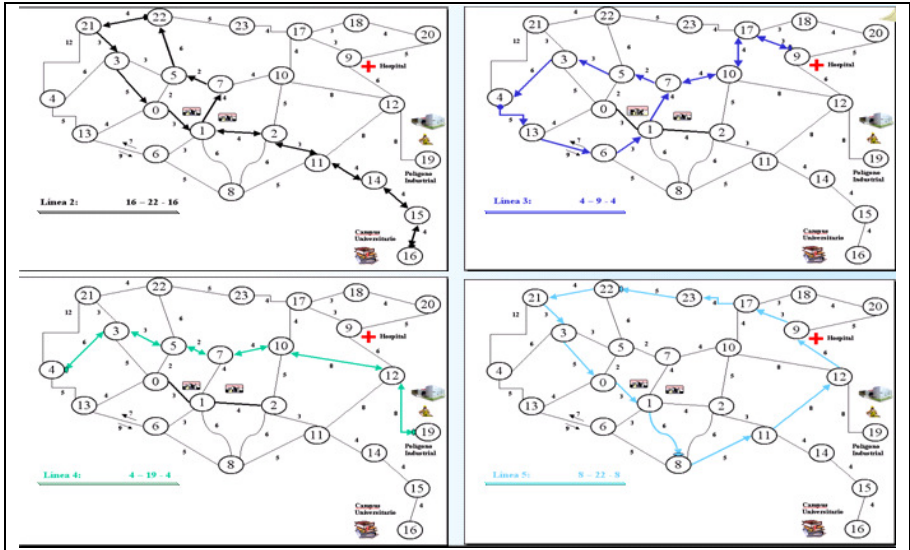


Fig. 7. Generation of Lines and their paths

These feasible routes satisfied specific conditions based on the number of stops, total distance, maximum time travel, passenger demand satisfied, etc.

4 Conclusions and Evaluation of the Strategy Used

A real problem of this type presents some features, mainly in terms of its size, excessive input variables, and also requires that the solution is for real-time.

An important contribution of this paper is that it differs from the resolution strategies that can be found in the existing literature on the subject. So, among the main difficulties of this work, to get the service previously established, is the need to know the number of people using each line in each time period. According to it, this work uses mathematical strategies, Fuzzy Sets Theory, to control uncertainty data, the passengers demand in this case. Consequently, the process adds complexity on how to solve the problem, but it ensures obtaining better solutions.

Moreover, the algorithms based on ACO are simple conceptually and, thus, they are easy to program and adapt to a high number of problems. However, it has been proved that for big networks, with a lot of restrictions and decision variables and other considerations, they obtain disappointing results, one reason is the use of randomness as the primary mechanism for searching.

By the other hand, all these algorithms can be improved with a local search algorithm that improves the quality of solutions, including 2-Opt once all the ants have completed their path and before the global update of the pheromone.

References

1. Baaj, M.H., Mahmassani, H.S.: An AI-Based Approach for Transit Route System Planning and Design. *Journal of Advanced Transportation* 25(2) (1991)
2. Bajo, J., Corchado, J.M.: Construyendo sistemas basados en agentes: de la teoría a la práctica. IWPAAMS, pp. 5–14. Universidad de León (2005) ISBN: 84-9773-222-7
3. Bentley, P.: Digital Biology. How Nature is Transforming our Technology. *Headline* (2001)
4. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence. From Nature to Artificial Systems*. Oxford University Press, Oxford (1999)
5. Bullnheimer, B., Hartl, R.F., Strauss, C.: A new rank-based version of the Ant System: A computational study. *Central European Journal for Operations Research and Economics* 7(1), 25–38 (1999)
6. Casillas, J., Cordon, O., Fernández de Viana, I., Herrera, F.: Learning Cooperative Linguistic Fuzzy Rules Using the Best-Worst Ant Systems Algorithm. *International Journal of Intelligent Systems* 20 (2005)
7. Cordon, O., Fernández de Viana, I., Herrera, F., Moreno, L.: A new ACO model integrating evolutionary computation concepts: The Best-Worst Ant System. In: *Abstract proceedings of ANTS 2000*, pp. 22–29. IRIDIA, Université Libre de Bruxelles, Belgium (2000)
8. Cordon, O., Herrera, F., Moreno, L.: Integración de Conceptos de Computación Evolutiva en un Nuevo Modelo de Colonias de Hormigas. In: *VIII Conferencia de la Asociación Española para la Inteligencia Artificial, Murcia, España, vol. II*, pp. 98–105 (1999)
9. Dorigo, M., Gambardella, L.M.: Ant Colony System. A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1(1), 53–66 (1997)
10. Dorigo, M., Maniezzo, V., Colomi, A.: The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans. On Systems, Man and Cybernetics-Part B* 26 (1996)
11. García-Martínez, C., Cordon, O., Herrera, F.: A Taxonomy and an Empirical Analysis of Multiple Objective Ant Colony Optimization Algorithms for Bicriteria TSP. *European Journal of Operational Research* 180(1) (2007)
12. Kaufmann, A., Gil Aluja, J.: *Introducción de la Teoría de los Subconjuntos Borrosos en la Gestión de Empresas*. Milladoiro, Santiago de Compostela (1986)
13. Sarker, R., Mohammadian, M., Yao, X.(eds.): *Evolutionary Optimization*, International Series in Operations Research and Management Science, vol. 48. Kluwer Academic, Dordrecht (2002) ISBN 0-7923-7654-4
14. Stützle, T., Hoos, H.H.: MAX-MIN Ant System. *Future Generation Computer Systems* 16(8), 889–914 (2000)

Computer-Assisted Diagnosis of Primary Headaches

Svetlana Simić¹, Dragan Simić², Petar Slankamenac¹, and Milana Simić-Ivkov³

¹ Institute of Neurology, Clinical centre Vojvodina, Hajduk Veljkova 1-9,
21000 Novi Sad, Serbia
dsimic@eunet.yu

² Novi Sad Fair, Hajduk Veljkova 11, 21000 Novi Sad, Serbia
dsimic@nsfair.co.yu

³ Clinic of Dermatovenerology, Clinical centre Vojvodina, Hajduk Veljkova 1-9,
21000 Novi Sad, Serbia
ssimic@uns.ns.ac.yu

Abstract. Headache is not a disease which typically shortens one's life. However, it can be a serious social as well as a health problem. Approximately 27 billion euros per year are lost through reduced work productivity in the European Community. While the diagnostic criteria developed by the International Headache Society (IHS) have been extensively used in the epidemiological research, there is no such a tool which helps physicians make diagnoses. This research focussed on diagnosing certain primary headache types in working people employing the rule-based fuzzy logic system. The rules were facilitated by the application of the IHS criteria for headache types. Clinical experience was used to extend the established rules and improve the system. The proposed system is in the starting phase of the implementation at the Clinical Centre Vojvodina, Institute of Neurology in Novi Sad.

Keywords: headache, diagnosing, rule-based fuzzy logic.

1 Introduction

Headache is not a disease which typically shortens one's life. However, it can be a serious social as well as a health problem. Headache is a common disorder amongst working people. A lot of people with headache are unaware of the type of headache they are suffering from. People do not always visit physicians for their complaints, and when they do, physicians do not always make a correct diagnosis.

Migraine is a highly disabling, associated with a substantial economic burden. Approximately 13 billion dollars per year are lost through reduced work productivity in the United States [1]. Migraine is the most expensive neurological problem in the European Community costing 27 billion euros annually [2].

While the diagnostic criteria developed by the International Headache Society (IHS) have been widely validated, there is no reliable (software) tool that would help physicians make diagnoses [3]. Our efforts are directed towards developing the software tool for daily clinical practice in order to distract physician's attention from primary headache. Sometimes, the doctor, preoccupied with work, is satisfied with

excluding the organic cause of headache. However, the diagnosis is not conducted to the end, which further on has an influence on to the therapy approach.

The results of our research on 80 patients which suffer from headache were: 12.5% patients with migraine, 47.5% patients with tension type headache, and 40% patients with other primary headache types; also 20% of those suffering from migraine suffer from migraine with aura, while 80% suffer from migraine without aura. The results coincide with the research presented in [4]. Now, for the time being, the tool is developed for migraine – migraine without aura, migraine with aura – tension type headache and other primary headaches, but our plan is to classify the cluster headache and other primary headaches as well.

The rest of the paper is organised as follows. Section 2 presents the rule-based fuzzy logic (RBFL) as a successful technique for knowledge-based decision support. Basic headache classification and designing the health questionnaire are outlined in section 3. Knowledge mining using health questionnaires, and the organisation of our questionnaire are given in section 4. Section 5 describes the results of the health questionnaire and some improvements to the RBFL system. Section 6 describes related work and section 7 concludes the paper.

2 Rule-Based Fuzzy Logic System

Rule-based fuzzy logic (RBFL) is considered to be a successful technique for knowledge-based decision support in many domains, including medicine. In this research, the rule-based fuzzy logic system was used as a tool for diagnosing certain primary headache types.

Rule-based fuzzy logic systems contain four components as shown in fig. 1. Once the rules have been established, the fuzzy logic system can be viewed as a mapping from inputs to outputs.

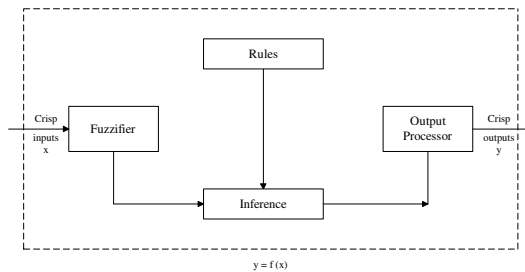


Fig. 1. Rule-Based Fuzzy Logic System - basic model

Rules are the heart of the fuzzy logic system, and may be provided by an expert or can be extracted from numerical data. In either case, the rules we are interested in can be expressed as a collection of IF-THEN statements. The IF-part of a rule is the antecedent, and the THEN-part of a rule is its consequent. In general, in their original form, rule-based fuzzy logic systems were initially used in: 1) forecasting time-series, and 2) knowledge mining using surveys [5].

3 Headache and Designing a Health Questionnaire

The International Classification of Headache Disorders 2nd edition (ICHD) is perhaps the single most important document to read for physicians taking an interest in the diagnosis and the management of headache patients. All headache disorders are classified into two major groups: 1) Primary headaches and 2) Secondary headaches. Each group is further subdivided into types, subtypes and sub-forms. Primary headaches are thus subdivided into: 1) Migraine, 2) Tension-type headache, 3) Cluster headache and other trigeminal autonomic cephalalgias and 4) Other primary headaches. There are only six types of migraine, yet these are further divided into 17 subtypes.

Considering the wide range of migraine and other headache types, our research focussed on migraine without aura, migraine with aura, tension-type headache and other primary headaches.

Health questionnaires can cover a variety of topics, such as the etiology of disease, response to illness or maintenance of health on the part of the patient or the public, and the health care staff and organisation. The health status is the explicit or implicit focus of health questionnaires - factors that directly or indirectly influence, measure or are affected by people's health. The diagnostic criteria developed by the IHS have been widely validated and were used for creating an original questionnaire. The methodological approach to diagnosis of certain headache types – the "Diagnosis Cycle" is presented in fig. 2.

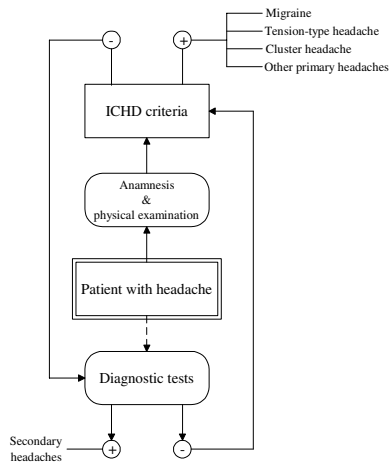


Fig. 2. The "Diagnosis Cycle" – methodological approach to diagnosing headache types

A number of the feed backloops reflect the fact that designing a questionnaire is a dynamic, iterative and interactive process. Many decisions have to be made in tandem, and the advent of a computerised system that can be used to carry out many or all the phases of a questionnaire has made this even more true.

4 Knowledge Mining Using Health Questionnaires

Knowledge mining means extracting information in the form of IF-THEN statements made by questioned people, in our research - patients. These rules can be modeled using the fuzzy logic system. It is very important to emphasise that establishing the rules in our research was facilitated by our knowledge and the application of the International Headache Disorder Criteria (IHDC) for headache types. The diagnostic criteria for migraine without aura according to the IHDC are as follows:

- A. At least 5 attacks fulfilling criteria B-D; *From the second questionnaire*
- B. Headache attacks lasting; *Question 3, answer b) from 4 hours to 3 days*
- C. Headache has at least two of the following characteristics:
 - 1. location; *Question 6, answer a) unilateral*
 - 2. quality *Question 4, answer a) pulsating*
 - 3. pain intensity *Question 5, answer b) temperate or c) strong*
 - 4. aggravation of routine physical activity *Questions 7 and 8, answer a) yes*
- D. During headache at least one of the following:
 - 1. nausea and/or vomiting
Question 9.1, answer a) yes; Question 9.2, answer a) yes, or b) no
 - 2. photophobia and phonophobia; *Questions 9.3 and 9.4, answer a) yes*
- E. Not attributable to another disorder; *From the second questionnaire*

Questions based on the IHDC represent an input on our model, according to the basic RBFL model (fig. 1.). Output represents the specific type of headache: migraine without aura, migraine with aura, tension type headache and other primary headaches. The rules, which are the heart of the fuzzy logic system, are provided according to the IHDC and presented as the collection of IF-THEN statements.

5 First Results and Improvements

After inserting the responses of 80 questioned subjects, an analysis of headache types, received as the inserted results of responses, was carried out. It was not expected to obtain a lot of other primary headaches diagnoses.

The high specificity of IHS diagnostic criteria, while ideal for clinical trials, appears to lack the necessary sensitivity for migraine mandated in routine clinical practice. This lack of clinical sensitivity may be a contribution factor to the continued underdiagnosis and undertreatment of migraine [6]. The system presents the following results (table 1.):

Table 1. First results of a Headache health survey

Headache type	Number	%
<i>Migraine with aura</i>	0	0
<i>Migraine without aura</i>	7	8.75
Migraine	7	8.75
Tension type headache	26	32.5
Other primary headaches	47	58.75

Working experience of physicians was used in order to extend the established rules and improve the system for suggesting diagnosis in patients suffering from headaches. This clearly shows that in order to establish the rules it is not enough to use defined standards or professional textbooks and manuals, but it is necessary to include physicians with, ad hoc, their practical experience. Also:

1. Differentiating between *migraine with aura* and *infrequent episodic tension type headache* may be difficult.
2. Differentiating between *migraine without aura* and *chronic migraine* may be difficult.
3. Pain in migraine is one-sided, pulsating and severe or moderate and unbearable. Unbearable headaches are often described as diffuse in patients suffering from *migraine*.
4. Patients sometimes characterise moderately severe headache as oppressive.

Based on our own clinical experience, we have made a decision to apply the modified established IHD criteria. The improvement of the system has been carried out by adding, subtracting or changing the criteria but by increasing the significance of some.

The improvement and the extension of the decision-making logic were conducted in a manner where IHD criteria for migraine without aura: B, C1, C2, and C3 were kept as obligatory conditions for migraine without aura determination. Other criteria, C4, D1, and D2, are optionally used with altered logic among them. Considering that C4, D1 and D2 represent optional decision-making criteria of migraine diagnoses, the term "surely assessed" is introduced. The term "surely assessed" shows the degree of assessed diagnoses of migraine types. Since there are 4 possible values observed, the value of "surely assessed" can vary from 0 to 3. It's obvious that 0 represents low level migraine type diagnosis assessment. The values 1 and 2 increase the value of accurate assessment in suggested diagnosis, while for "surely assessed" it could be said that the highest value is 3 (table 2.).

The improvements to the established rules removed certain fuzzy system shortcomings, making it possible to establish the diagnosis in patients suffering from headaches with more accuracy. Having improved the rules with new facts the new diagnoses were: 12.5% patients with migraine (20% of those suffering from migraine suffer from migraine with aura, while 80% suffer from migraine without aura), 47.5% patients with tension type headache, and 40% patients with other primary headache types. Our results are in the expected range but with a little higher number of other primary headaches and a smaller number of patients with tension-type headache.

Table 2. Improving the rules and assessment headache type of Headache health survey

Headache type	Number	Surely assessed	%
Migraine without aura	4	1	5
	4	3	5
	8		10
<i>Migraine with aura</i>	2	1	2.5
Migraine	10		12.5
Tension type headache	38	47.5	47.5
Other primary headaches	32	40	40

Epidemiological data indicates that migraine affects over 10% of the world population. The prevalence of migraine in Norway is 12%, similar to other western countries - Denmark 10%, France 12%, the USA 12%. In other cultures the migraine prevalence is lower: Japan 8%, Hong Kong 1%. These differences in the migraine prevalence amongst studies may be attributed to cultural, genetic, dietary, climatic or environmental factors. The prevalence of tension-type headache varies between 29-71%, the wide range being partly a result of different study designs [4].

The responses, given by the questioned patients suffering from headache are observed as a fuzzy process. While the same response applies to the different headache types, some of them apply to just one type. However, a patient may suffer from one or more headache types, but in one attack, the patient suffers from the specific one. The results of the questionnaire show presence of both types, simultaneously. The proposed software tool could be coded as a flexible rule-based model with parameters tuning, according to the [7], and also presented system in the [8].

According to the previous statement, it should be predicted, for the future development, that the system output may represent several diagnoses and not just one.

6 Related Work

As mentioned earlier, the original form of rule-based fuzzy logic systems were initially used for: 1) forecasting time-series and 2) knowledge mining using surveys [5]. The systems have already been applied along with their further development.

6.1 Application of Rule-Based Fuzzy Logic Systems

Rule-based fuzzy logic systems are already used in different domains:

1. D-FLER, is a distributed, fuzzy logic engine for rule-based wireless sensor networks [9]
2. A fuzzy rule-based methodology for downscaling local hydrological variables from large scale atmospheric circulation [10]
3. A fuzzy logic system applied to umbilical Acid-Base assessment, which can provide vital information on the infant's health and guide requirements for neonatal care [11].

6.2 Studies Involving Headache Patients

1. An approach to therapeutic management of patients suffering from migraine and primary headache in general
2. AIDA Cefalee is a diagnostic expert system able to suggest the correct ICDH-II diagnosis [12]

6.3 Applications of Modern Information and Communication Technologies

Latest information technologies (IT) are increasingly used in modern research in medicine, in addition to the interview and questionnaire, as illustrated by the internet questionnaire research conducted by the Swiss Society of Radiology [13].

6.4 Headache Research in (Post) Transitional Countries

1. The prevalence study comparing the lifestyles of female Belgrade students with migraine and non-migraine primary headache [14]
2. The epidemiological survey of the prevalence of migraine, tension-type headache and chronic daily migraine in Georgia [15].

7 Conclusion and Future Work

Headache is common in working people and represents a significant health as well as social and economic problem. The paper presents one of the approaches to diagnosing certain primary headache types. Our research demonstrated that the rule-based fuzzy logic method is a promising support in diagnostic decision making in medicine.

The rules in our research were facilitated by the application of the IHS criteria for headache types. Physicians' clinical experience was used in order to extend the established rules and improve the system. After the general rules were improved, our results were in the expected range.

Having improved the rules with the new facts the new diagnoses were: 12.5% patients with migraine, 47.5% patients with tension type headache, and 40% patients with other primary headache types. Our results are in the expected range but with a little higher number of other primary headaches and a smaller number of patients with tension type headache.

Now, for the time being, the software tool is developed for migraine and tension type headache but our plan is to classify cluster headache as well. This system for diagnosing primary headache types is the part of a project conducted at the Institute of Neurology of the Clinical Centre Vojvodina in Novi Sad [16], aimed at improving the primary headache diagnosis and headache severity assessment in patients suffering from headache.

References

1. Hu, X.H., Makson, L.E., Lipton, R.B., Stewart, W.F., Berger, M.L.: Burden of migraine in the United States: Disability and economic cost. *Arch. Intern. Medicine* 159, 813–818 (1999)
2. Andlin-Sobocki, P., Jonsson, B., Wittchen, H.U., Olsen, J.: Cost of disorder of the brain in Europe. *Journal European Neurology* 12 (suppl. 1), 1–27 (2005)
3. El Hasnaoui, A., Vraly, M., Blin, P.: Assessment of migraine severity using the MIGSEV scale: relationship to migraine features and quality of life. *Blackwell Publishing Ltd Cephalalgia* 24, 262–270 (2004)
4. Wober-Bingol, C., Wober, C., Karwautz, A., Schnider, P.: Tension-type headache in different age groups at two headache centers. *Pain* 67, 53–58 (1996)
5. Mendel, J.M.: *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Direction*. Prentice-Hall, Englewood Cliffs (2001)
6. Cady, R.K., Schreiber, C.P., Farmer, K.U.: Understanding the Patient With Migraine: The Evolution From Episodic Headache to Chronic Neurologic Disease. A Proposed Classification of Patients With Headache, *Headache* 44, 426–435 (2004)

7. Angelov, P.P.: Evolving Rule-based Models – A Tool for Design of Flexible Systems. *Studies in Fuzziness and Soft Computing*, vol. 92. Springer/Physica-Verlag, Heidelberg (2002)
8. Lemke, F., Müller, J.A.: Carcinogenicity Prediction of Aromatic Compounds Based on Molecular Description (2000), <http://www.knowledgeminer.net.pdf/carcino.pdf>
9. Marin-Perianu, M., Havinga, P.: D-FLER: A Distributed Fuzzy Logic Engine for Rule-based Wireless Sensor Networks. In: Ichikawa, H., Cho, W.-D., Satoh, I., Youn, H.Y. (eds.) *UCS 2007. LNCS*, vol. 4836, pp. 86–101. Springer, Heidelberg (2007), <http://www.springerlink.com/index/d3g3112884464315.pdf>
10. Bardossy, A., Bogardi, I., Matyasovszky, I.: Fuzzy rule-based downscaling of precipitation. *Journal Theoretical And Applied Climatology* 2(1-2), 119–129 (2005)
11. Ozen, T., Gribaldi, J.M.: Investigating Adaptation in Type-2 Fuzzy Logic Systems Applied to Umbilical Acid-Based Assessment. In: *Proceedings of 3rd International Conference of the European Network of Intelligent Technologies EUNITE* (2003)
12. De Simone, R., Coppola, G., Ranieri, A., Bussone, G., Cortelli, P., D'Amico, D.: Validation of AIDA Cefalee, a computer-assisted diagnosis database for the management of headache patients. *Journal Neurological Science* 28(suppl. 2), 213–216 (2007)
13. Lienemann, B., Hodler, J., Luetolf, M., Pfirrmann, C.W.A.: Swiss teleradiology survey: present situation and future trends. *Journal European Radiology* 12(10), 2157–2162 (2005)
14. Vlajinac, H., Špetić, S., Džoljić, E., Maksimović, J., Marinković, J., Kostić, V.: Some lifestyle habits of female Belgrade university students with migraine and non-migraine primary headache. *The Journal of Headache and Pain* 4(2), 67–71 (2003)
15. Katsarava, Z., Kukava, M., Mirvelashvili, E., Tavazde, A., Dzagnidze, A., Djibuti, M., Steiner, T.J.: A pilot methodological validation study for a population-based survey of the prevalences of migraine, tension-type headache and chronic daily headache in the country of Georgia. *The Journal of Headache and Pain* 8(2), 77–82 (2007)
16. Simić, S.: Assessment of quality of life in patients with migraine and tension type headache, Master thesis, Faculty of Medicine, University Novi Sad (2006)

Ambient Temperature Modelling through Traditional and Soft Computing Methods

Francesco Ceravolo, Matteo De Felice, and Stefano Pizzuti

E.N.E.A. – Casaccia R.C., Via Anguillarese, 301
00123 Rome, Italy
{francesco.ceravolo,matteo.defelice,
stefano.pizzuti}@casaccia.enea.it

Abstract. This paper presents a new hybrid approach based both on traditional and soft computing techniques to provide ambient temperature for those places where such a datum is not available. Indeed, we combine neural networks with the nearest neighbouring algorithm; we use a fuzzy logic decision maker and later compare the results of each single technique to the hybrid one. Experiments have been performed on several Italian places; results have shown a remarkable improvement in accuracy compared to single methods.

Keywords: Neural Networks, Fuzzy Logic, Nearest Neighbour algorithm.

1 Introduction

The design of energy production systems on an efficiently solar basis and projecting eco-sustainable buildings strongly depends on simulations where the accuracy of the hourly computation of several environmental values is critical.

When approaching this task two principal problems rise : the only available data are often based on monthly averages ; the data we possess concern few places nearby airports. As far as the first problem algorithms [4] help in evaluating to estimate reliable hourly values given the monthly ones. The second problem is slightly tougher. So far, tools like TRNSYS [6] are based on the fact that data are taken from historical databases and if it occurs that the database is lacking in the information we are looking for , the data of the nearest unknown location will be given as an estimate of the unknown one (the Nearest Neighbour - NN). This approach does not satisfy since climate is a highly non linear system and depends on a large number of variables. It is not always true that locations close to each other have similar environmental behaviours (for example temperature) thus causing a great number of mistakes.

On the other hand the scientific world has shown an increased interest in soft computing methodologies, like Artificial Neural Networks (ANN) and Fuzzy Systems (FS), which have proved [7][10] to be powerful tools to solve complex modelling problems for non-linear systems. Indeed, some applications of soft computing techniques in the field of modelling environmental parameters[8][12][13][14] have provided interesting results.

In this context we try to carry out a new hybrid approach trying to combine traditional (NN) with soft computing methods (ANN, FS) in order to model ambient temperatures. The main advantage of this approach is that we have a non linear interpolation tool capable to provide a reliable estimate when dealing with localities not present in the database. Experiments have been performed in several Italian localities.

2 The Methodologies

In this work we applied the following three methodologies : Nearest Neighbour (NN), Artificial Neural Networks (ANN) and Fuzzy Systems (FS).

2.1 Nearest Neighbour

This is a very simple, and for this reason widely used, method to estimate unknown climate data of a locality because it provides as estimation the data of the closest known locality according to the following formula :

$$t = t_i \quad (1)$$

Where t is the parameter to be estimated (for instance ambient temperature) and t_i is the datum of the i -th locality which corresponds to the closest one, therefore

$$i = \min(d_j) \quad (2)$$

for $j=1, \dots, n$ where n is the number of known localities and

$$d_j = \sqrt{(x - x_j)^2 + (y - y_j)^2} \quad (3)$$

Where x and y are the latitude and longitude of the requested locality and x_j and y_j are the latitude and longitude of the known localities.

2.2 Artificial Neural Networks

ANN are nowadays a widely known tool applied to solve a wide range of modelling problems. A broad scope of architectures as well as learning algorithms and a detailed comprehensive foundation on these topics can be found in [5]. In this work we applied the simplest neural paradigm : a feed-forward architecture trained by the Back-Propagation algorithm [11].

The topology we used is made of 4 input neurons (the three geographical coordinates and the month of the year), 6 hidden neurons and one output neuron (the average monthly temperature). All the transfer functions are the classic sigmoid. The neural models were trained for approximately 1000000 cycles and no early stopping criterion was used.

2.3 Fuzzy Systems

Since the first paper on fuzzy sets [15] a wide range of applications and theoretical issues have been developed [3][16] and nowadays this is a mature methodology. In this work we applied fuzzy sets in order to develop a Fuzzy Decision Maker (FDM) capable to combine the NN and the ANN estimations.

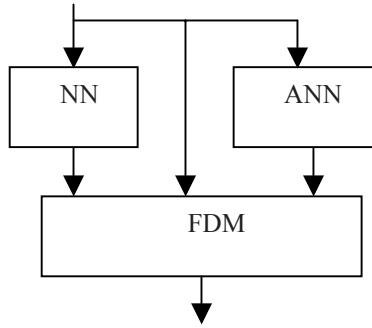


Fig. 1. Overall hybrid architecture

In general, to build up a fuzzy system it is possible to think of fuzzy rules typical of human reasoning and therefore build up qualitative and quantitative knowledge. In the figure below the steps composing a fuzzy system are represented.

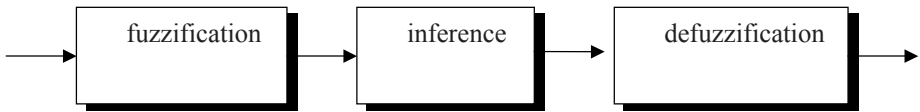


Fig. 2. Fuzzy system steps

The first step, the *fuzzification*, consists in the representation of the input values as fuzzy sets. For each input its membership value to the a priori defined fuzzy sets is determined by the corresponding membership functions. The choice of the most appropriate fuzzy sets, with the corresponding membership function, is a very important stage which remarkably affects the performance of the whole system. In the inferential step the composition rules among the fuzzy sets of the previous step are embedded. The result of this stage is the truth value of each rule. In the last step, the *defuzzification*, the final output of the system is the effect of the input and truth values of the rules of the previous step. For this task formal methods exist, among these the most effective is the ‘centre of gravity’ for which the output is computed as

$$\text{output} = \frac{\sum_{i=1}^n x_i \mu_R(x_i)}{\sum_{i=1}^n \mu_R(x_i)} \tag{4}$$

where n is the number of inputs, x_i is the i -th input, $\mu_R(x_i)$ is the membership value of the fuzzy set R , resulting from the composition rules of the inferential step, referred to x_i .

For this work we designed a very simple fuzzy system made of one simple rule measuring the reliability of the NN estimate. Indeed, analysis performed on the NN method showed that it is mostly reliable when the altitude difference between the requested locality and the chosen one is *little*. Therefore, the inference engine has just one (following) simple rule:

IF the NN is *reliable* THEN $y = \bar{Y}_{NN}$
 ELSE IF the NN is NOT *reliable* THEN $y = \bar{Y}_{ANN}$

where \bar{Y}_{NN} and \bar{Y}_{ANN} are the estimation provided by the NN and ANN modules. Then, for the fuzzification step we may define the following fuzzy set

F_{nn} = “Altitude difference is *little*”
 whose membership function is defined as

$$\mu_{F_{nn}}(d_h) = \begin{cases} \text{if } d_h < \delta & \text{then } 1 \\ \text{else } e^{-\frac{(d_h - \delta)^2}{2\sigma^2}} & \end{cases} \tag{5}$$

$$d_h = |h - h'| \tag{6}$$

where h is the altitude of the requested locality and h' is the altitude of its nearest neighbour.

Since the proposed method is very simple, fuzzy inference is actually not used, therefore the output is a simple fuzzy weighted evaluation.

$$\text{Output} = \mu_{F_{nn}}(d_h) \bar{Y}_{NN} + (1 - \mu_{F_{nn}}(d_h)) \bar{Y}_{ANN} \tag{7}$$

3 Experimentation

Experiments focussed on the estimate of the monthly ambient temperature of three different Italian selected areas using data taken from [1]. At present not all the Italian areas have been tested but we performed the experimentation on three areas representative of northern, central and southern Italy. In particular, we tested the following areas :

- North : Trentino AA – Veneto – Friuli VG – Emilia R (TVFE)
- Centre : Marche-Abruzzo (MA)
- South : Sicily

For each area the data have been partitioned into train and test set. In the following tables we show the results of blind testing on the latter. In two cases out of three (TVFE and MA) to achieve better accuracy we had to split the data into two main seasons: summer and winter. The tables report the average and, in brackets, the maximum absolute error made by each approach (NN=Nearest Neighbour, ANN=Artificial Neural Networks, HYB=Hybrid) in Celsius degrees. Moreover, as a benchmark, we show the theoretical error (TH) that an ideal judge, with the perfect knowledge of the error committed by each model in each point, would get.

The TVFE data set was composed by 180 localities which were split into 162 and 18 for training and testing. The MA data set was composed by 56 localities which

were split into 43 and 13 for training and testing. The Sicily data set was composed by 52 localities which were split into 47 and 5 for training and testing.

Although the data sets we used are not very large, at present they are the most complete available in Italy. We think that in the near future these data sets will be expanded in order to include more localities.

Table 1. Results on the TVFE area

	NN	ANN	HYB	TH
Summer	1.7 (6.8)	0.68 (1.82)	0.63 (1.74)	0.55 (1.74)
Winter	1.24 (6.8)	0.92 (3.45)	0.73 (1.92)	0.59 (1.92)
Year	1.47 (6.8)	0.78 (3.45)	0.73 (2.4)	0.54 (1.99)

Table 2. Results on the MA area

	NN	ANN	HYB	TH
Summer	1.03 (4.2)	0.82 (2.92)	0.53 (1.3)	0.46 (1.28)
Winter	1.02 (3.8)	0.73 (2.02)	0.57 (1.6)	0.46 (1.52)
Year	1.02 (4.2)	0.77 (2.21)	0.57 (2.21)	0.45 (2.20)

Table 3. Results on Sicily

	NN	ANN	HYB	TH
Year	3.11 (9.1)	0.66 (1.87)	0.56 (1.21)	0.55 (1.21)

Experiments show the effectiveness of the proposed hybrid approach. In fact it always outperforms the best of the two single models (NN and ANN) in terms of average and what is more relevant, in terms of maximum absolute estimation error which is getting very close to the theoretical lower limits.

From the tables above we point out that the average error is about 0.6 degrees and the maximum is always less than 2 degrees.

Moreover, as example we report two graphs comparing the real temperature (Treal) to the estimated one given by the proposed hybrid approach (Thyb).

In table 4 we report the values of the parameters of eq. (5) for each of the three areas which we performed the experimentation on.

Table 4. Eq. (5) parameters

	δ	σ
TVFE	winter=80 summer=40	20
MA	winter=250 summer=90	20
Sicily	160	20

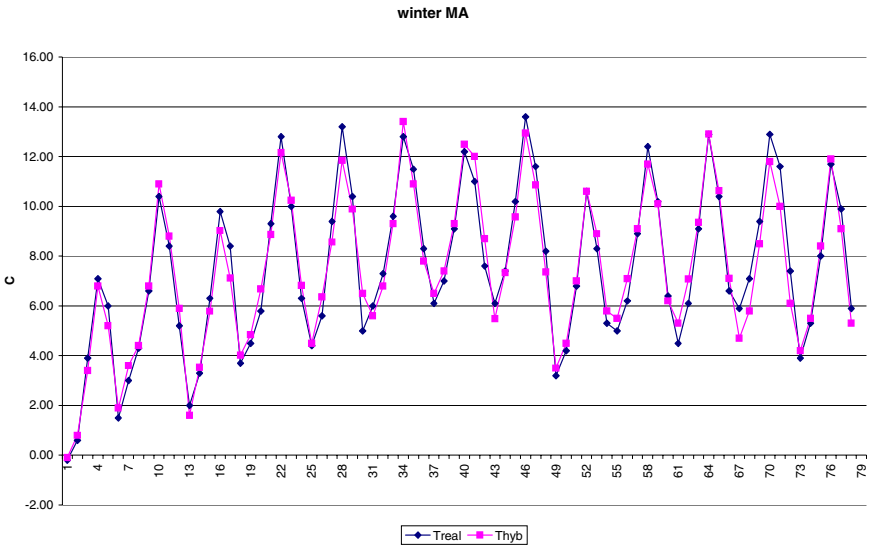


Fig. 3. Winter testing results on the MA area

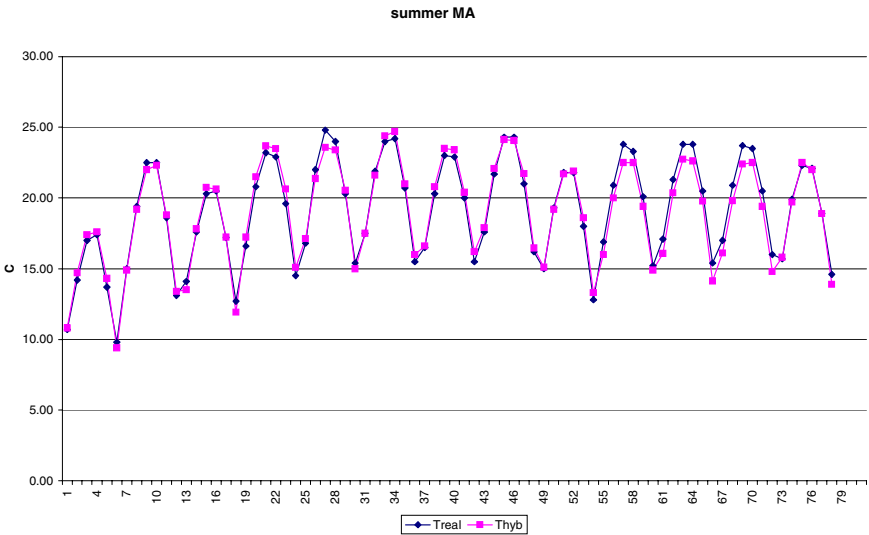


Fig. 4. Summer testing results on the MA area

4 Conclusion

In this paper we showed a novel hybrid approach based both on traditional and soft computing techniques to estimate ambient temperature for those places where such

datum is not available. Indeed, we combined neural networks with the nearest neighbour algorithm; we used a fuzzy logic decision maker and compared the results of the single techniques to the hybrid one. Experimentation was performed on three Italian areas and results showed the effectiveness of the proposed hybrid approach. The hybrid model always outperformed the best out of the two single models in terms of average and maximum absolute estimation error, achieving about 0.6 degrees of average error and less than 2 degrees as maximum, being very close to the theoretical lower limits of the perfect judge.

Future work will expand the test on all Italian areas and different modelling methods, like Radial Basis Networks and Support Vector Machines, will be compared and/or added to the current framework.

References

1. Atlante Italiano della Radiazione Solare, <http://www.solaritaly.enea.it/>
2. Babuska, R.: Fuzzy modelling: principles, methods and applications. In: International Summer School on Fuzzy Logic Control: Advances in Methodology, Ferrara (1998)
3. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
4. Erbs, D.G., Klein, S.A., Beckman, W.A.: Estimation of degree-days and ambient temperature bin data from monthly-average temperatures. *Ashrae Journal*, 60–65 (1983)
5. Haykin, S.: Neural Networks, a comprehensive foundation, 2nd edn. Prentice Hall, New Jersey (1999)
6. Klein, S.A., Beckman, W.A., Mitchell, J.W., Duffie, J.A., Duffie, N.A., Freeman, T.L., Mitchell, J.C., Braun, J.E., Evans, B.L., Kummer, J.P., Urban, R.E., Fiksel, A., Thornton, J.W., Blair, N.J., Williams, P.M., Bradley, D.E.: TRNSYS, a Transient System Simulation Program. Solar Energy Lab, Univ. of Wisconsin-Madison (2000)
7. Kosko, B.: Neural networks and fuzzy systems. Prentice-Hall, Englewood Cliffs (1992)
8. Mitra, A.K., Nath, S.: Forecasting maximum temperatures via fuzzy nearest neighbour model over Delhi. *Appl. Comput. Math.* 6(2), 288–294 (2007)
9. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. *IEEE Trans. Neur. Netw.* 1(1), 4–27 (1990)
10. Nikravesh, M., Farrell, A.E., Stanford, T.G.: Model identification of non linear time variant processes via artificial neural network. *Computes and Chemical Engineering* 20(11), 1277–1290 (1996)
11. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by backpropagating errors. *Nature* 323, 533–536 (1986)
12. Şen, Z.: Fuzzy algorithm for estimation irradiation from sunshine duration. *Solar Energy* 63(1), 39–49 (1998)
13. Sodoudi, S.: Estimation of temperature, precipitation and evaporation with Neuro-Fuzzy method. In: Workshop of statistical downscaling, Oslo (2005)
14. Tatli, H., Sen, Z.: A new fuzzy modeling approach for predicting the maximum daily temperature from a time series. *Journal of Engineering and Environmental Science* 23, 173–180 (1999)
15. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)
16. Zimmerman, H.J.: Fuzzy set theory. Kluwer Academic, Boston (1991)

Providing Dynamic Instructional Adaptation in Programming Learning

Francisco Jurado¹, Olga C. Santos², Miguel A. Redondo¹, Jesús G. Boticario²,
and Manuel Ortega¹

¹ Computer Science and Engineering Faculty
University of Castilla-La Mancha

Paseo de la Universidad, 4 – 13071 Ciudad Real – Spain

{Francisco.Jurado,Miguel.Redondo,Manuel.Ortega}@uclm.es

² aDeNu Research Group. Computer Science School. UNED

Juan del Rosal, 16 – 28040 Madrid – Spain

{ocsantos,jgb}@dia.uned.es

Abstract. This paper describes an approach to create an Intelligent Tutoring System that provides dynamic personalization and learning activities sequencing adaptation by combining eLearning standards and Artificial Intelligent techniques. The work takes advantage of the functionalities provided by an open source Learning Management System, dotLRN, which supports eLearning standards such as IMS-LD, and generates personalized sequences of learning activities. Moreover, the user model draws on standards such as IMS-LIP and IMS-AccLIP and the personalized learning path provided to the user is enriched with feedback coming from various Agents. In turn, the agents apply Fuzzy Logic to evaluate the students' assignments and to update the user model with their preferences by means of machine learning techniques.

1 Introduction

Programming learning implies that students of computer science should acquire and develop several competences. Students must solve problems presented strategically by the teacher applying the notion of *trial and error*. Solving several kinds of problems will make students acquire the abilities and aptitudes they need to solve the real problems in their future labour life.

Students of programming algorithms use computers for developing the learning activities specified by the teacher.. This makes it an ideal environment for Computer Assisted Learning (CAL), where students are assisted by an Intelligent Tutoring System (ITS) that guides them in their learning process [1], helping them to improve and to acquire the abilities and aptitudes they should acquire and develop, reducing the need for the slow *trial and error* process.

An ITS allows adapting the learning process to each student. For this, ITSs focus on determining which the student cognitive model is by knowing which parts of the domain he/she knows. Thus, ITSs can infer the next learning activity for each specific student. With that approach an ITS for programming learning algorithms in Computer

Science is proposed in [2]. In that work, a set of functional services are outlined, taking into account eLearning standards and several Artificial Intelligent techniques. Instructional adaptation is provided by merging eLearning specifications, allowing communication, integration and interoperability with other eLearning systems, with Artificial Intelligent techniques for enabling ITS , so that students can achieve the abilities and aptitudes they need for their future work.

The paper is structured as follows: First, we will introduce the framework in which the work is based (section 2). Second, we outline how to provide adaptation to the student profile (section 3). Third, we will describe the way in which the learning flow is adapted to lead the students to reach the learning objectives by combining eLearning standards and Fuzzy Logic (section 4). And finally, we will outline some concluding remarks and future works (section 5).

2 Framework: ADAPTAPlan Project

The work presented in this paper is being developed within ADAPTAPlan project [3]. This project focuses on the automatic generation of standard-based adaptive learning routes by considering some initial requirements from the course author and the meta-data that characterizes the learning objects. The system uses a combination of planning techniques and user modelling techniques to lead the students through learning activities according to their specific characteristics.

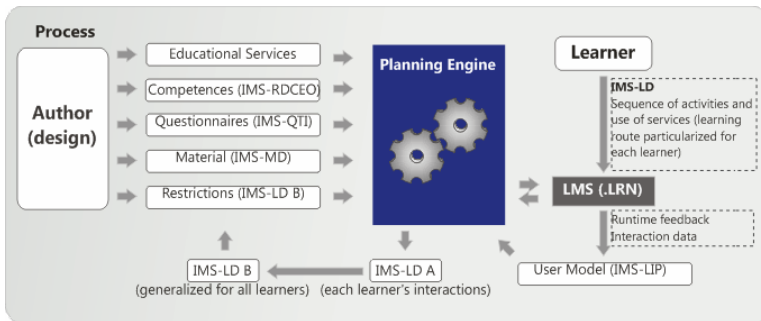


Fig. 1. ADAPTAPlan architecture [3]

As an application domain to work with, the research presented in this paper focuses on helping learners to create programming algorithms. Processing the student developed algorithms as solution to their assignments is an essential input for planning the subsequent learning tasks. This allows the system 1) to assess the degree in which the objectives are being achieved in the learning process, 2) to determine whether there is a need for re-planning the activities, and 3) to adapt the learning process to the student's specific cognitive characteristics. Thus, Figure 1 shows the ADAPTAPlan architecture [3], where the planner is the centrepiece. Once a set of requirements about the course (e.g. learning objects/ materials, questionnaires, restrictions, competences, educational services, etc.) have been given by the teacher, the planner is

responsible for determining the sequence of activities where the previous requirements fit for each student to achieve the specific learning objectives. The activities plan is delivered as an IMS-LD unit of learning [4]. The interactions of the learner with the learning management system (LMS) feed back the planner, indicating the degree in which the objectives are being achieved. If those objectives are not being fulfilled by a particular student, the planner generates a new plan accordingly.

Groups such as IEEE LTSC, IMS Global Learning Consortium or ADL, work on standards allowing reusability and interoperability into the eLearning industry, setting the standard base where engineers and developers must work for getting eLearning systems integration. ADAPTAPlan uses dotLRN as open source LMS since it supports these standards (e.g. IMS-LD [4], IMS-QTI [5], IMS-CP [6], IEEE-LOM [7], IMS-MD [8] and SCORM [9]) and provides the necessary features for users' managements, instructional design deployment, collaboration features, etc. Furthermore, to interact with external tools, it uses Web Services as engine [10].

We aim at providing the planner with valuable feedback in order to allow the best adaptations. In next sections, we will present how the user modelling is supported in this approach by analyzing the student actions to adapt the way in which the students interact with the system, and how fuzzy logic techniques can be used to evaluate the student's assignments in a semi-automatic way to allow an adaptation in the learning flow, leading the student through a personalized activity sequence.

3 Student Profile Adaptation

Multi Agent Systems (MAS) are commonly used to maintain and update a user model to generate dynamic adaptations to relevant users' characteristics due to their flexibility for combining different proposed solutions [11]. In ADAPTAPlan, the user model is described in terms of IMS-LIP [12] and IMS-AcclIP [13] specifications. Moreover, a multi-agent architecture is built in order to monitor the performance of the learners in the system and constantly to update their user model. This architecture is called ADA+ and is described elsewhere [14]. It offers some autonomous entities (agents) who, acting on a coordinate way, generate recommendations (taking into account the individual user features and the context) on what to do during the course in those situations in which the IMS-LD defined for the learner does not cover the learner's needs. The generation of these recommendations takes as input the information about the course structure in IMS-LD and the learners' interactions. In particular, it considers the following information to produce the adaptations [3]: a) Felder's Learning styles, b) Collaboration level and c) Knowledge level.

A learning style is defined as characteristic strengths and preferences in the ways people take in and process information [15], and it determines the unique way of learning for each student. Felder's Learning Style Inventory consists of a 44 questions test. From the user responses, the four dimensions of Felder learning style (i.e. processing, perception, input and understanding) are obtained.

The collaboration competence of a learner in a course is computed taking into account the usage of the course services, such as forums, shared files, comments, etc. This competence is learnt by ADA+ and is used to establish a relationship of the user behaviour in the course with his/her success of the learning process. The definition of

this collaborative competency takes six levels, in an incremental manner. The learner shifts from Non_Collaborative to Communicative, as she/he progresses on collaborative tasks until the highest appreciated level (Useful) is achieved (see [3]).

The knowledge level associated to a learning objective within the course is computed through the analysis of the learner interactions with the learning objects and activities, and the evaluations results obtained from test, questionnaires or other evaluation tasks. In [3], the knowledge level of a learner within an objective is described by one attribute taking one of the possible values: novice, average or expert.

With these user features in mind, it is possible to link them together with learning objects and resources, to be integrated into an IMS-LD unit of learning, as described in [3].

4 Instructional Adaptation: Combining IMS-LD and Fuzzy Logic

In this section we will describe how we can combine eLearning standards and Fuzzy Logic to provide a mechanism that allows the students to be led through a personalized learning activity flow as a consequence of their solution delivered to each assignment.

4.1 IMS-LD as Scripting Language for Instructional Adaptation

Since our aim is to develop an ITS that allows to apply instructional strategies according to the subject to learn/teach, we use IMS-LD [4] for specifying the method that must be used in the teaching/learning process, that is, for specifying the instructional adaptation model. The IMS-LD specification runs in dotLRN, receiving the feedback from the execution environment.

IMS-LD represents instructional strategies using the theatre metaphor. Hence, an instructional *method* is divided in *play* elements containing several *acts* where different *roles* bring *activities* in specific *environments* in their *role-parts*. These activities can be classified into three kinds: 1) *learning-activities* that lead the learner to get the knowledge; 2) *support-activities* that do not contribute to the learning process itself, but are needed for the success of learning activities; 3) *structure-activities* that allow to structure learning activities, support-activities or other structure-activities in sequence or selection order. All these activities can be brought into specific *environments*. These environments have *learning objects* and *services* that can be used by the different roles in the activities.

IMS-LD can be used for developing Adaptive Learning (AL) [16]. The learning design can be enriched with a collection of properties from the student profile. These properties allow to specify conditions which can determine if a learning activity or a branch of the learning flow (a set of learning activities) is shown or hidden to a specific student. This can be done in the following way: each *method* has a section for defining *conditions* that points how it must be adapted to specific situations through rules.

In our system, the variables used for defining the adaptation rules in the condition section of an instructional method consider the learning styles, the collaboration level and the knowledge level, which is obtained by evaluating the algorithm developed by the student as solution to his/her assignment.

```

IF student::(Knowledge, less-than, 5)
    THEN hide activity A1 and show activity A2
    ELSE show activity A1 and hide activity A2
    
```

Fig. 2. Example of rule for adapting an instructional design

In figure 2, a rule is shown as an example: the rule considers the results from the evaluation algorithm: “if the student evaluation for a task is less than 5, then activity A1 is hidden and activity A2 is shown, in the opposite case, activity A1 is showed and activity A2 is hidden”. In the next subsection we will explain how to evaluate the algorithms that the students design as a result of the learning activities.

4.2 Evaluating the Student Algorithm to Feedback

For evaluating the student algorithm developed as solution to an assignment, we use the method proposed in [17] In brief, the objective is getting a tool that allows to compare the algorithm developed by the student with that specified by the teacher. It provides a way for representing the *approximate ideal algorithm* that the teacher estimates for solving a certain problem using Fuzzy Logic. Next, the algorithm that the student has written is compared with that *approximate ideal representation*.

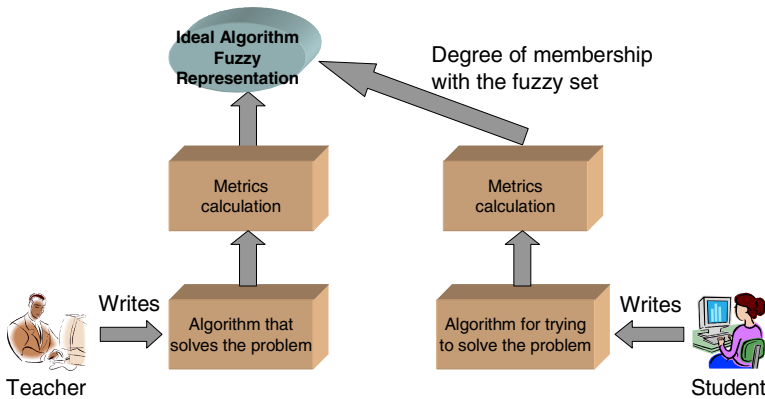


Fig. 3. Evaluating the student algorithm

The evaluation process is summarized in figure 3. In this figure, the teacher writes an implementation for the *ideal approximate algorithm* that solves a problem (at the figure lower-left-hand corner). Next, several software metrics shaping its functionality are calculated. That provides an *instance* of the *ideal approximated algorithm*. After that, the fuzzy set for each metric is established in the next way: initially, each fuzzy set will be a default trapezoidal function around the metric value from the *approximate algorithm*; the teacher can evaluate first students’ algorithms indicating whether an algorithm is not correct, little correct or correct; thus, each fuzzy set will be adapting and deforming itself. With all that, we obtain a collection of fuzzy sets that

characterizes the algorithm. Thus, we get a *fuzzy representation* of that *ideal approximated algorithm*, that is, we obtain an *ideal approximated algorithm fuzzy representation* that solves a concrete problem (at the top of the figure 2). Algorithms that students have written (on the right of the figure) will be correct if they are *instances* of that *ideal algorithm fuzzy representation*. Knowing the degree of membership for each software metric, obtained from the algorithm wrote by students in the correspond fuzzy set for the *ideal approximated algorithm fuzzy representation*, gives us an idea of how good is the algorithm that the students have developed.

As working environment we use the widely available Eclipse platform (<http://www.eclipse.org/>). Eclipse is an integrated development environment that works with Java, C/C++ and other programming languages, which allows creating extensions by using its own API. Such extensions are implemented as plug-ins that can be optionally loaded by users. We have developed a plug-in for the Eclipse environment to be able to do the described evaluation process and to communicate with other software entities.

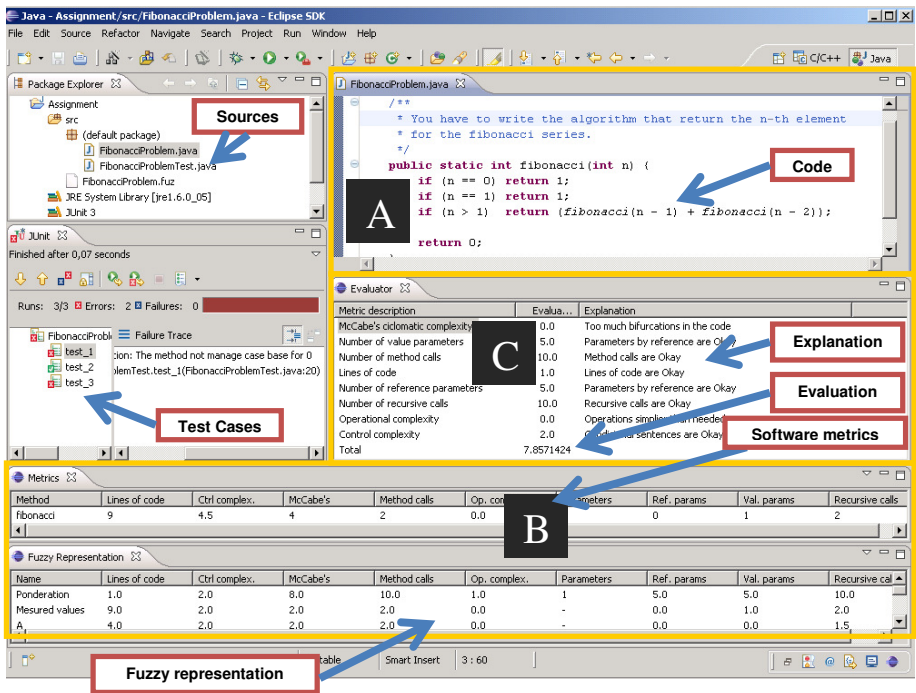


Fig. 4. The students' environment for developing the algorithm

Figure 4 shows the environments that the students use for developing the assignments the system proposes: in zone A we can see the code the student has developed as a solution for a concrete assignment; zone B shows the fuzzy loaded representation, used to do the evaluation, and the software metrics obtained from the code that is evaluated; in zone C the result of the evaluation is shown providing a bit explanation

about the evaluation to notify to the student a tip about what is wrong, and providing the total evaluation that the system will take into account to determinate the next learning activity.

This evaluation functionality will interact with IMS-LD unit of learning. For this, an activity of the course design should consist in developing a programming algorithm. When the learning design is running, if the properties evaluated for the user model (e.g. learning style, knowledge and collaboration level) consider that the user is ready to perform this activity, dotLRN IMS-LD player will indicate it to our customized eclipse framework and the user will work in the assignment. When the student finishes, the evaluation is performed and the result obtained stored, thus available for next runs in IMS-LD as well as for other adaptations in the platform.

5 Concluding Remarks and Future Works

In this paper we have described a standard-based approach to provide dynamic instructional adaptation in Learning of Programming. We have shown a system that combines eLearning standards with different Artificial Intelligent Techniques within the ADAPTAPlan framework. In particular, three components are integrated: 1) a user model, following IMS LIP and IMS-AccLIP specifications, whose attributes can be learned by machine learning techniques; 2) an eclipse-based framework that evaluates the algorithms developed by students as solutions to their assignments; and 3) an open source standard-based LMS called dotLRN with an IMS-LD player.

In ADAPTAPlan, the teacher can set several requirements about a course, then a planner proposes a sequence of learning activities specified in IMS-LD and optimized for a particular student. The interactions of students in the LMS, as well as the solutions they deliver for the assignments feedback, are the basis to carry out a better adaptation. To this, the system follows a twofold strategy. Firstly, a Multi Agent System is used to maintain and update the user model so as to generate dynamic adaptations to relevant users' characteristics. Secondly, a technique based on Fuzzy Logic is used to analyze and evaluate the algorithms the students have developed as solution to the assignment. This evaluation is used together with the IMS-LD specification to allow the students to be led through a personalized learning activity flow as a consequence of their solution delivered to each assignment.

With all this, we have a system that supports the adaptation of the learning process to each student. Starting from a set of specifications about a concrete course, the system provides a first approximation to an IMS-LD specification and reacts to the students' behaviour and performance. In our next steps, we must check if the system performs the correct adaptations. Comparing the adaptations done automatically by the system with those suggested by a teacher will indicate how good the adaptation that the system provides is.

Acknowledgements

This research work has been partially supported by the Junta de Comunidades of Castilla-La Mancha (Spain) [projects AULA-T (PBI08-0069) and M-GUIDE (TC20080552)] and Ministerio de Educación y Ciencia (Spain) [projects E-Club-II (TIN2005-08945-C06-04), ADAPTAPlan (TIN2005-08945-C06-00)].

References

1. Murray, T.: Authoring Knowledge Based Tutors: tools for Content Instructional Strategy, Student Model and Interface design. *Journal of the Learning Sciences* 7(1), 5–64 (1998)
2. Jurado, F., Redondo, M.A., Ortega, M.: An architecture to support programming algorithm learning by problem solving. In: Corchado, E., Corchado, J.M., Abraham, A. (eds.) *Innovations in Hybrid Intelligent Systems, Proceedings of Hybrid Artificial Intelligent Systems (HAIS 2007)*, pp. 470–477. Springer, Spain (2007)
3. Baldiris, S., Santos, O., Barrera, C., Boticario, J.G., Velez, J., Fabregat, R.: Integration of Educational Specifications and Standards to Support Adaptive Learning Scenarios in ADAPTAPlan. Sp. Issue on New Trends on AI techniques for Educational Technologies. *International Journal of Computer Science and Applications (IJCSA)* (January 2008)
4. IMS-LD: IMS Learning Design. Information Model, Best Practice and Implementation Guide, XML Binding, Schemas. Version 1.0 Final Specification, Technical report, IMS Global Learning Consortium Inc. (2003), <http://www.imsglobal.org/learningdesign/index.cfm>
5. IMS Question and Test Interoperability. Version 1.2.1 Final Specification (2003)
6. IMS Content Packaging Specification. v1.1.4 final specification (2004)
7. Learning Technology Standards Committee. Standard for Learning Object Metadata Final version 1.2 (2002)
8. IMS Metadata 1.2.1 Final Specification (2001)
9. SCORM 2004. Overview, Technical report, ADL (Advanced Distributed Learning) (2004), <http://www.adlnet.gov/scorm/index.cfm>
10. Santos, O.C., Boticario, J.G., Raffenne, E., Pastor, R.: Why using dotLRN? UNED use cases. In: *FLOSS (Free/Libre/Open Source Systems) International Conference* (2007)
11. Wooldridge, M.: *An Introduction to Multiagent Systems*. John Wiley and Sons Ltd, Chichester (2002)
12. IMS Learner Information Package. Version 1.0 Final Specification (2001)
13. IMS Learner Information Package Accessibility for LIP. Version 1.0 Final Spec. (2003)
14. Santos, O.C., Baldiris, S., Velez, J., Boticario, J.G., Fabregat, R.: Dynamic Support in ADAPTAPlan: ADA+. In: Borrajo, D., Castillo, L., Corchado, J.M. (eds.) *CAEPIA 2007. LNCS (LNAI)*, vol. 4788, pp. 131–140. Springer, Heidelberg (2007)
15. Felder, R.M., Silverman, L.K.: Learning and Teaching Styles In Engineering Education. *Engr. Education* 78(7), 674–681 (1988)
16. Towel, B., Halm, M.: Learning design: A handbook on modelling and delivering networked education and training. In: *Designing Adaptive Learning Environments with Learning Design*, ch. 12, pp. 215–226. Springer, Heidelberg (2005)
17. Jurado, F., Redondo, M.A., Ortega, M.: Fuzzy algorithm representation for its application in Intelligent Tutoring Systems for the learning of programming, in Rogério PC do Nascimento. In: Gerqia, A., Serendero, P., Carrillo, E. (eds.) *EuroAmerican Conference On Telematics and Information Systems, EATIS 2007 ACM-DL Proceedings*. Association for Computing Machinery. ACM, Faro (2007)

Modelling Radial Basis Functions with Rational Logic Rules

Davide Sottara and Paola Mello

Department of Electronics, Computer Science and Systems
Faculty of Engineering, University of Bologna
Viale Risorgimento 2, 40129, Bologna (BO) Italy
{davide.sottara2,paola.mello}@unibo.it

Abstract. Connectionist systems such as Radial Basis Function Neural Networks and similar architectures are commonly applied to solve problems of learning relations from available examples. To overcome their limits in clarity of representation, they are often interfaced with symbolic rule-based systems, provided that the information they have memorized can be interpreted. In this paper, an automatic implementation of a RBF-like system is presented using only gradual fuzzy rules learned by induction directly from training data. It is then shown that the same formalism, used with type-II truth values, can learn second-order, fuzzy relations.

Keywords: Radial Basis Function, Fuzzy Logic, Induction.

1 Introduction

Many real world problems can be formalized as finding the optimal general relation between a “data” domain X and an “answer” range Y : solving classification problems requires to find the boolean characteristic function of a tuple of features with respect to some target class; decision and control problems consist in mapping states to actions by means of policies; forecasting is the act of estimating of the future state of a system given the actual one and so on.

In many cases the ideal relation is too complex to be computed – and sometimes it is not even a relation, e.g. when alternative responses exist - so it is necessary to approximate it with a simpler model, usually local in the sense that its accuracy is guaranteed only for some parts of the domain X . In the context of Artificial Intelligence these models are defined – or at least tuned in their parameters – by means of machine learning techniques, which generalize a relation from a set of training examples.

The existing approaches are usually divided in “symbolic” and “connectionist”, depending on whether they represent the knowledge using human-readable encodings – such as in Rule-Based Systems – or not, like Neural Networks. The latter are generally more adaptable, but less understandable due to the non evident way the process the information. The former, instead, are preferred when their modularity and clarity can be exploited.

Several hybrid systems have been developed combining modules of different type in order to get the best of both worlds, while overcoming the individual limitations:

the degree of interaction between the modules ranges from loose coupling to full integration, but specific interfaces are needed to allow communication. Unified systems do not have this recoding requirement, but are commonly made integrating connectionist modules.

In this work, instead, a module usable in both hybrid and unified symbolic systems is proposed. Many systems learn relations using Radial Basis Function Neural Networks and then try to extract rules from the trained network (e.g. [1]): in the following sections, instead, similar results are attained generating and learning gradual rules [2] directly.

2 Radial Basis Function Networks

RBFs have been designed to solve problems of exact interpolation of multi-dimensional data points: given a set of inputs $\mathbf{x}_{j:1..N}$, and a set of targets \mathbf{t}_j , the goal is to find a function h such that $\forall j: h(\mathbf{x}_j) = \mathbf{t}_j$. The RBF approach [3] uses N basis functions in the form $\phi(\|\mathbf{x}-\mathbf{x}_j\|)$, depending on the distance $\|\mathbf{x}-\mathbf{x}_j\|$ between points of the domain and reference inputs. The output is usually a linear combination:

$$h(x) = \sum_j w_j \cdot \phi(\|\mathbf{x} - \mathbf{x}_j\|). \quad (1)$$

The interpolation constraints can be written in matrix form: $\Phi \mathbf{w} = \mathbf{t}$, leading to the exact solution $\mathbf{w} = \Phi^{-1} \mathbf{t}$ when all the inputs \mathbf{x}_j are different and Φ is non-singular. Several types of basis functions have been used in literature, all being nonlinear, unimodal, localized functions: the most common is the Gaussian one [3]

$$h_j(x) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_j\|^2}{\sigma^2}\right). \quad (2)$$

Others exists, such as the triangular:

$$h_j(x) = \max\{0, 1 - \sigma\|\mathbf{x} - \mathbf{x}_j\|\}. \quad (3)$$

The model, known as Radial Basis Function (Neural) Network is obtained relaxing the exact interpolation constraints:

1. The number M of basis functions is much less than N
2. The basis centres \mathbf{x}_j are no longer required to be input points

In this case, the matrix Φ is no longer square and an exact solution may not exist, but the optimal solution, in a least-square error sense, is obtained computing the pseudo-inverse matrix Φ^+ and taking $\mathbf{w}_{\text{opt}} = \Phi^+ \mathbf{t}$.

This model requires that the centers \mathbf{x}_j are set, usually during a previous, independent training stage: for this, various clustering and data analysis techniques have been used, from data clustering to Self-Organizing Maps [3].

3 Rational Logic

Fuzzy logic [4] is an extension of first-order predicate logic which adds vagueness to a predicate P annotating each statement $P(\mathbf{x})$ with a truth value ε which models the

degree of membership of the arguments \mathbf{x} in the set defined by P . Formally, a membership function $\mu : \mathbf{X} \rightarrow L$ is defined to map arguments in the domain \mathbf{X} to truth values defined over a set L . Hence the notation:

$$P(\mathbf{x})_\varepsilon \leftrightarrow \varepsilon = \mu_P(\mathbf{x}) \tag{4}$$

Given a particular choice of L , the operators ($\wedge, \vee, \neg, \dots$) and the quantifiers (\forall, \exists) have to be defined accordingly: in the simplest case $L \equiv \mathfrak{R}$, but other alternatives exist such as intuitionistic fuzzy logic, where truth values are actually intervals ($L \equiv \mathfrak{R} \times \mathfrak{R}$). In this work, they will be modeled using multimodal discrete type-II fuzzy sets [5], which represent vague and uncertain truth values. The truth interval $[0,1]$ is divided into N slots: each slot $\varepsilon[j]$ has an associated weight β_j , so that $\varepsilon[j] = \beta_j / \sum_k \beta_k$ is the degree at which $\varepsilon[j]$ represents the actual truth value. From a probabilistic point of view, the weights can be considered the hyperparameters of a multinomial distribution [6] denoting the likelihood of a given truth value, with the maximum being at $\varepsilon[j]$. This representation includes the intuitionistic one, by observing that a truth interval $[\tau, 1-\varphi]$ can be obtained setting $\tau = \varepsilon[j_*]$ and $\varphi = \varepsilon[j^*]$. j_* (resp. j^*) is the maximum (resp. minimum) index such that all the previous (resp. following) $\varepsilon[j]$ are below a given threshold: $j_* = \arg \max_k (\min_k) : j < (>) k \rightarrow \varepsilon[j] < \varepsilon_0$.

3.1 Logic Operators

In fuzzy logic, there is not a unique definition for the logic operators as in standard logic. Instead, there is a class of fuzzy logics whose instances are different in the way they coherently define the concepts of t-norm $*$ (strong conjunction), its conjugate norm \Rightarrow (logic implication) and negation \neg . For the reasons outlined in [7], in the following sections, Pavelka’s linear operators, some of which are briefly recalled below (for more details see [4] and [8]), will be adopted. The definitions generalize to intervals and can be applied to type-II fuzzy truth values using Dubois and Prade’s extended composition principle [2].

- Strong And $\otimes P_{i:1..n}$ $\varepsilon_\otimes = \max \{ 0, \sum_i \varepsilon_i - (n - 1) \}$
- Negation $\neg P$ $\varepsilon_\neg = 1 - \varepsilon_P$
- Implication $P \rightarrow C$ $\varepsilon_{\rightarrow} = \min \{ 1, 1 - \varepsilon_P + \varepsilon_C \}$
- Equivalence $P \equiv Q$ $\varepsilon_\equiv = | \varepsilon_P - \varepsilon_C |$

3.2 Fuzzy Inference

Fuzzy rule bases usually have the form of a list of implications of the form “if- then”:

$$\begin{aligned} \mathbf{X} \text{ is } P &\rightarrow \mathbf{Y} \text{ is } C \\ \mathbf{X} \text{ is } Q &\rightarrow \mathbf{Y} \text{ is } D \\ &\dots \end{aligned}$$

The basic inference mechanism is Modus Ponens, according to the extended schema which states that a partial match in the premise leads to a partial match in the expected conclusions: $\langle P'(\mathbf{X}), P(\mathbf{X}) \rightarrow C(\mathbf{Y}) \rangle / C'(\mathbf{Y})$.

This schema is commonly applied to fuzzy sets as well as to fuzzy predicates. The truth value of the conclusion is computed using the t-norm:

$$\epsilon_C = \epsilon_P * \epsilon_{\rightarrow} \tag{5}$$

In rational logic, however, it must be considered that the result of MP is not necessarily the truth value of the conclusion, but only a lower bound [4]: in fact, an interval can be constructed by deducing the conclusion and its negation:

$$\tau_C = \epsilon_P \otimes \epsilon_{P \rightarrow C} \qquad \varphi_C = \epsilon_P \otimes \epsilon_{P \rightarrow \neg C} \tag{6}$$

When using type-II truth values, MP gives a distribution for τ , which can be transformed into an estimate distribution for ϵ using one of the techniques in [5]. Finally, when multiple implications lead to the same conclusions, the results must be combined: Dempster-Shafer’s combination rule can be used for type-II distributions:

$$\epsilon_{i \cap j}[j] = \epsilon_i[j] \cdot \epsilon_j[j] / \sum_k \epsilon_i[k] \cdot \epsilon_j[k] \tag{7}$$

It is the natural extension of the intersection of two intervals which, in turn, is an extension of the *max* composition principle for simple real truth values. Globally:

$$C(\mathbf{Y}) = \cap_i P_i(\mathbf{X}) \otimes P_i(\mathbf{X}) \rightarrow \{\neg\} C(\mathbf{Y}) \tag{8}$$

3.3 Induction

In general, the implication $P(\mathbf{X}) \rightarrow C(\mathbf{Y})$ can be fuzzy and, more importantly, can be learned by induction from a proper set of examples $\{\mathbf{x}_i^t, \mathbf{y}_i^t\}$. It is possible to compute \rightarrow_i directly for each given couple, but then the values must be combined and generalized. The safe and sound canonical approach requires that $\forall \mathbf{X}, \mathbf{Y}: P(\mathbf{X}) \rightarrow C(\mathbf{Y})$ but this generalizes poorly in practice: in order to comply to the definition of \forall , the minimum of the \rightarrow_i must be taken. While it does not ensure the soundness, since an input never seen before could lead to a lower implication, it is also unsafe as a single false exception is sufficient to invalidate any number of positive examples. Rather than the minimum, using a (weighted) average several yields several properties, other than being intuitive:

$$\epsilon_{\rightarrow}[j] = \sum_i w_i \cdot \epsilon_i^t[j] / \sum_i w_i \tag{9}$$

The training procedure can be incremental: if type-II fuzzy sets are used, it is equivalent to a Bayesian update of the available distribution after each new observation.

The weight of each observation should be equal to, or at least a non decreasing function of, the truth value of the premise P_i alone. This is a (fuzzy) measure of the relevance of a couple $\{\mathbf{x}_i^t, \mathbf{y}_i^t\}$ in determining the truth value of the implication: notice, in fact, that an implication is trivially true if the premise is false, which can lead to undesirable results. With this scaling, instead, falser premises are penalized.

Finally, considering an estimate of the global approximation error once ϵ_{\rightarrow} has been fixed equal to ϵ_{\rightarrow}^* :

$$E = \sum_i (y_i^t - x_i^t \otimes \epsilon_{\rightarrow}^*)^2 \tag{10}$$

The derivative with respect to ϵ_{\rightarrow}^* leads to the condition:

$$0 = \sum_i (y_i^t - x_i^t \otimes \epsilon_{\rightarrow}^*) \cdot \delta_{x_i^t > \epsilon_{\rightarrow}^*} \tag{11}$$

Since \otimes is linear or constant, depending on whether \mathbf{x} is greater than $-\epsilon^*_{\rightarrow}$ or not:

$$\sum_{i: x_i > -\epsilon^*_{\rightarrow}} \epsilon^*_{\rightarrow} = \sum_{i: x_i > -\epsilon^*_{\rightarrow}} (y_i^t - x_i^t + 1) \tag{12}$$

The optimal value for the induction is the average of the relevant couples. Unfortunately, ϵ^*_{\rightarrow} is not known: if its prior probability distribution is uniform in $[0,1]$, x_i^t is the probability that the i^{th} couple will be used for induction and so the weighted average over the training set is also the expected value of the optimal truth degree of the implication.

4 Emulating RBF-NNs within a Rule-Based System

The problem of extracting sensible rules from a trained neural network and its dual, to force a network to learn a given set of rules, is of great importance in the development of hybrid systems. The common approach involves a two-stage process: first, a NN is trained, then some interpretation algorithm is applied to extract the rules describing the learned knowledge. When the problem is learning a relation, RBF-NNs are convenient since, once the centers have been placed over the domain, the training is optimal and fast. In the rest of this work, starting from the formal similarity between equations (4) and (8) [9], a proper set of rules will be defined to emulate the training procedure and the outputs of a RBF-NN, even if with some important differences, forming a parallel between fuzzy sets and basis functions. The function to be approximated must be defined on a domain and a range compatible with a description in term of truth values, i.e. it must be a map

$$F : \mathbf{X} \subseteq [0,1]^m \rightarrow Y \subseteq [0,1] \tag{13}$$

For real-valued functions this may require a scaling and/or a translation in the original domain, while more complex domains may require other forms of fuzzification. When the domain involves multiple features ($m > 1$), they are typically aggregated, possibly using logical operators, so it can be assumed that $m = 1$ without losing in generality.

A single rule, implemented by a gradual implication, is the basic way of defining a relation between fuzzy inputs and outputs. For example, consider a relation between the level of the liquid in a tank, Full(Tank), and the volume of an alarm, Loud(Alarm). The rule

$$\text{Full(Tank)} \rightarrow \text{Loud(Alarm)}$$

can be read “the fuller the Tank, the louder the Alarm” and is useful when there is a direct (the exact form depends on the definition of \rightarrow) relation between the two quantities. A single global rule, however, is unlikely to model a more complex non-linear relation, so common controllers recur to a deeper level of detail, using several rules capturing the local behavior at different points in the domain, usually divided and described using terms such as “Very Low”, “Low”, and so on.

4.1 Basis Definition

The local rules correspond directly to different fuzzy sets or basis functions. Like in standard RBF-NNs, it can be assumed that the centers have been placed using some criteria: many standard fuzzy controllers deploy centers uniformly over the domain, while RBF-NNs typically cluster the data. Under the given assumptions, the centers correspond to particular truth values $\epsilon_{c:1..M}$, and the “distance” becomes the opposite of a logical equivalence, so each center defines implicitly a triangular basis function/fuzzy set by the formula: $P_c(X) := X \equiv \epsilon_c$.

The “scope” parameter σ has a similar logic interpretation: the operator \otimes can be applied n times to a predicate, yielding $\epsilon^n = \max \{0, 1 - n(1-\epsilon)\}$. The operator, altering the steepness of the membership function, model the so-called linguistic edges “very”, “more or less”, ..., as n , which need not be an integer, varies, so:

$$P_{c,\sigma}(X) := X \equiv^\sigma \epsilon_c \tag{14}$$

4.2 Training and Evaluation

For each basis a pair of rules is written in the form:

$$\begin{aligned} P_{c,\sigma}(X) &\rightarrow F(Y) \\ P_{c,\sigma}(X) &\rightarrow \neg F(Y) \end{aligned} \tag{15}$$

to obtain a lower and an upper bound for Y . Given the training set $\{\mathbf{x}_i^t, \mathbf{y}_i^t\}$, the $2M$ implications are learned by induction. Notice that inputs too far from the centre ϵ_c become irrelevant and so do not contribute to the final value of the implication. The learning of each rule also does not depend on the others.

The logic definition of the premises is not mandatory: in fact, any hybrid system capable of describing a property P with a truth degree can exploit this system (e.g. see [10]). After all the implications have been fixed, the function F can be reconstructed using MP and DS combination according to (8) to yield $F'(Y)$. The reconstruction error can be quantified at each point by the logic expression $\neg\{F(y) \equiv F'(y)\}$. A global descriptor E_{avg} , corresponding to the average error (distribution) can be computed by induction using uniform weights $w = 1/|Y|$. An even more synthetic result is obtained by taking the average:

$$E_{avg} = \frac{j}{N} \cdot \mathcal{E}[j]_{\neg\{F(Y) \equiv F'(Y)\}} \tag{16}$$

5 Experimental Results

In a first experiment, the induction framework has been tested on the reconstruction of elementary functions. 500 training points $\{\mathbf{x}_i^t, \mathbf{y}_i^t\}$ have been sampled uniformly from a Gaussian function $N(0.5,0.25)$ and a Linear function (slope = 3) clamped within the range $[0,1]$. The centers of $M = 1, 5$ and 9 basis rules have been placed uniformly over the domain $[0,1]$: each implication has been induced and the rules have been used to estimate the distributions $\mathbf{d}_i(\mathbf{x}^t) = [\tau_i(x), 1-\varphi_i(x)]$.

Figure 1 shows the basis position and the reconstructed results for $M = 5$ and 9 : the color is darker where the probability of locating the predicted output value is higher; Table 1 reports the average approximation error, compared to the RMSE of a canonic gaussian RBF-NN with 5 basis functions.

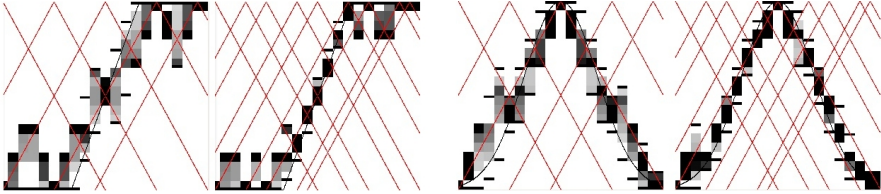


Fig. 1. Approximation of simple functions by $M=5,9$ rules ($\sigma = 2$)

Table 1. Approximation error for different functions and configurations

	M=1	M=5	M=9	RBF (M=5)
Linear	0.404	0.101	0.055	0.055
Gaussian	0.243	0.086	0.052	0.040

The advantage of using type-II truth values is that F needs not strictly be a function, but can be a generic (fuzzy) relation. A second test has been carried out to learn and predict the effect of seasonality on a chemical nitrification process (a wastewater treatment reaction [10]). Its duration varies from about 20 to 200 minutes, so the process has been monitored for two years, in the period from May to October. The data show a weak long-term trend, but also a high short term variance due to variations in the load which makes it difficult to predict the duration of the current process. In order to model the relation, the function approximation techniques have been applied. The data have been learned by a 5-neuron RBF-NN as shown in figure 2a: it captures the average trend (RMSE = 0.118) but can't memorize the local variability, which is not constant. A set of uncertain rules ($E_{\text{avg}} = 0.115$) of the same size, instead, associates a distribution to each input, yielding a range for the expected duration, instead of a point estimate.

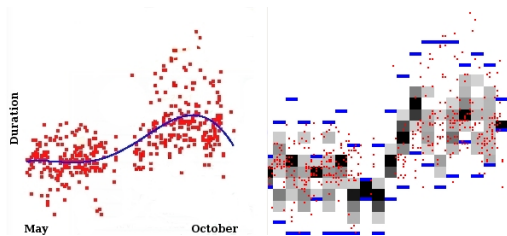


Fig. 2. Approximation of process duration data using a 5 neuron RBF-NN and 5 rules

6 Conclusions and Future Works

The method shows a way of learning relations while suggesting an implementation of a RBF-like module in the context of a fuzzy Rule-Based System. Unlike the RBF algorithm, where the output is globally optimal, the rules are learned locally and independently. This allows a graceful degradation of the performances: inputs not matching any premise lead to an “unknown” conclusion, i.e. are mapped into a uniform distribution over $[0,1]$ rather than on any particular value. The system, is suitable for automatic rule generation once the parameters (M , σ , C_i) have been fixed; compared to a canonical RBF-NN, it achieves similar performances with a slightly higher number of basis, with the error being influenced by the use of discrete values. A hybrid version of the learning algorithm, using a SOM with Gaussian activation functions, has already been applied to the recognition of the sub-phases of a chemical reaction from the analysis of indirect signals [10], in the context of a monitoring and diagnosis project where these techniques will be applied.

References

1. Li, W., Hori, Y.: An Algorithm for Extracting Fuzzy Rules Based on RBF Neural Network. *IEEE Transactions on Industrial Electronics* 51(4), 1269–1276 (2006)
2. Dubois, D., Prade, H.: *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, London (1980)
3. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Clarendon Press (1995)
4. Hajek, P.: *Metamathematics of Fuzzy Logic*. *Trends in Logic: Studia Logica Library*, vol. 4. Kluwer Academic Publishers, Dordrecht (1998)
5. Denooux, T.: Reasoning with imprecise belief structures. *IJAR: International Journal of Approximate Reasoning* 20 (1999)
6. Utkin, L.V.: Extensions of belief functions and possibility distributions by using the imprecise dirichlet model. *Fuzzy Sets and Systems* 154(3), 413–431 (2005)
7. Kundu, S., Chen, J.: Fuzzy logic or Lukasiewicz logic: a clarification. *Fuzzy Sets and Systems* 95, 369–379 (1998)
8. Vojtas, P.: Fuzzy logic programming. *Fuzzy Sets and Systems* 124(3), 361–370 (2001)
9. 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)
10. Sottara, D., et al.: Artificial Intelligence based rules for event recognition and control applied to SBR systems. In: *Conference Proceedings: SBR4 4th Sequencing Batch Reactor Conference, Rome, Italy, April 7-10 (2008)* (to be published)

On Combining Classifiers by Relaxation for Natural Textures in Images

María Guijarro¹, Gonzalo Pajares², and P. Javier Herrera²

¹ Centro Superior de Estudios Felipe II. Ingeniería Técnica en Informática de Sistemas,
28300 Aranjuez, Madrid, Spain
mgujarro@cesfelipesecondo.com

² Dpt. Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad
Complutense, 28040 Madrid, Spain
pajares@fdi.ucm.es

Abstract. One objective for classifying textures in natural images is to achieve the best performance possible. As reported in the literature, the combination of classifiers performs better than simple ones. The problem is how they can be combined. We propose a relaxation approach, which combines two base classifiers, namely: the probabilistic Bayesian and the fuzzy clustering. The first establishes an initial classification, where the probability values are reinforced or punished by relaxation based on the support provided by the second. A comparative analysis is carried out against classical classifiers, verifying its performance.

Keywords: Relaxation, Bayesian, fuzzy clustering, classifier, image texture classification.

1 Introduction

Nowadays the increasing technology of aerial images is demanding solutions for different image-based applications. The natural texture classification is one of such applications due to the high spatial resolutions achieved in the images. The areas where textures are suitable include agricultural crop ordination, forest or urban identifications and damages evaluation in catastrophes or dynamic path planning during rescue missions or intervention services also in catastrophes (fires, floods).

Different classical techniques have been studied for image texture classification, namely: Bayesian, K-Nearest, Neural Networks, Vector Quantization [1], [2], [3], [4], [5], among others.

An important issue reported in the literature is that the combination of classifiers performs better than simple ones [6], [7], [8], [9]. The studies carried out in [10] and [11] report the advantages of using combined classifiers against simple ones. Nevertheless, the main problem is: which is the best strategy for combining simple classifiers? This is an issue still open. Indeed in [8] and [9] it is stated that there is not best combination method, where a revision of different approaches is reported including the way in which the classifiers are combined. In this work we propose the combination of two base classifiers namely: the probability Bayesian (BP) [12] and the Fuzzy

Clustering (FC) [13], [14]. They are combined under a relaxation scheme for classifying textures in natural images. BP establishes an initial classification based on probability values and FC reinforces or punishes these values by applying a consistency criterion. This makes the main contribution of the work.

There are pixel-based and region-based approaches. A pixel-based approach tries to classify each pixel as belonging to one of the classes. The region-based identifies patterns of textures within the image and describes each pattern by applying filtering (laws masks, Gabor filters, Wavelets, etc.), it is assumed that each texture displays different levels of energy allowing its identification at different scales [15], [16], [17]. This is out of the scope of this paper.

The aerial images used in our experiments do not display texture patterns. This implies that textured regions cannot be identified by applying region-based. In this work we use a pixel-based approach under RGB color representation because it performs favorably against other color mappings, as reported in [16]. Hence, the three RGB spectral values are the features used in our method. The same texture could display different RGB levels.

The paper is organized as follows. Section 2 describes the relaxation hybrid approach, where the BP and FC base classifiers are briefly introduced. Section 3 shows experimental and comparative results. Finally, in the section 4 some concluding remarks are presented.

2 Combining Classifiers under Relaxation

Our system works in two phases: training and classification. We have available a set of training samples clustered in c classes w_j , where $j = 1, 2, \dots, c$. Each cluster consists of n_i samples where $\mathbf{x}_i^j \in w_j$ and $i = 1, 2, \dots, n_i$.

2.1 Probability Bayesian Estimator (BP)

Given a sample \mathbf{x}_i the problem is to classify it as belonging to a cluster. This is carried out by the Bayesian framework. The Bayes rule computes the *posterior* probability,

$$P(w_j | \mathbf{x}_i) = p(\mathbf{x}_i | w_j)P(w_j) / p(\mathbf{x}_i), \quad j = 1, 2, \dots, c \quad (1)$$

$P(w_j)$ is the *a priori* probability that the sample belongs to the class w_j ; $p(\mathbf{x}_i)$ is the c -fold mixture density distribution. So, the decision is made through the equation (2),

$$\mathbf{x}_i \in w_j \text{ if } p(\mathbf{x}_i | w_j)P(w_j) > p(\mathbf{x}_i | w_k)P(w_k), \quad \forall w_k \neq w_j \quad (2)$$

The main problems to be solved are the estimations of the class-conditional probability density functions $p(\mathbf{x}_i | w_j)$ and the *a priori* probabilities, $P(w_j)$. The first ones are obtained via the well-known parametric Bayesian estimator assuming a Gaussian distribution [17]. The second ones requires some previous knowledge, but if this is not possible the *a priori* probabilities are all set to the same fixed value, this is

our case; $p(x_i)$ in (1) has not discriminatory properties. Hence, the decision through (2) is made based only on the probability provided by $p(x_i | w_j)$.

2.2 Fuzzy Clustering (FC)

Given the number of clusters c and following [13], [14] the FC algorithm is based on the minimization of the objective function J ,

$$J(U; \mathbf{v}) = \sum_{i=1}^n \sum_{j=1}^c (\mu_i^j)^m d_{ij}^2. \tag{3}$$

subject to
$$\mu_i^j \in [0,1]; \sum_{j=1}^c \mu_i^j = 1; \sum_{i=1}^{n_i} \mu_i^j < n_i; 1 \leq j \leq c, 1 \leq i \leq n_i. \tag{4}$$

where $\mathbf{v} = \{v_1, v_2, \dots, v_c\}$. These cluster centers are to be updated. The $n \times c$ matrix $U = [\mu_i^j]$ contains the membership grade of pattern i with cluster j ; $d_{ij}^2 = d^2(x_i, v_j)$ is the squared Euclidean distance. The number m is called the exponent weight [13]. In order to minimize the objective function (2), the cluster centers and membership grades are chosen so that high memberships occur for patterns close to the corresponding cluster center. The higher the value of m , the less those patterns whose memberships are low contribute to the objective function. Such patterns tend to be ignored in determining the cluster centers and membership grades [14]. The parameter m is set to 4 in this paper by cross-validation [12].

The original FC computes, for each x_i at the iteration t , its membership grade and updates the cluster centers according to equation (5),

$$\mu_i^j(t) = \left(\sum_{r=1}^c (d_{ij}(t)/d_{ir}(t))^{2/(m-1)} \right)^{-1}; v_j(t+1) = \sum_{i=1}^n (\mu_i^j(t))^m x_i / \sum_{i=1}^n (\mu_i^j(t))^m \tag{5}$$

The stopping criterion of the iteration process is achieved when $\|\mu_i^j(t+1) - \mu_i^j(t)\| < \epsilon \forall ij$ or a number N_{max} of iterations is reached.

2.3 Classifier Combination by Relaxation

Given any image, the problem is to classify each pixel i located at (x,y) as belonging to a cluster w_j . The probability that the pixel i with features x_i belongs to the cluster w_j is given by the probability density function $p(x_i | w_j)$. For simplicity this probability is renamed as $p(x_i | w_j) \equiv p_i^j$. Classical approaches in relaxation labelling can be found in [18], [19]. The problem can be formulated as follows. For each cluster we build a network of nodes net_j , where each node represents a pixel location in the image $i \equiv (x, y)$. The node i in this network is initialized with p_i^j (state value) but mapped linearly to the range $[-1,+1]$ instead of $[0,+1]$. The proposed relaxation approach updates through the iteration t the initial states by considering the support provided by the FC classifier taking into account a neighbourhood of nodes

$k \in N_i^m$ around i . This is the combination scheme. The relaxation process obtains a new state at the iteration $t + 1$ by adding the fraction $f(\cdot)$ to the previous one and then averaging these values,

$$p_i^j(t+1) = \frac{1}{2} [f(l_i^j(t)) + p_i^j(t)] = \frac{1}{2} [\tanh(l_i^j(t)) + p_i^j(t)] \tag{6}$$

where $l_i^j(t) = \sum_k s_{ik}^j(t) p_k^j(t)$ is the total force exerted on node i by the other nodes $k \in N_i^m$ at the iteration t ; the term s_{ik}^j is a regularization coefficient representing the mutual influence exerted by the k neighbors over i for each net_j . The neighborhood is defined as the m -connected spatial region, N_i^m , where m is set to 8 in this work. We define a compatibility coefficient at the iteration t as follows,

$$r_{ik}^j(t) = \begin{cases} 1 - |p_i^j(t) - \mu_k^j| & k \in N_i^m, i \neq k \\ 0 & k \notin N_i^m, i = k \end{cases} \tag{7}$$

where μ_k^j is supplied by *FC*, it is the membership degree that a node (pixel) k with attributes \mathbf{x}_k belongs to the cluster w_j , computed through the equation (5). These values are mapped linearly to range in $[-1, +1]$ instead of $[0, +1]$ for compatibility computation. From (7) we can see that $r_{ik}^j(t)$ ranges in $[-1, +1]$ where the lower/higher limit means minimum/maximum influence respectively. Finally, the regularization coefficient is defined as,

$$s_{ik}^j = [\text{sgn}(r_{ik}^j)]^{\nu+1} r_{ik}^j; \quad \text{sgn}(r_{ik}^j) = \begin{cases} -1 & r_{ik}^j \leq 0 \\ +1 & r_{ik}^j > 0 \end{cases} \tag{8}$$

sgn is the *signum function* and ν is the number of negative values in the set $C \equiv \{r_{ik}^j(t), \mu_i^j(t), p_k^j\}$, i.e. given $S \equiv \{q \in C / q < 0\} \subseteq C$, $\nu = \text{card}(S)$. The process is as follows:

Initialize: $t = 0$; load each node with $p_i^j(t = 0)$; set $\mathcal{E} = 0.05$ (constant to accelerate the convergence); $t_{max} = 100$. Define *nc* as the number of nodes that change their state values at each iteration.

while $t < t_{max}$ *or* $nc \neq 0$ *set* $t = t + 1$; $nc = 0$; *for* each node i *update* $p_i^j(t + 1)$ according to the equation (6); *if* $|p_i^j(t + 1) - p_i^j(t)| > \mathcal{E}$ *then* $nc = nc + 1$.

Outputs: the states $p_i^j(t)$ for all nodes updated.

The final decision is made as described in the section 2.1 with the output states. The compatibility coefficient is the kernel of the proposed hybrid approach. Indeed, it combines both, the probabilities and membership degrees provided by BP and FC respectively. Then, through the relaxation process the initial probabilities are updated based on the own influence and the influences exerted by the neighborhoods through their corresponding membership degrees, Eq. (6).

3 Comparative Analysis and Performance Evaluation

We have used a set of 26 digital aerial images acquired during May in 2006 from the Abadin region located at Lugo (Spain). They are multispectral images with 512x512 pixels in size. The images are taken at different days from an area with several natural textures. The initial training patterns are extracted from 10 images of the full set. The remainder 16 images are used for testing and four sets, S0, S1 S2 and S3 of 4 images each one, are processed during the test according to the strategy described below. The images assigned to each set are randomly selected from the 26 images available.

3.1 Design of a Test Strategy

In order to assess the validity and performance of the proposed approach we have designed a test strategy with two goals: 1) to verify the performance of our hybrid approach as compared against some existing strategies; 2) to study the behaviour of the method as the training (i.e. the learning) increases.

Our combined *relaxation* (RS) method is compared against the base classifiers used for the combination. RS is compared against the hybrid classifier, based on the combination of BP and FC (BF) [20]. The combination is carried out through the equation (1) where the a priori probabilities are considered as supports provided by FC. RS is also compared against the following classical hybrid strategies [9]: Mean (ME), Max (MA) and Min (MI). Yager [20] proposed a multicriteria decision making approach based on fuzzy sets aggregation. So, RS is finally compared against the fuzzy aggregation (FA) defined by the equation (9),

$$\gamma_i^j = 1 - \min \left\{ 1, \left((1 - p_i^j)^a + (1 - \mu_i^j)^a \right)^{1/a} \right\} \quad a \geq 1 \quad (9)$$

This represents the join support provided by BP and FC; the decision is made based on the maximum support for all clusters. The parameter a has been fixed to 4 after trial and error.

In order to achieve the above two goals, each pixel in the images included in S0, S1, S2 and S3 is classified as belonging to a cluster. This is carried out in three STEPs. The percentage of error is computed based on the ground truth established by each cluster for each image under the supervision of the expert human criterion. In STEP 1 we classify S0 and S1. The classified samples coming from S1 are added to the previous training patterns and a new training process is carried out for BP and FC where their parameters are updated with all available samples at this STEP. In STEP 2 we classify S0 and S2 and the classified samples coming from S2 are added as new training samples for a new training process where the parameters are again updated. Finally in STEP 3 S0 and S3 are classified.

Through the three STEPs we verify the performance of the different methods as the learning increases. At the STEPs 1 and 2 we add 4x512x512 new training samples. The set S0 is a pattern set which is classified at each STEP but without intervention in the training processes carried out in STPEs 1 and 2. This allows us to verify specifically the relevance of the learning process as it increases.

Based on the assumption that the automatic training process determines four classes, we classify each image pixel with the simple classifiers obtaining a labeled image with

four expected clusters. For each class we build a binary image, which is manually touched up until a satisfactory classification is obtained under the human supervision.

Figure 1 (a) displays an original image belonging to the set S0; (b) displays the correspondence between the classes and the color assigned to the corresponding cluster center according to a color map previously defined; (c) labeled image for the four clusters obtained by our proposed RS approach; (d) ground truth for the cluster number two. The labels are artificial colors identifying each cluster. The correspondence between labels and the different textures is: 1.-yellow with forest vegetation; 2.- blue with bare soil; 3.- green with agricultural crop vegetation; 4.- red with buildings and man made structures.

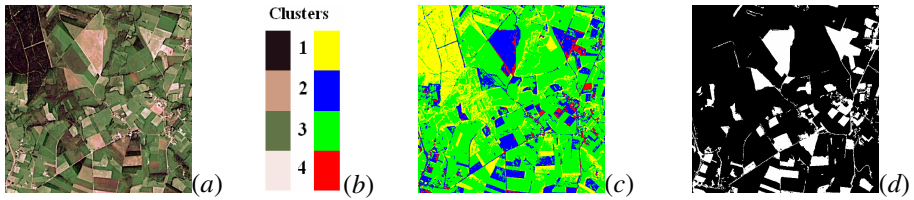


Fig. 1. (a) Original image; (b) colors-labels; (c) labeled textures (b) ground truth for cluster #2

3.2 Analysis of Results

Table 2 shows the percentage of successes in terms of correct classifications obtained for the different classifiers. For each STEP s we show the results obtained for both sets of tested images S0 and either S1 or S2 or S3. These percentages are computed taking into account the correct classifications for the four clusters according to the corresponding ground truth. The numbers in square brackets in the row RS indicates the rounded and averaged number of iterations required by each set at each STEP. From results in table 2 one can see that the best performance is achieved by the proposed RS strategy. The best performance for the simple hybrid methods is achieved by ME and for the simple methods is BP. The results show that the hybrid approaches perform favourably for the data sets used. The MA and ME fusion methods provide best results than the

Table 1. Percentage of successes obtained for the methods analysed at the three STEPs

	% Error	STEP 1		STEP 2		STEP 3	
		S0	S1	S0	S2	S0	S3
Relaxation	[iterations]	[12]	[15]	[11]	[9]	[9]	[11]
	RS	22.6	22.7	20.6	17.3	18.9	19.6
Fuzzy aggregation	FA	25.1	26.2	23.8	24.1	20.9	20.1
Hybrid Bayesian-Fuzzy	BF	23.8	24.5	22.2	20.1	19.9	20.0
Simple hybrid methods	MA	30.4	29.6	27.8	26.9	26.6	26.0
	MI	36.4	36.1	31.4	34.3	29.9	28.3
	ME	28.6	28.3	24.8	26.1	24.3	24.2
Simple methods	FC	32.1	30.2	27.1	27.4	26.0	25.9
	BP	30.2	29.1	26.1	26.4	25.2	24.7

individual ones. This means that fusion strategies are suitable for classification tasks. Moreover, as the learning increases through STEPs 1 to 3 the performance improves and the number of iterations for S_0 decreases. This means that the learning phase is important and that the number of samples affects the performance.

The best performance is achieved by the proposed method. This is based on the following two reasons: 1) the influences exerted by the neighbours nodes try to modify the probability of the node which is analysed based on their criteria; 2) the relaxation process also modifies that probability progressively and considers the own criterion of the node which is being updated.

The main drawback is its execution time, which is greater than for the remaining methods and directly proportional to the number of iterations. The implementation has been carried out in Matlab and executed on a Pentium M, 1.86 GHz with 1 GB RAM. On average the execution per iteration is 11.2 seconds.

4 Conclusions

We have proposed the combination of classifier through a relaxation strategy that performs favourably as compared with other existing strategies. This method combines supports of two simple classifiers. The BP classifier is influenced by the FC, more classifiers could be added for influencing. Also, in the future, a mutual influence could be considered. The method has been tested for images where the spatial neighbourhood is an important issue. The method can be applied to problems where a neighbourhood can be established (image change detection), but also in problems where some nodes are affected by others (social or political decisions).

Acknowledgments. Thanks to SITGA (Servicio Territorial de Galicia) and Dimap company (<http://www.dimap.es/>) for the aerial images supplied and used in this work.

References

1. del Frate, F., Pacifici, F., Schiavon, G., Solimini, C.: Use of Neural Networks for Automatic Classification from High-Resolution Images. *IEEE Trans. Geoscience and Remote Sensing* 45(4), 800–809 (2007)
2. Giacinto, G., Roli, F., Bruzzone, L.: Combination of Neural and Statistical Algorithms for Supervised Classification of Remote-Sensing. *Pattern Recognition Letters* 21(5), 385–397 (2000)
3. Chan, J.W.C., Laporte, N., Defries, R.S.: Texture Classification of Logged Forest in Tropical Africa Using Machine-learning Algorithms. *Journal Remote Sensing* 24(6), 1401–1407 (2003)
4. Hanmandlu, M., Madasu, V.K., Vasikarla, S.: A Fuzzy Approach to Texture Segmentation. In: *Proc. IEEE Int. Conf. Information Technology: Coding and Computing (ITCC 2004)*, The Orleans, Las Vegas, Nevada, pp. 636–642 (2004)
5. Lam, E.P.: Wavelet-based Texture Image Classification Using Vector Quantization. In: Astola, J.T., Egiiazarian, K.O., Dougherty, E.R. (eds.) *Proc. SPIE, Image Processing: Algorithms and Systems V*, vol. 6497, pp. 687–696 (2007)

6. Kong, Z., Cai, Z.: Advances of Research in Fuzzy Integral for Classifier's fusion. In: Proc. 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Qingdao, China, vol. 2, pp. 809–814 (2007)
7. Valdovinos, R.M., Sánchez, J.S., Barandela, R.: Dynamic and Static weighting in classifier fusion. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) *IbPRIA 2005*. LNCS, vol. 3523, pp. 59–66. Springer, Heidelberg (2005)
8. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, Chichester (2004)
9. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20(3), 226–239 (1998)
10. Partridge, D., Griffith, N.: Multiple Classifier Systems: Software Engineered, Automatically Modular Leading to a Taxonomic Overview. *Pattern Analysis and Applications* 5, 180–188 (2002)
11. Deng, D., Zhang, J.: Combining Multiple Precision-Boosted Classifiers for Indoor-Outdoor Scene Classification. *Information Technology and Applications* 1(4-7), 720–725 (2005)
12. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. John Wiley and Sons, New York (2001)
13. Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer/Plenum Press, New York (1981)
14. Zimmermann, H.J.: *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers, Norwell (1991)
15. Puig, D., García, M.A.: Automatic Texture Feature Selection for Image Pixel Classification. *Pattern Recognition* 39(11), 1996–2009 (2006)
16. Maillard, P.: Comparing Texture Methods through Classification. *Photogrammetric Engineering and Remote Sensing* 69(4), 357–367 (2003)
17. Drimbarean, P.F., Whelan, P.F.: Experiments in Colour Texture Analysis. *Pattern Recognition Letters* 22, 1161–1167 (2001)
18. Rosenfeld, A., Hummel, R., Zucker, S.: Scene labeling by relaxation operation. *IEEE Trans. Systems Man Cybernetics* 6, 420–453 (1976)
19. Hummel, R., Zucker, S.: On the foundations of relaxation labeling process. *IEEE Trans. Pattern Anal. Mach. Intell.* 5, 267–287 (1983)
20. Guijarro, M., Pajares, G., Abreu, R., Garmendia, L., Santos, M.: Design of a hybrid classifier for natural textures in images from the Bayesian and Fuzzy paradigms. In: *WISP 2007: Proceedings of the IEEE Int. Symposium on Intelligent signal Processing*, Alcalá de Henares, Spain, pp. 431–436 (2007)
21. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. System Man and Cybernetics* 18(1), 183–190 (1988)

An Ensemble Approach for the Diagnosis of Cognitive Decline with Missing Data

Patricio García Báez¹, Carlos Fernández Viadero², José Regidor García³,
and Carmen Paz Suárez Araujo³

¹ Department of Statistics, Operating Research and Computation, University of La Laguna,
38271 La Laguna, Canary Islands, Spain

pgarcia@ull.es

² Unidad de Atención a la Dependencia de Santander, Gobierno de Cantabria, 39012 Santander,
Cantabria, Spain

³ Institute of Cybernetic Sciences and Technology, University of Las Palmas de Gran Canaria,
35017 Las Palmas de Gran Canaria, Canary Islands, Spain

cpsuarez@dis.ulpgc.es

Abstract. This work applies new techniques of automatic learning to diagnose neuro decline processes usually related to aging. Early detection of cognitive decline (CD) is an advisable practice under multiple perspectives. A study of neuropsychological tests from 267 consultations on 30 patients by the Alzheimer's Patient Association of Gran Canaria is carried out. We designed neural computational CD diagnosis systems, using a multi-net and ensemble structure that is applied to the treatment of missing data present in consultations. The results show significant improvements over simple classifiers. These systems would allow applying policies of early detection of dementias in primary care centers where specialized professionals are not present.

Keywords: Cognitive Decline; Dementia; Diagnosis; Alzheimer's Disease; Cognitive Tests; Missing Data; k -fold Cross-validation; Artificial Neural Network; Modular Neural Network; Counterpropagation; Multi-net; Ensemble.

1 Introduction

The progressive aging of the world population has brought with it an increase in the incidence of diseases that are associated with this advanced age. Neurodegenerative processes, especially those including dementias, are two areas that are noteworthy. One outstanding feature is its high prevalence, 8-10% in individuals older than 65, and more than 25% of the population who are very advanced in years [1]. This prevalence has some very important repercussions in the patient's life and in other spheres. Its impact on the quality of life and individual autonomy is apparent, but we must not ignore the changes seen in the familiar, social and healthcare settings as well [2][3].

These consequences in these settings are a clear signal for research and applications to be carried out in possible precocious diagnosis. Of course the diagnosis has to be as accurate as possible while trying to resist the devastating effects of this pathology through the use of a therapeutic plan and suitable approach [4]. This is why the

early detection of CD is an advisable practice at every level of attention. However, the high uncertainty diagnosis [5], and the degree of the underdiagnosis which is so crucial and can reach 95% of the cases in some settings [6] highlights the need to develop detection programs and new instruments of diagnosis.

Three types of CD evaluation are possible. The first uses an open form, another is semi structured, while the third evaluation is carried out using the application of a series of cognitive evaluation scales. It must be stressed, however, that a test or a cognitive-based scale does not necessarily confirm deterioration or a normality diagnosis, but only indicates the possible existence of a cognitive deficit. Several neuropsychological tests [7] have been developed and used to evaluate different cognitive sections of a patient throughout the years. However, it is not a simple task to define a clear relationship between test results and specific symptoms or different levels of CD. Wide validation and testing are, of course, needed to determine levels of sensitivity, specificity and predictive value. 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 [8][9].

Automated decision making applications using consultation with *several experts* are now being rediscovered by the computational intelligence community. This application has emerged as a popular and heavily researched area, known as ensemble systems, and has produced favorable results when compared to results from single expert systems for a broad range of applications and under a variety of scenarios [10].

We present a proposal for CD diagnosis based on modular neural computation systems by means of a multi-net and an ensemble system that is directed towards the treatment of frequently missing data in physician-clinical environment.

2 Data Environment

The dataset includes results from 267 clinical consultations on 30 patients during 2005 at the Alzheimer's Patient Association of Gran Canaria [11]. The data structure includes a patient identifier, results from 5 neuropsychological tests, and a diagnosis of CD as well as differential dementia. An important advantage in this data set is in 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.

A collection of 5 different data tests are used [11]: 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).

The missing data from the MMSE test were introduced using interpolation from both annual tests. Almost all of the patients were subject to this test, and clinical experts agreed with this approximation. Still other missing values had to be accounted for, in this case 71 from a total of 1335. See Table 1 for a distribution by test and number of patients.

Table 1. Statistics of missing data in data set

<i>Test type</i>	<i>Number of missing data</i>	<i>Number of patients</i>
<i>MMSE</i>	36 (13.5%)	4 (13.3%)
<i>FAST</i>	0 (0%)	0 (0%)
<i>Katz</i>	33 (12.4%)	12 (40%)
<i>Barthel</i>	1 (0.3%)	1 (3.3%)
<i>Lawton-Brody</i>	1 (0.3%)	1 (3.3%)
<i>Total</i>	71 (5.3%)	16 (53.3%)

Different fields that make up successfully obtained information were preprocessed. This step was carried out to facilitate posterior convergence (in the use of the data set). The neuropsychological tests were standardized based on the minimum and maximum values that could be attained in these tests. Fields that were not completed were labeled as having missing values and were submitted to special treatment later.

Diagnostic values are used at a control stage. The diagnosis of the CD can result in one of four possible values: Without CD, Slight CD, Moderate CD and Severe CD. An analysis of the complete set of consultations reveals that 15% were diagnosed with Slight CD, 34% with Moderate CD and 51% with Severe CD. Recall that the data source does not include patients without CD. Regarding the diagnosis of dementias, 73% of the consultations correspond to Alzheimer-type dementias, while the remaining 27% were included in other types of dementias.

Data set size is one of the added difficulties which must be addressed when designing an artificial intelligence system which will aid in the diagnosis of DC. The data set limits how suitable the training of the used neural models will be, in addition to accurate generalization error. A method must be found that allows an efficient approach to this aspect of the problem. A cross-validation method is used and it improves training and error considerations [12]. These methods are based on making several partitions on the total data employed, or *resampling*. Diverse training is carried out with these data. A first part is used as a training set while the second one functions as a validation set. The generalization average error of the different estimations carried out on the different validation sets will provide a trustworthy measurement of the evaluated model error. The cross-validation variant was the denominated *k*-fold, and consisted in partitioning the data set into *k* subgroups, performing *k* training exercises, and leaving a validation data subgroup in each one while using the remaining (*k* - 1) as training data. The conducted partition on our queries was based on the identification of the patient involved in the consultation, that is, the consultations performed on the same patients were grouped in the same subgroup. Consequently we created 30 different subgroups, or a 30-fold consultation. This approach allows more objective results to be obtained because a model that has not been trained with other source consultations from the same patient will be used when the error validation of the consultation is carried out. In other cases the consultation using to train would have a much greater correlation with the evaluation consultation.

3 Diagnosis Systems

Two modular neural computing systems employing a multi-net and an ensemble approach [10] were used to approach the problem of CD diagnosis. The systems are based on modules that implement Counterpropagation neural networks (CPNs) [13]. The first system assembles these modules in a competitive way using a Decision Tree based Switching Ensemble (DTBSE) [14]. The desired goal is that the system chooses the module that, a priori, offers a more effective diagnosis in that type of consultation, based on available tests to diagnose a patient in a determined consultation. The second system uses a Simple Majority Voting Ensemble (SMVE) combining method to assemble the modules [10]. This system counts the outputs from the best modules as votes, and then selects as system output the class with the maximum number.

3.1 Counterpropagation Modules

Counterpropagation architecture [13] is a modular neural network of two independent-learning cascaded layers. One layer is based on supporting the classification under self-organizing learning. It includes one first stage consisting of a quantifier layer with competitive learning, in our case a Kohonen self-organizing map (SOM) [15]. The second stage implements delta rule learning, since it receives a single input value of 1 from the previous stage. This second stage is equivalent to outstar learning [16]. 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 the second stage to use a simple classifier that produces proven convergence in linear problems, better generalization skills and a reduction in the consumption of computing resources.

The problem of missing input data in the first stage is avoided by using a variant of the SOM architecture that is able to process missing data [17]. This variant prevents missing values from contributing when coming out or modifying the weights. Even so, this way of approaching missing values is insufficient by itself, the proportion of missing values is excessive for this network because of its low tolerance.

3.2 Decision Tree Based Switching Multi-net System

Our multi-net system is an alternative to the missing data processing of CPN. It treats the missing data training in all the CPN modules with different input subgroups that can be given as a variant of the 5 dimensions corresponding to each of the tests used during consultations when diagnosing a patient. The design of a DTBSE [14], adapted to the obtained results, causes the CPN module to be chosen because, a priori it more effectively diagnoses the input consultation according to the corresponding test.

The design of the DTBSE was based on the accomplishment of training and evaluation of the generalization error of the different CPN modules with every possible input combination configuration. The dominations between classifiers were obtained from the analysis of the obtained results. In other words, we considered that a classifier dominates another one if using an input subgroup used by the second one

produces a validation error equal to or less than the second one. Some modules were quickly discarded according to the obtained classifier that dominates and the suitable rules were designed to select at every moment the module with smaller generalization error according to the presence of tests in the input consultation.

3.3 Simple Majority Voting Ensemble System

Finally our DTBSE is not using the ability of CPN to process missing data, because the system only sends data with all available inputs to each module. The domination exclusion method also eliminates some of the better modules. The ability of our CPN modules to process inputs with missing data is incorporated into the design of a neural ensemble, SMVE. The neural ensemble uses the modules with the lowest average of validation errors and creates a simple majority voting process based on their outputs. The class with the maximum number of votes is selected as the system output. 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 [10].

4 Results and Discussion

Fig. 1 summarizes average error for the calculated training and validation sets by means of the cross-validation method for all of the CPN modules with different input subgroups. The validation error for each module was calculated using only the data without missing values according to the inputs of the module. It can be deduced that MMSE, FAST and Katz, in this order, are the most important tests when carrying out the diagnosis. The best module using all validation data is the CPN with the MMSE and FAST tests inputs (CPN MMSE+FAST). Average error reached in the validation sets are 8.99% and 7.44% in the training ones. Table 2 includes all the statistics for the validation sets.

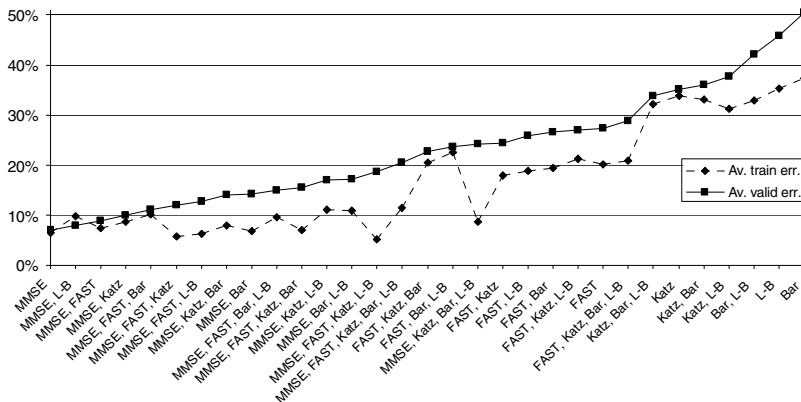


Fig. 1. Average training and validation errors for all the CPN modules with different inputs

The obtained results from Fig. 1 led to a decision to only use 12 of the most dominating modules. The DTBSE scheme used is described in Fig. 2. In it, depending on the not missing values present in the input consultation, the module with the best expected outcomes is selected.

The DTBSE obtained system presents an average error in the validation sets of 7.49%, Table 2 summarizes all the statistics for the validation sets. These results improve successfully, in a 1.5% the average validation error, to the best modules (CPN MMSE+FAST) applied on the total of the consultations, which reaffirms the utility of using a multi-net system such as the one described.

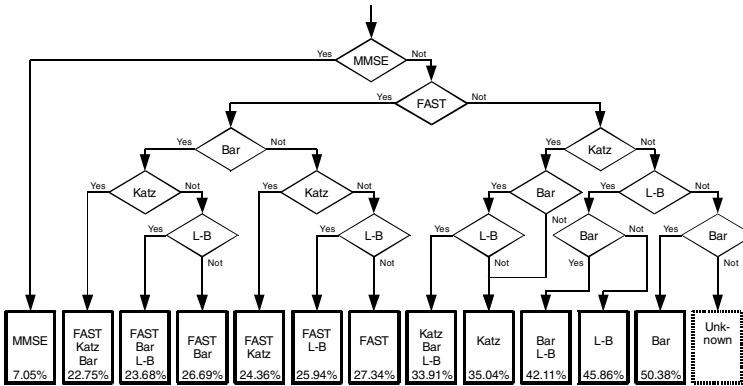


Fig. 2. DTBSE scheme and the modules used in it. Each module number is the validation error calculated using only the data without missing values according to the inputs of the module.

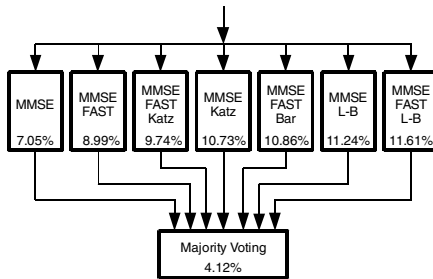


Fig. 3. SMVE scheme and the modules used in it. The number of each module is the validation error calculated using only the data with some available value to the inputs of the module.

The 7 best modules were used in the SMVE. The SMVE scheme used is described in Fig. 3. The class with the maximum number of votes is selected.

The SMVE obtained system with this approach presents an average error in the validation sets of 4.12% (see Table 2). These results successfully improve, by 4.87% the average validation error when compared to the best one from the modules (CPN MMSE+FAST) applied on the total of all consultations, and by 3.37% to the DTBSE. The drawback of this system is its computational requirements. Output computation of more than one module is required, unlike DTBSE, and combines the exit of theirs.

Table 2. Comparison between results for the validation sets of the three approaches

		<i>CPN MMSE+FAST</i>	<i>DTBSE</i>	<i>SMVE</i>
<i>Sensitivity</i>	<i>Slight CD</i>	59%	71%	76%
	<i>Moderate CD</i>	97%	93%	99%
	<i>Severe CD</i>	97%	99%	100%
<i>Specificity</i>	<i>Slight CD</i>	96%	85%	97%
	<i>Moderate CD</i>	81%	86%	90%
	<i>Severe CD</i>	100%	89%	100%
<i>Average error</i>		8.99%	7.49%	4.12%

5 Conclusions

The present work offers an important advance in the area of neuroinformatics and medical computer science based on a proposal to apply new techniques from automatic learning to diagnose cognitive decline processes usually related to aging.

The combination between the knowledge retrieved through cognitive tests and the ensemble techniques used to diagnose put our proposal in the field of Hybrid Intelligent Systems.

Required clinical tests may not be available at the time of a patient evaluation. This difficulty is overcome through the use of a variant of CPN with capacity for processing missing data. In addition, we also designed two multi-net systems that, according to missing input data, use the more suitable CPN modules during their treatment.

The use of this new instrument can help alleviate the degree of existing underdiagnosis, since it would allow policies of early detection of dementias to be applied in primary care centers where specialized professionals are not on staff. In addition the analysis conducted on the different modules that make up the system 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.

Acknowledgments. We would like to thank the Gobierno de Canarias for their support under Research Project PI2003/178A. Additional thanks are extended to the Alzheimer's Patient Association of Gran Canaria for their generous input of data.

References

1. López-Pousa, S.: La demencia: concepto y epidemiología. In: Enfermedad de Alzheimer y otras demencias, pp. 33–42. Médica Panamericana, Madrid (2006)
2. 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)
3. DeKosky, S.: Early intervention is key to successful management of Alzheimer disease. *Alzheimer Disease and Associated Disorders* 17, S99–S104 (2003)
4. Doody, R.S.: Refining treatment guidelines in Alzheimer's disease. *Geriatrics* 60, 14–20 (2005)

5. 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)
6. Solomon, P., Murphy, C.: Should we screen for Alzheimer's disease? *Geriatrics* 60, 26–31 (2005)
7. Nestor, P.J., Scheltens, P., Hodges, J.R.: Advances in the early detection of Alzheimer's disease. *Nature Reviews Neuroscience* 5, S34–S41 (2004)
8. 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)
9. Damian, M., Kreis, M., Krumm, B., Hentschel, F.: Optimized neuropsychological procedures at different stages of dementia diagnostics. *Journal of the Neurological Sciences* 229, 95–101 (2005)
10. Polikar, R.: Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine* 6, 21–45 (2006)
11. 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)
12. Hjorth, J.S.U.: *Computer Intensive Statistical Methods Validation, Mod. Sel., and Boot-stap*. Chapman & Hall, London (1994)
13. Hecht-Nielsen, R.: Counterpropagation networks. *Applied Optics* 26, 4979–4984 (1987)
14. Sharkey, A.J.C.: Multi-Net Systems. In: *Combining Artificial Neural Nets*, pp. 1–30. Springer, Heidelberg (1999)
15. Kohonen, T.: *Self-Organization and Associative Memory*, 3rd edn. Springer Series in Information Sciences, Berlin (1989)
16. Grossberg, S.: Embedding fields: A theory of learning with physiological implications. *Journal of Mathematical Psychology* 6, 209–239 (1969)
17. Samad, T., Harp, S.A.: Self-organization with partial data. *Network* 3, 205–212 (1992)

Fusers Based on Classifier Response and Discriminant Function – Comparative Study

Michał Wozniak and Konrad Jackowski

Chair of Systems and Computer Networks, Wrocław University of Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Michał.Wozniak@pwr.wroc.pl

Abstract. The *Multiple Classifier Systems* are nowadays one of the most promising directions in pattern recognition. There are many methods of decision making by the ensemble of classifiers. The most popular are methods that have their origin in voting method, where the decision of the common classifier is a combination of individual classifiers' decisions. This work presents methods of classifier combination, where neural networks plays a role of fuser block. Fusion on level of recognizer responses or values of their discriminant functions is applied. The qualities of proposed methods are evaluated via computer experiments on generated data and two benchmark databases.

1 Introduction

This paper presents an algorithm of training a compound classifier. Multiple classifier systems are currently the focus of intense research. The mentioned concept has been known for over 15 years [22]. Some works in this field were published as early as the '60 of the XX century [2], when it was shown that the common decision of independent classifiers is optimal, when chosen weights are inversely proportional to errors made by the classifiers. In the beginning in literature one could find only majority vote, but in later works more advanced methods of receiving common answer of the classifier group were proposed. Attempt to estimate classification quality of the classifier committee is one of fundamental importance. Known conclusions, derived from analytic way, concern the particular case of the majority vote [10] when classifier committee is formed on the basis of independent classifiers. Unfortunately this case has only theoretical characteristic and is not useful in practice. A weighted vote is also taken into consideration. In [18] it was stated that the optimal weight value should be dependent on the error of the single classifier and on the *prior* probability of the class, on which classifier points. One also has to mention many other works that describe analytical properties and experimental results, like [9, 11]. The problem of establishing weights for mentioned voting procedure is not simple. Many of authors have proposed treating the block of voting as a kind of classifier [12, 15] but the general question is “does the fuser need to be trained?” [7].

This paper presents method of a trained voting procedure obtained via neural networks learning. Its quality is compared to another voting schemes.

2 Problem Statement

This section presents two methods of classifier combination. The first one allows us to obtain classifier on the basis of the simple classifiers responses and the second one uses discriminant functions for the decision values.

2.1 Classifier Fusion Based on Classifier Response

Let us assume that we have n classifiers $\Psi^{(1)}, \Psi^{(2)}, \dots, \Psi^{(n)}$. Each of them decides if object belongs to class $i \in M = \{1, \dots, M\}$. For making common decision by the group of classifiers we use following classifier $\bar{\Psi}$:

$$\bar{\Psi} = \arg \max_{j \in M} \sum_{i=1}^k \delta(j, \Psi_i) w_i \Psi_i, \quad (1)$$

where w_i is the weight of i -th classifier and

$$\delta(i, j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}. \quad (2)$$

Let us note that w_i could play role of the quality of classifier $\Psi^{(i)}$. The accuracy of classifier $\bar{\Psi}$ is maximized by assigning weights

$$w_i \propto \frac{P_{a,i}}{1 - P_{a,i}}, \quad (3)$$

where $P_{a,i}$ denotes probability of accuracy of i -th classifier. Unfortunately it is not sufficient to guarantee the smallest classification error. The *prior* probability for each class has to be also taken into account. In real decision problems the values of the *prior* probabilities are usually unknown.

In [10] authors proposed to train the fuser. In our research we propose to use AWC algorithm (Adaptive Weights Calculation) [23]. Its idea has been derived from the well known single perceptron learning concept [19], where weight values are corrected in each iteration. The correction depends on estimator of probability errors of single classifiers and estimator of classifier committee based on weights counted in this step. AWC increases the weights of classifiers whose accuracies are higher than accuracy of common classifier. The pseudocode of procedure is shown below.

Procedure AWC

Input: $\Psi_1, \Psi_2, \dots, \Psi_k$ - set of k classifiers for the same decision problem, Learning set LS ,

T number of iterations

1.. For each classifier Ψ_i

a) compute weights of classifier $w_i(0) = 1/k$

b) estimate accuracy probability of $\Psi^{(i)}$ - $\hat{P}_{a,i}$

2. For $t:=1$ do T
 - a) estimate accuracy probability of $\bar{\Psi} - \hat{P}_a(t)$
 - b) $\text{sum_of_weights}:=0$
 - c) for each classifiers Ψ_i

$$w_i(t) = w_i(t-1) + \left(\frac{(\hat{P}_a(t) - \hat{P}_{a,i})}{(t + \gamma)} \right)$$
 - if $w_i(k) > 0$
 - then $\text{sum_of_weights} := \text{sum_of_weights} + w_i(k)$
 - else $w_i(t) := 0$;
 - d) if $\text{sum_of_weights} > 0$ then
 - i) for each classifier Ψ_i

$$w_i(t) := \frac{w_i(t)}{\text{sum_of_weights}}$$
 - else
 - i) for each classifier Ψ_i

$$w_i(t) = w_i(t-1)$$
 - ii) $t := T$

The γ is integer, constant value fixed arbitrary (usually given by expert or obtained via experiments) which is connected with the speed of the weights learning process. The weight corrections in each step is reverse proportional to the γ value.

2.2 Classifier Fusion Based on Values of Classifiers' Discrimination Function

Another proposition of combining method is formed by fusions of simple classifier discriminant functions, the main form of which are *posterior* probability estimators [5, 9], referring to the probabilistic model of a pattern recognition task [1, 3, 28]. The aggregating methods, which do not require training, perform fusion with the help of simple operators such as maximum or average. However, they can be used in clearly defined conditions, as it has been presented in a research paper by Duin [7], which limits their practical applications. Weighting methods are an alternative and the selection of weights has a similar importance as it is in case of a weighted majority voting. The advantages of this approach include an effective counteraction against the occurrence of elementary classifier overtraining.

Let's assume that we have n classifiers $\Psi^{(1)}, \Psi^{(2)}, \dots, \Psi^{(n)}$. Each of them makes decision based on the value of discriminant function. Let $F^{(l)}(i, x)$ means such a function of class i with given value of x , which is used by the l -th classifier $\Psi^{(l)}$. For making common decision by the mentioned group of classifiers let's use the following classifier $\hat{\Psi}(x)$

$$\hat{\Psi}(x) = i \quad \text{if} \quad \hat{F}(i, x) = \max_{k \in M} \hat{F}(k, x), \tag{4}$$

where

$$\hat{F}(i, x) = \sum_{l=1}^n \eta_l F^{(l)}(i, x) \quad \text{and} \quad \sum_{i=1}^n \eta_l = 1. \tag{5}$$

We could give varied interpretations of the discriminant function. It could be the *posterior* probability for the classifiers based on Bayes decision theory or outputs of neural network. In general the value of such function means support given for distinguished class.

We use a neural network as the fusion block where for the input neurons values of discriminant functions of elementary classifiers will be given, what is shown in Fig.1.

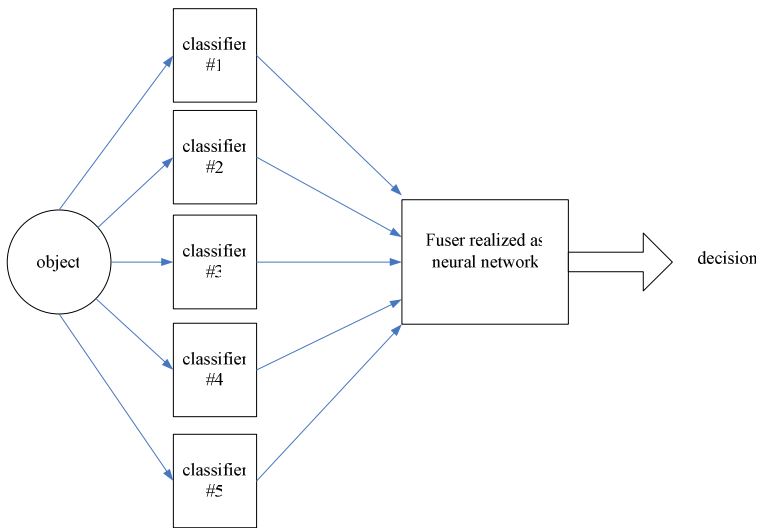


Fig. 1. Idea of classifier fusion using neural networks as fusion block. As classifier output we could use value of discriminant functions.

The details of the net are as follow:

- sigmoidal transfer function in hidden and output layers,
- back propagation learning algorithm for hidden and output layers (input layers was not trained),
- input layer of the fuser is connected to all output neurons of all elementary classifiers, therefore, its number equals number of classes of given problem multiplied by number of elementary classifiers,
- 5 neurons in hidden layers,
- as it took place in elementary classifiers, number of neurons in the last layer is equal to number of classes.

3 Experimental Investigation

The aim of the experiment is to compare the performance of combined classifiers based on simple classifier responses or values of classifier discriminant functions with the qualities of simple classifiers and with another types of classifier fusion, namely described in [13, 17].

As we mentioned we used three generated sets of data. Each of them consisted of 300 elements generated according the Higleman's [24], Banana and difficult distributions [8]. In those experiments two-class recognition task was considered with the same *prior* probability values for each class.

Additionally experiments were carried out on two benchmark databases:

- *Balance* database was originally generated to model psychological experiments reported by Siegler [25]. Database is accessible at [26]. There are 625 instances in dataset described by 4 attributes (left-weight, left-distance, right-weight, right-distance) and being classified as having the balance scale tip to the right, tip to the left, or be balanced.
- *Phoneme* database was in use in the European ESPRIT 5516 project: ROARS, the aim of which was to implement a system used for French and Spanish speech recognition. The aim of the database is to distinguish between nasal and oral vowels, which leads to dichotomy recognition problem. There are 5404 samples (vowels) coming from 1809 syllables described by five attributes (forming x vectors) that are the amplitudes of the five first harmonics.

The set of five elementary classifiers consisting of undertrained or trained neural networks (denoted as s_1 , s_2 , s_3 , s_4 , and s_5) has been used in experiments for the purpose of ensuring diversity of simple classifiers that allows their local competences to be exploited.

They were used to construct the committees proposed in previous section. The first one (based on the class numbers given by single classifiers) presented in 2.1 was denoted as AWC and FDF stood for the second one (based on the discriminant function values of single classifiers). As it was mentioned in 2.2 a neural network was used as the fuser of the discriminate functions of the elementary classifiers. The details of used neural nets are as follow:

- sigmoidal transfer function,
- back propagation learning algorithm,
- 5 neurons in hidden layer,
- 15 and 10 neurons in the input layer for Balance and Phoneme databases respectively.
- number of neurons in last layer equals number of classes of given experiment.

Proposed fusers were well trained using learning set. It has to be underlined that during fuser training process no alteration of the elementary classifiers were made as they were fully trained and ready to work.

Additionally the qualities of mentioned classifiers were compared with classifier using majority voting of all simple recognizers (denotes ad MV) and Oracle classifier. Oracle is an abstract fusion model, where if at least one of the classifiers produces the

correct class label, then the committee of classifiers produces the correct class label too. It is usually used in comparative experiments [20].

Other details of experiments were as follow:

- All experiments were carried out in Matlab environment using the PRtools toolbox [8] and own software.
- Errors of the classifiers were estimated using the ten fold cross validation method [14].

Results of experiments are presented in Fig 2.

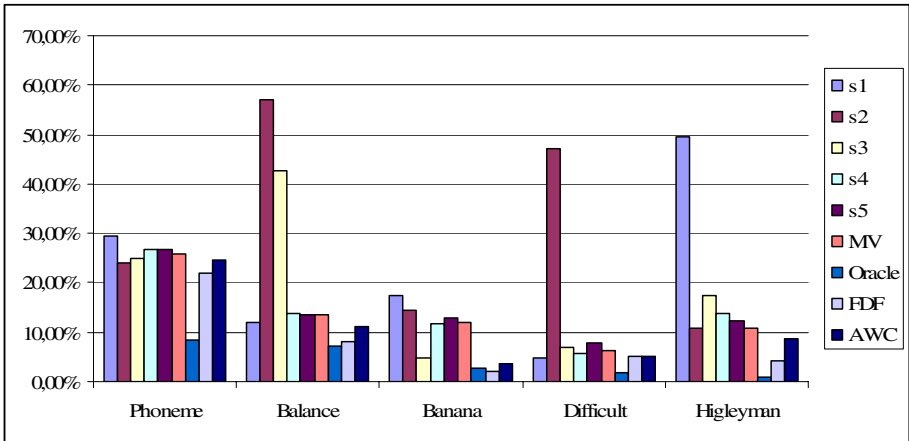


Fig. 2. Classification error for different classifiers and distributions

Firstly, one has to note that we are aware of the fact that the scope of computer experiments were limited. Therefore, making general conclusions based on them is very risky. In our opinion mentioned below statements should not be generalized at this point, but they should be confirmed by other experiments in much broader scope.

In the case of the presented experiment:

1. We could observe that the quality of committee using majority voting scheme gave worse results than the best individual or two best individual classifiers. It never achieves higher quality than the best single classifier. We have to notice that when the differences between the qualities of individual classifiers had been quite big then decision of the group classifiers should be the decision of the best classifier. It is not undesirable behavior so let's consider the following team of five independent classifiers $\Psi^{(1)}, \Psi^{(2)}, \Psi^{(3)}, \Psi^{(4)}, \Psi^{(5)}$, with accuracies $P_{a,1} = 0.55$, $P_{a,2} = 0.60$, $P_{a,3} = 0.65$, $P_{a,4} = 0.70$, $P_{a,5} = 0.80$. The probability of error of the committee of classifiers (voting according majority rule) is $P_e^{(5)} = 0.20086$. As we see it would be better if we remove the worst classifiers and reduce the ensemble to the individual and the most accurate classifier $\Psi^{(5)}$ [18].
2. Proposed committees with well trained fuser always gave better results than the individual classifiers.

3. FDF classifier always gave better results than AWC. Additionally we have to notice that FDF was definitely better than the best individual classifier, but quality of AWC is only slightly better or similar to the quality of the best individual classifier.
4. For some data sets FDF gave results slightly worse than Oracle.
5. We observed that γ factor had been playing the important role in the learning process of AWC. For presentation we chose only one value of γ with the best quality of classification. We stated that the weights values established in the first 5-10 iterations. For some experiments we observed that after about 10 iterations classification error of combined classifier was increasing. In some cases the weights of the best individual classifier exceeded 0.5 what caused that quality of decision of committee of classifiers was the same as for the best single classifier.

4 Final Remarks

The original method of learning fusion block for classifier committee was presented in this paper. The proposed methods of fusion were evaluated via computer experiments on benchmark and computer generated databases.

The obtained results justify the use of methods based on weighted combination and they are similar as published in [4, 6, 16, 21]. Unfortunately, as it was stated, it is not possible to determine weight values in the analytical way. One should although hope that in case of application of above mentioned methods for real decision tasks, we have judgment of an expert, who can formulate the heuristic function of weights selection or can be learned by methods proposed in this paper or any cited ones. The results of experiments encourage us to improve fusion method on the level of discriminant function also.

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

References

1. Biggio, B., Fumera, G., Roli, F.: Bayesian Analysis of Linear Combiners. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472, pp. 292–301. Springer, Heidelberg (2007)
2. Chow, C.K.: Statistical independence and threshold functions. *IEEE Trans. on Electronic Computers EC-16*, 66–68 (1965)
3. Devijver, P.A., Kittler, J.: *Pattern Recognition: A Statistical Approach*. Prentice Hall, London (1982)
4. Duch, W., Grudziński, K.: Ensembles of Similarity-Based Model. In: *Advances in Soft Computing*, pp. 75–85. Springer, Heidelberg (2001)
5. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. John Wiley and Sons, Chichester (2001)
6. Duin, R.P.W., Tax, D.M.J.: Experiments with Classifier Combining Rules. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 16–29. Springer, Heidelberg (2000)

7. Duin, R.P.W.: The Combining Classifier: to Train or Not to Train? In: Proc. of the ICPR 2002, Quebec City (2002)
8. Duin, R.P.W., Juszczak, P., Paclik, P., Pekalska, E., de Ridder, D., Tax, D.M.J.: PRTools4, A Matlab Toolbox for Pattern Recognition. Delft University of Technology (2004)
9. Fumera, G., Roli, F.: A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems. *IEEE Trans. on PAMI* 27(6), 942–956 (2005)
10. Hansen, L.K., Salamon, P.: Neural Networks Ensembles. *IEEE Trans. on PAMI* 12(10), 993–1001 (1990)
11. Hashem, S.: Optimal linear combinations of neural networks. *Neural Networks* 10(4), 599–614 (1997)
12. Inoue, H., Narihisa, H.: Optimizing a Multiple Classifier Systems. In: Ishizuka, M., Sattar, A. (eds.) PRICAI 2002. LNCS (LNAI), vol. 2417, pp. 285–294. Springer, Heidelberg (2002)
13. Kittler, J., Alkoot, F.M.: Sum versus Vote Fusion in Multiple Classifier Systems. *IEEE Trans. on PAMI* 25(1), 110–115 (2003)
14. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proc. of the 14th Int. Joint Conf. on Artificial Intelligence, San Mateo, pp. 1137–1143 (1995)
15. Kuncheva, L.I., Jain, L.C.: Designing classifier fusion systems by genetic algorithms. *IEEE Trans. on Evolutionary Computation* 4, 327–336 (2000)
16. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition* 34, 299–314 (2001)
17. Kuncheva, L.I., Whitaker, C.J., Shipp, C.A., Duin, R.P.W.: Limits on the Majority Vote Accuracy in Classifier Fusion. *Pattern Analysis and Applications* 6, 22–31 (2003)
18. Kuncheva, L.I.: Combining pattern classifiers: Methods and algorithms. Wiley, Chichester (2004)
19. Nilsson, N.J.: Learning machines. McGraw-Hill, New York (1965)
20. Tumer, K., Ghosh, J.: Analysis of Decision Boundaries in Linearly Combined Neural Classifiers. *Pattern Recognition* 29, 341–348 (1996)
21. Van Erp, M., Vuurpijl, L.G., Schomaker, L.R.B.: An overview and comparison of voting methods for pattern recognition. In: Proc. of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR.8), Niagara-on-the-Lake, Canada, pp. 195–200 (2002)
22. Xu, L., Krzyżak, A., Suen C.Y.: Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. *IEEE Trans. on SMC* 3, 418–435 (1992)
23. Wozniak, M.: Experiments with trained and untrained fusers. In: Corchado, E., et al. (eds.) Innovations in hybrid intelligent systems, Advances in Soft Computing. Springer, Berlin, pp. 144–150 (2007)
24. Higleyman, W.H.: The design and analysis of pattern recognition experiments. *Bell Systems Technical Journal* 41, 723–744
25. Siegler, R.S.: Three Aspects of Cognitive Development. *Cognitive Psychology* 8, 481–520 (1976)
26. Newman, D.J., Hettich, S., 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/~mllearn/MLRepository.html>

Simple Clipping Algorithms for Reduced Convex Hull SVM Training

Jorge López, Álvaro Barbero, and José R. Dorronsoro*

Dpto. de Ingeniería Informática and Instituto de Ingeniería del Conocimiento
Universidad Autónoma de Madrid, 28049 Madrid, Spain
{jorge.lopez, alvaro.barbero, jose.r.dorronsoro}@iic.uam.es

Abstract. It is well known that linear slack penalty SVM training is equivalent to solving the Nearest Point Problem (NPP) over the so-called μ -Reduced Convex Hulls, that is, convex combinations of the positive and negative samples with coefficients bounded by a $\mu < 1$ value. In this work we give a simple approach to the classical Gilbert-Schlesinger-Kozinec (GSK) and Mitchell-Demyanov-Malozemov (MDM) algorithms for NPP that naturally leads to their updating pattern selection criteria and suggests an easy way to extend them to coefficient clipping algorithms for NPP over the μ -Reduced Convex Hulls. Although the extended GSK algorithm does not perform as well as the more complex recent proposal by Mavroforakis and Theodoridis, clipping MDM coefficient updates results in a fast and efficient algorithm.

1 Introduction

Given a sample $S = \{(X_i, y_i) : i = 1, \dots, N\}$ with $y_i = \pm 1$, the standard formulation of SVM for linearly separable problems seeks [1] to maximize the margin of a separating hyperplane by solving the problem

$$\min \frac{1}{2} \|W\|^2, \text{ with } y_i(W \cdot X_i + b) \geq 1, \text{ } i = 1, \dots, N. \quad (1)$$

In practice, however, the problem actually solved is the simpler dual problem of minimizing

$$W(\alpha) = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j X_i \cdot X_j - \sum_i \alpha_i \text{ with } \alpha_i \geq 0, \sum_i \alpha_i y_i = 0 \quad (2)$$

The optimal weight W^o can be then written as $W^o = \sum \alpha_i^o y_i X_i$ and patterns for which $\alpha_i^o > 0$ are called support vectors (SV). There are quite a few proposals of algorithms to solve (2) which can be broadly divided into two categories. In the first

* All authors have been partially supported by Spain's TIN 2007-66862. The first author is supported by an IIC doctoral fellowship and the second author by the FPU-MEC grant reference AP2006-02285.

one, decomposition algorithms, Platt's SMO [2] or Joachims' SVM-Light [3] are the best-known ones. Here, however, we will consider the alternative geometrical methods, that train SVMs by solving the Nearest Point Problem (NPP; see [4]) in which we want to find the nearest points W_+^* and W_-^* of the convex hulls $C(S_{\pm})$ of the positive $S_+ = \{X_i : y_i = 1\}$ and negative $S_- = \{X_i : y_i = -1\}$ sample subsets. The maximum margin hyperplane is then $W^* = W_+^* - W_-^*$ and the optimal margin is given by $\|W^*\|/2$. If we write a $W_+ \in C(S_+)$ as $W_+ = \sum \alpha_p X_p$, with $\sum \alpha_p = 1$ and a $W_- \in C(S_-)$ as $W_- = \sum \alpha_q X_q$, with $\sum \alpha_q = 1$ we have $W = W_+ - W_- = \sum \alpha_i y_i X_i$ with $X_i \in S = S_+ \cup S_-$. We can thus state the NPP problem as follows:

$$\min \frac{1}{2} \|W\|^2, \text{ with } \alpha_i \geq 0, \sum_i \alpha_i y_i = 0, \sum_i \alpha_i = 2, \tag{3}$$

where we assume again a linearly separable training sample. In [5, 6] specific algorithms have been proposed for NPP that originate in the more classical Gilbert-Schlesinger-Kozinec (GSK; [7, 8]) and Mitchell-Demyanov-Malozemov (MDM; [9]) algorithms to find the minimum norm vector of a convex set.

Although SVM algorithms for linearly separable problems extend immediately to non separable ones if square penalties $\sum \xi_i^2$ are applied to margin slacks ξ_i [10], a different setup has to be pursued if linear penalties $\sum \xi_i$, the most widely used alternative, are considered. Geometric algorithms have then (see [4]) to solve NPP over the so-called μ -Reduced Convex Hulls (μ -RCH), where an extra restriction $\alpha_i \leq \mu$ has to be added to those in (3). Recently some proposals [11, 12] have been presented to solve NPP over RCHs which are based on the observation that the optimal W^* must be a combination of extreme points of the μ -RCH, that have the form $W^* = \mu \sum_{p=1}^K X_{i_p} + (1 - K\mu) X_{i_{K+1}}$ for an appropriate sample pattern selection X_{i_p} and where $K = \lfloor 1/\mu \rfloor$. Both algorithms adapt respectively the standard GSK and MDM algorithms to this situation.

However, a simpler approach could be to simply “clip” the GSK and MDM α coefficient update rules so that the $\alpha_i \leq \mu$ bound is kept. In this work we shall follow this approach and study such clipped versions of the GSK and MDM algorithms. In section 2 we will briefly review the standard GSK and MDM algorithms under a new, simple and clear viewpoint, that also leads naturally to clipped versions that we shall introduce in section 3, where we shall also show how clipped GSK may run into some pathological situations that may render its convergence quite slow while, on the other hand, this is much less likely for clipped MDM. We will numerically illustrate this in section 4 where we will compare our clipped GSK algorithm with the one proposed by Mavroforakis and Theodoridis in [11], showing the latter to be faster even if at each step it

must perform an ordering of the projections $W \cdot X_i$ of the sample points X_i over the current $W = W_+ - W_-$. However, we shall also see that the clipped MDM is much faster than either RCH version of GSK. A brief discussion ends the paper.

2 Standard GSK and MDM Algorithms

2.1 Standard GSK Algorithm

The GSK algorithm uses a single pattern X_l to update one of the components W_{\pm} of the current weight vector $W = W_+ - W_-$ to a new one $W'_{\pm} = (1 - \delta)W_{\pm} + \delta X_l$ where X_l is in the same class of the W_{\pm} vector being updated. The new coefficients are thus $\alpha'_i = (1 - \delta)\alpha_i + \delta \Xi_{il}$, with $\Xi_{il} = 1$ if $i = l$ and 0 otherwise. Assume first that we want to update W_+ . The new vector

$$W' = W'_+ - W_- = (1 - \delta)W_+ + \delta X_l - W_- = W + \delta(X_l - W_+) = W + \delta y_l \tilde{X}_l,$$

where we write $\tilde{X}_l = X_l - W_+$ and use the fact that now $y_l = 1$. Similarly, if we want to update W_- , we would have $W' = W'_+ - W'_- = W_+ - (1 - \delta)W_- - \delta X_l = W + \delta y_l \tilde{X}_l$, where we now use $y_l = -1$. In both cases we have

$$\|W'\|^2 = \|W + \delta y_l \tilde{X}_l\|^2 = \|W\|^2 + 2\delta y_l W \cdot \tilde{X}_l + \delta^2 \|\tilde{X}_l\|^2 = \Phi(\delta) \tag{4}$$

and, therefore, $\Phi'(\delta) = 2(y_l W \cdot \tilde{X}_l + \delta \|\tilde{X}_l\|^2)$. Thus, $\Phi'(0) = 2y_l W \cdot \tilde{X}_l$ and the \tilde{X}_l update will provide a decrease in $\|W'\|^2$ whenever $y_l W \cdot \tilde{X}_l = y_l W \cdot (X_l - W_{\pm}) < 0$, that is, $y_l W \cdot X_l = y_l W \cdot W_{\pm} < 0$. If this is the case, the optimal δ^* value is

$$\delta^* = -\frac{y_l W \cdot \tilde{X}_l}{\|\tilde{X}_l\|^2}$$

and the corresponding $\Phi(\delta^*)$ becomes

$$\Phi(\delta^*) = \|W\|^2 - 2 \frac{(y_l W \cdot \tilde{X}_l)^2}{\|\tilde{X}_l\|^2} + \frac{(y_l W \cdot \tilde{X}_l)^2}{\|\tilde{X}_l\|^2} = \|W\|^2 - \frac{(y_l W \cdot \tilde{X}_l)^2}{\|\tilde{X}_l\|^2}.$$

If we ignore the $\|\tilde{X}_l\|^2$ denominator, the maximum decrease in $\|W'\|^2$ will be obtained if we take $l = \arg \max_l \{(y_l W \cdot X_l - y_l W \cdot W_{\pm})^2\} < 0$ which, since we must have $y_l W \cdot X_l < y_l W \cdot W_{\pm}$, will be given by

$$l = \arg \min_i \{ y_i W \cdot X_i : y_i W \cdot X_i < y_i W \cdot W_{\pm} \} . \tag{5}$$

We finally observe that we must clip δ^* if necessary so that we ensure $(1 - \delta^*)\alpha_l + \delta^* \leq 1$, which means that we must force $\delta^* \leq 1$.

2.2 Standard MDM Algorithm

The MDM algorithm uses two pattern vectors X_{i_1}, X_{i_2} to update the current weight W to another one of the form $W' = W + \delta_{i_1} y_{i_1} X_{i_1} + \delta_{i_2} y_{i_2} X_{i_2}$. The advantage of this is that at each step we will have to modify only two multipliers α_{i_1} and α_{i_2} instead of all, as it was the case in GSK. The restrictions in (3) imply $2 = \sum \alpha'_i = \sum \alpha_i + \delta_{i_1} + \delta_{i_2} = 2 + \delta_{i_1} + \delta_{i_2}$ and $0 = \sum y_i \alpha'_i = \sum y_i \alpha_i + y_{i_1} \delta_{i_1} + y_{i_2} \delta_{i_2} = y_{i_1} \delta_{i_1} + y_{i_2} \delta_{i_2}$. The second one implies that $y_{i_1} \delta_{i_1} = -y_{i_2} \delta_{i_2}$ and, since the first one gives $\delta_{i_1} = -\delta_{i_2}$, we must also have $y_{i_1} = y_{i_2}$. As a consequence, $W' = W + \delta_{i_2} y_{i_2} (X_{i_2} - X_{i_1}) = W + \delta_{i_2} y_{i_2} Z_{i_2, i_1}$, where $Z_{i,j} = X_i - X_j$. Thus, $\|W'\|^2$ is a function of δ_{i_2} and we have

$$\begin{aligned} \Phi(\delta_{i_2}) &= \|W'\|^2 = \|W\|^2 + 2\delta_{i_2} y_{i_2} W \cdot Z_{i_2, i_1} + \delta_{i_2}^2 \|Z_{i_2, i_1}\|^2, \\ \Phi'(\delta_{i_2}) &= 2\{y_{i_2} W \cdot Z_{i_2, i_1} + \delta_{i_2} \|Z_{i_2, i_1}\|^2\}. \end{aligned}$$

In particular, since now $\Phi'(0) = \|W'\|^2 = 2y_{i_2} W \cdot Z_{i_2, i_1}$, $\|W'\|^2$ will decrease provided we find two vectors X_{i_2}, X_{i_1} such that $y_{i_2} \Delta = y_{i_2} W \cdot (X_{i_2} - X_{i_1}) < 0$. As done before, solving $\Phi'(\delta_{i_2}) = 0$ gives the α increments

$$\delta_{i_2} = -y_{i_2} \frac{\Delta}{\|Z_{i_2, i_1}\|^2}, \quad \delta_{i_1} = -\delta_{i_2} = y_{i_2} \frac{\Delta}{\|Z_{i_2, i_1}\|^2},$$

and it follows that

$$\|W'\|^2 = \|W\|^2 - \frac{\Delta^2}{\|Z_{i_2, i_1}\|^2} .$$

Notice that the condition $y_{i_2} \Delta < 0$ makes δ_{i_2} be positive. In other words, we have to make $\Delta < 0$ if $y_{i_2} = 1$ and $\Delta > 0$ if $y_{i_2} = -1$. If we ignore now the $\|Z_{i_2, i_1}\|^2$ denominator and write $I_{\pm} = \{i : y_i = \pm 1\}$, we simply select

$$\Delta^+ = \min_{j^+} \{W \cdot X_j\} - \max_{j^+} \{W \cdot X_j\}, \Delta^- = \max_{j^-} \{W \cdot X_j\} - \min_{j^-} \{W \cdot X_j\},$$

which we do by choosing

$$\begin{aligned} i_2^+ &= \arg \min_{j^+} \{W \cdot X_j\}, i_1^+ = \arg \max_{j^+} \{W \cdot X_j\}, \\ i_2^- &= \arg \max_{j^-} \{W \cdot X_j\}, i_1^- = \arg \min_{j^-} \{W \cdot X_j\}, \end{aligned}$$

Since this implies that $\delta_{i_2^\pm} > 0$, but $\delta_{i_1^\pm} < 0$, we must refine the i_1^\pm choices to

$$\begin{aligned} i_1^+ &= \arg \max_j \{W \cdot X_j : y_j = 1, \alpha_j > 0\}, \\ i_1^- &= \arg \min_j \{W \cdot X_j : y_j = -1, \alpha_j > 0\}. \end{aligned}$$

Finally, we select $i_{1,2} = i_{1,2}^+$ if $\Delta^+ > \Delta^-$ and $i_{1,2} = i_{1,2}^-$ otherwise.

3 Clipped GSK and MDM Algorithms for RCH SVM

The basic idea in the work by Mavroforakis et al. [11] is that it is possible to calculate the updating X_l in the RCH as the vertex of the RCH with least margin. This calculation implies a combination of the projections of a certain number of vertices of the standard CH ($\mu = 1$ case) along the direction W , so that a computationally expensive sorting of the projections is required.

On the other hand, in order to arrive to a clipped GSK algorithm, we may dispense with the sorting step simply by working with the X_l selected as in (5), that is, the vertex with least margin in the standard CH, instead of in the μ -RCH. With such a choice, the update will obviously be the same as in GSK, but this time we have to clip δ so as not to get out of the RCH. Recall that the updates in GSK are $\alpha_i' = (1 - \delta)\alpha_i + \delta \Xi_{il}$, with $0 \leq \delta \leq 1$. Thus, if $i \neq l$ and $0 \leq \alpha_i \leq \mu$, we clearly have $0 \leq \alpha_i' \leq \mu$. However, if $i = l$ and we want the update $\alpha_l' = (1 - \delta)\alpha_l + \delta$ to be $\leq \mu$, we must take $\delta \leq (\mu - \alpha_l)/(1 - \alpha_l)$ (observe that with $\mu = 1$ we recover the previous $\delta \leq 1$ bound for standard GSK). Moreover, since we have $\alpha_l' = \alpha_l + (1 - \alpha_l)\delta > \alpha_l$, it is clear that we cannot consider X_i patterns with $\alpha_i = \mu$ as updating candidates. We must thus refine the previous l choice as

$$l = \arg \min_i \left\{ y_i W \cdot X_i : \alpha_i < \mu, y_i W \cdot X_i < y_i W \cdot W_\pm \right\}. \quad (6)$$

We will call the resulting algorithm clipped GSK, or C-GSK in short. When applying the last l selection there is the obvious risk that the right hand side index set is empty. How this may happen is shown in figure 1, where a convex hull formed by

four vectors is depicted. We assume $W_- = (0,0)$, $W = W_+$ is the point in the center of the convex hull (which lies obviously inside the reduced hull), $\mu = 1/2$ and the coefficients are the ones shown. It is clear that the only possible choices for X_l would be the points with a zero coefficient, but for them $W \cdot X_i = W \cdot W_{\pm}$. No updating vector could then be found and C-GSK would be stuck at a clearly wrong W weight. Interestingly enough, this condition never happened in our experiments so the patterns provided by (6) did always improve $\|W\|^2$. Nevertheless, the situation described might be very close to actually happen, so that the norm decrease may be very small. As we shall see, this may explain why after many iterations C-GSK often gives a poor performance.

Turning our attention to the MDM algorithm, since it only increases the $\alpha_{i_2^+}$ and $\alpha_{i_2^-}$ coefficients, in a μ -RCH setting we can simply refine our previous choice of i_2^{\pm} as

$$\begin{aligned} i_2^+ &= \arg \min_j \{W \cdot X_j : y_j = 1, \alpha_j < \mu\}, \\ i_2^- &= \arg \max_j \{W \cdot X_j : y_j = -1, \alpha_j < \mu\} \end{aligned} \quad (7)$$

Moreover, after the optimal i_2, i_1 have been chosen, we must make sure that

$\alpha'_{i_2} = \alpha_{i_2} + \delta_{i_2} \leq \mu$ and $\alpha'_{i_1} = \alpha_{i_1} - \delta_{i_2} \geq 0$. This forces us to clip δ_{i_2} as

$$\delta_{i_2} = \min(\delta_{i_2}, \mu - \alpha_{i_2}, \alpha_{i_1}).$$

In what follows we shall refer to this clipped MDM procedure as C-MDM.

4 Numerical Experiments

We will first compare the performance of C-GSK and that of the RCH GSK algorithm (R-GSK) given in [11] on 10 of G. Rätsch's benchmark datasets [13]. In table 1 each algorithm's performance is measured in terms of test classification errors, final number of support vectors and number of iterations done. The stopping criterion for R-GSK will be a small reduction of the norm; more precisely, we will stop when $\|W\|^2 - \|W'\|^2 \leq \varepsilon \|W\|^2$, with a tolerance level of $\varepsilon = 10^{-6}$. As for C-GSK, we will perform three times more iterations than in R-GSK in order to achieve a similar execution time in both algorithms (in R-GSK we have used the well known QuickSort algorithm, that has an average complexity of $2N \log N$, with N being the sample size); it is clear that an iteration of R-GSK will be more expensive than one of C-GSK.

As it can be seen in table 1, the performance of C-GSK is significantly worse than that of R-GSK in most of the datasets, even if it is allowed to have three times more iterations. Besides, C-GSK shows an inability to remove support vectors: both algorithms use each hull's barycenter as the starting weight vectors and, when it ends, all

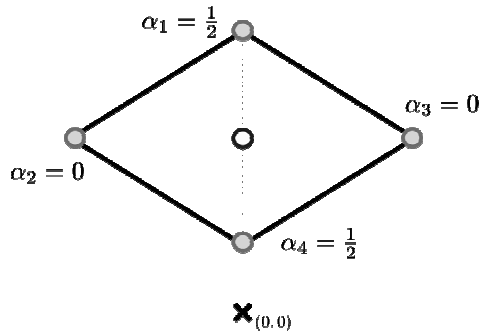


Fig. 1. Example of a situation where C-GSK gets stuck without finding the optimum. The optimum point is a bounded support vector with $\mu = 1/2$. Any movement towards the two possible choices for X_l would produce an increase of the norm.

patterns are support vectors for the C-GSK's final weight W . R-GSK is slightly better on this, but it is neither able to remove many wrong initial SV choices. This is in fact a characteristic inherited from standard GSK. In short, besides not been able to remove any support vector, C-GSK seems to progress much more slowly than R-GSK towards the optimum and, when it ends, seems still to be far from reaching it, as its test errors are clearly worse.

On the other hand, the results obtained by C-MDM are quite different and much better. They are shown in table 2 over the previous 10 datasets. The stopping criterion is the same than for R-GSK (this time we do not round the number of iterations) and the algorithm also starts at the barycenters of the reduced convex hulls. Comparing both tables, it is clear that the C-MDM error rates are smaller than those of C-GSK; they are also better than those of R-GSK except for the *image* and *splice* datasets, for which the ε value used may not be accurate enough. Besides, the number of final SVs is much smaller in C-MDM, something to be expected, as standard MDM is designed to get rid of wrong SV choices; C-MDM seems to inherit this property. The number of C-MDM iterations is also greatly reduced; moreover, it should also be noted that the cost of a C-MDM iteration is smaller than that of a C-GSK one and even more so for R-GSK.

Table 1. Average test errors, number of support vectors and number of iterations given by the R-GSK and C-GSK algorithms, with $\varepsilon = 10^{-6}$

DATASET	ERR. R	ERR. C	SVs R	SVs C	IT. R	IT. C
TITANIC	24.0±7.3	25.1±5.4	148.5±3.4	150.0±0.0	20000	60000
HEART	16.0±3.1	16.1±3.5	107.7±11.8	170.0±0.0	500	1500
DIABETES	23.8±1.8	29.1±4.8	354.3±23.3	468.0±0.0	500	1500
CANCER	29.1±4.8	31.1±4.1	199.8±2.2	200.0±0.0	2000	6000
THYROID	4.3±2.0	4.6±2.1	128.8±16.8	140.0±0.0	1500	4500
FLARE	32.9±1.6	34.9±1.7	658.7±16.8	666.0±0.0	1000	3000
SPLICE	11.1±0.9	11.1±1.0	1000.0±12.8	1000.0±0.0	15000	45000
IMAGE	3.2±0.6	4.5±1.5	1300.0±0.0	1300.0±0.0	19000	57000
GERMAN	24.6±2.3	31.8±2.4	697.7±12.1	700.0±0.0	1000	3000
BANANA	13.4±2.7	13.5±3.3	400.0±0.8	400.0±0.0	35000	105000

Table 2. Average test errors, number of support vectors and number of iterations given by the C-MDM algorithm, with $\varepsilon = 10^{-6}$

DATASET	ERR. C-MDM	SVs C-MDM	IT. C-MDM
TITANIC	23.7±6.2	114.0±9.1	125.4±21.6
HEART	16.0±3.2	83.2±6.7	208.9±21.9
DIABETES	23.8±1.8	265.4±7.6	534.7±27.4
CANCER	28.8±4.9	116.0±6.4	694.3±135.3
THYROID	4.3±2.0	25.1±5.6	225.0±35.1
FLARE	32.9±1.7	514.5±25.6	684.1±137.5
SPLICE	11.5±1.8	630.8±13.3	1081.1±348.5
IMAGE	5.1±3.0	327.2±155.8	4718.8±3488.5
GERMAN	24.0±2.1	413.3±15.2	922.2±209.5
BANANA	12.3±1.3	102.1±18.5	20890.0±42118.4

5 Discussion and Conclusions

The equivalence between SVM training for a linearly separable sample and solving the Nearest Point Problem for the convex hulls of the positive and negative subsamples is well-known. It is also known that linear slack penalty SVM training is equivalent to solving NPP over the so-called μ -Reduced Convex Hulls, where the convex combination coefficients are bounded by μ instead of 1 as it is the case for standard convex hulls. Recently some algorithms have been proposed for μ -RCH that extend the classical Gilbert-Schlesinger-Kozinec and Mitchell-Demyanov-Malozemov algorithms. On the other hand, a natural idea to solve μ -RCH is to appropriately clip the coefficient update rules of the GSK and MDM algorithms.

In this work we have given a simple approach to the linearly separable sample GSK and MDM algorithms that naturally leads to their updating pattern selection criteria and suggests an easy way to extend them to coefficient clipping algorithms for μ -RCH. Although the so extended GSK algorithm does not perform as well as the more complex proposal presented in [11], clipping results in a fast and efficient μ -RCH MDM algorithm. Future work will focus on refining the algorithms proposed here and comparing the clipped MDM algorithm with the more involved MDM extensions proposed in [12].

References

- Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization and beyond. MIT Press, Cambridge (2002)
- Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Advances in Kernel Methods – Support Vector Machines, pp. 185–208 (1999)
- Joachims, T.: Making Large-Scale Support Vector Machine Learning Practical. In: Advances in Kernel Methods – Support Vector Machines, pp. 169–184 (1999)
- Bennett, K., Bredensteiner, E.: Duality and Geometry in SVM Classifiers. In: 17th Int. Conf. on Machine Learning, pp. 57–64 (2000)

5. Keerthi, S., Shevade, S., Bhattacharyya, C., Murthy, K.: A Fast Iterative Nearest Point Algorithm for Support Vector Machine Classifier Design. *IEEE Trans. on Neural Networks* 11(1), 124–136 (2000)
6. Franc, V., Hlaváč, V.: Simple Solvers for Large Quadratic Programming Tasks. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) *DAGM 2005. LNCS*, vol. 3663, pp. 75–84. Springer, Heidelberg (2005)
7. Gilbert, E.: Minimizing the Quadratic Norm on a Convex Set. *SIAM J. Contr.* 4, 61–79 (1966)
8. Franc, V., Hlaváč, V.: An Iterative Algorithm Learning the Maximal Margin Classifier. *Pattern Recognition* 36, 1985–1996 (2003)
9. Mitchell, B., Dem'yanov, V., Malozemov, V.: Finding the Point of a Polyhedron Closest to the Origin. *SIAM J. Contr.* 12, 19–26 (1974)
10. Shawe-Taylor, J., Cristianini, N.: On the Generalisation of Soft Margin Algorithms. *IEEE Trans. on Information Theory* 48(10), 2711–2735 (2002)
11. Mavroforakis, M., Theodoridis, S.: A Geometric Approach to Support Vector Machine (SVM) Classification. *IEEE Trans. on Neural Networks* 17(3), 671–682 (2006)
12. Tao, Q., Wu, G.W., Wang, J.: A General Soft Method for Learning SVM Classifiers with L1-Norm Penalty. *Pattern Recognition* 41(3), 939–948 (2008)
13. Rätsch, G.: Benchmark Repository,
<http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm>

A WeVoS-CBR Approach to Oil Spill Problem

Emilio Corchado¹, Bruno Baruque², Aitor Mata², and Juan M. Corchado²

¹ University of Burgos, Spain

² University of Salamanca, Spain

escorchado@ubu.es, bbaruque@ubu.es, corchado@usal.es,
aitor@usal.es

Abstract. The hybrid intelligent system presented here, forecasts the presence or not of oil slicks in a certain area of the open sea after an oil spill using Case-Based Reasoning methodology. The proposed CBR includes a novel network for data classification and data retrieval. Such network works as a summarization algorithm for the results of an ensemble of Visualization Induced Self-Organizing Maps. This algorithm, called Weighted Voting Superposition (WeVoS), is mainly aimed to achieve the lowest topographic error in the map. The system uses information obtained from various satellites such as salinity, temperature, pressure, number and area of the slicks. WeVoS-CBR system has been able to accurately predict the presence of oil slicks in the north west of the Galician coast, using historical data.

Keywords: Case-Based Reasoning; Oil Spill; Topology Preserving Maps; Ensemble summarization; Self Organizing Memory; RBF.

1 Introduction

When an oil spill is produced, the natural risks are evident, and complicated decisions must be taken in order to avoid great natural disasters. Predicting if an area is going to be affected by the slicks generated after an oil spill will provide a great aid to take those decisions.

The ocean is a highly variable environment where accurate predictions are difficult to achieve. The complexity of the system is increased if external elements are introduced in the analysis. In this case, oil spills are added, generating a rough set of elements.

The solution given by the system presented in this paper is a probability of finding oil slicks in a certain area. The proposed system is a forecasting Case-Based Reasoning system [1]. A CBR system has the ability to learn from past situations, and to generate solutions to new problems based in the past solutions given to past problems.

The hybrid system combines the efficiency of the CBR systems with artificial intelligence techniques in order to improve the results and to better generalize from past data.

The developed system has been constructed using historical data and the knowledge generated after the Prestige accident, from November 2002 to April 2003. The WeVoS-ViSOM algorithm [2] is applied for the first time under the frame of a CBR to perform classification tasks, when creating the case base. The algorithm also facilitates the retrieval phase, and makes it faster by properly grouping together those cases that are actually similar.

After explaining the oil spill problem, both the CBR methodology and the WeVoS-ViSOM algorithm are explained. Then, the developed system is described, ending with the results, conclusions and future work.

2 Description of the Oil Spill Problem

Once an oil spill occurs, the evolution of the oil slicks generated must be supervised or even predicted, in order to know if an area is going to be contaminated or even better, to avoid contamination in some critical areas. To get an accurate prediction it is necessary to know how the oil slicks behave or, at least, the probability of finding oil slicks in an area.

First, position, shape and size of the oil slicks must be identified. Data related with the oil slicks, like their positions and sizes, has been obtained by treating SAR (*Synthetic Aperture Radar*) satellite images [3, 4]. The satellite images show certain areas where it seems to be nothing, like zone with no waves; that are the oil slicks. With these kind of images it is possible to distinguish between normal sea variability and slicks. It is also important to distinguish between oil slicks and look-alikes.

Once the slicks are identified, it is also essential to know the atmospheric and maritime situation that is affecting the slick in the moment that is being analysed. Information collected from satellites is used to obtain the atmospheric data needed. That is how different variables such as temperature, sea height and salinity are measured in order to obtain a global model that can explain how slicks evolve.

3 Case-Based Reasoning Systems

Case-Based Reasoning origins are in knowledge based systems. CBR systems solve new problems acquiring the needed knowledge from previous situations [5]. The principal element of a CBR system is the case base, a structure that stores the information used to generate new solutions. In the case base, data is organized into cases, where problems and its solutions are related. A case base can then be seen as a kind of database where a series of problems are stored, as long as their solutions and the relation between them.

The learning capabilities of the CBR systems are due to its own structure, composed of four main phases [6]: *retrieval*, *reuse*, *revision* and *retention*.

Applying CBR to solve a problem generally implies using other artificial intelligence techniques to solve the problems related with the different phases of the CBR cycle. In this study, a new algorithm is used to structure the case base and to easily and fast recover the most similar cases from the case base. That algorithm is the WeVoS-ViSOM algorithm, which will be explained next.

4 WeVoS-ViSOM for CBR

In this section, a novel classification algorithm for CBR is presented. It is used to sort out all the information that is stored in the case base and to retrieve the most similar cases to the one introduced in the system as a problem to solve.

4.1 Visualization Induced SOM (ViSOM)

The Self-Organizing Map (SOM) algorithm [7] and the Visualization Induced Self-Organizing Map (ViSOM) [8] are different types of Topology Preserving Mappings with a common target: to provide a low dimensional representation of multi-dimensional datasets while preserving the topological properties of the input space.

The ViSOM, aims to directly preserve the local distance information on the map, along with the topology. It constrains the lateral contraction forces between neurons and hence regularises the interneuron distances so that distances between neurons in the data space are in proportion to those in the input space [8].

Update of neighbourhood neurons in ViSOM:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t) \left([x(t) - w_v(t)] + [w_v(t) - w_k(t)] \left(\frac{d_{vk}}{\Delta_{vk}\lambda} - 1 \right) \right) \quad (1)$$

where w_v is the winning neuron, α the learning rate of the algorithm, $\eta(v,k,t)$ is the neighbourhood function where v represents the position of the winning neuron in the lattice and k the positions of the neurons in the neighbourhood of this one, x is the input to the network and λ is a “resolution” parameter, d_{vk} and Δ_{vk} are the distances between the neurons in the data space and in the map space respectively.

4.2 Weighted Voting Summarization (WeVoS)

The idea behind the novel fusion algorithm, WeVoS, is to obtain a final map keeping one of the most important features of this type of algorithms: its topological ordering. WeVoS is an improved version of an algorithm presented in several previous works: [2, 9, 10] and in this study is applied for the first time to the ViSOM in the frame of a CBR.

It is based in the calculation of the “quality of adaptation” of a unit in the same position of different maps, in order to obtain the best characteristics vector in each of the units of the final one. This calculation is performed as follows:

$$V(p) = \frac{|x_p|}{\sum_{m=1}^M |x_{mp}|} \cdot \frac{q_p}{\sum_{m=1}^M q_{mp}} \quad (2)$$

The model, called WeVoS is described in detail in the *algorithm 1*.

Algorithm 1. Weighted Voting Superposition (WeVoS)

-
- 1: train several networks by using the bagging (re-sampling with replacement) meta-algorithm
 - 2: **for each map** (m) **in the ensemble**
 - 3: **for each unit position** (p) **of the map**
 - 4: calculate the quality measure/error chosen for the current unit
 - 5: **end**
 - 6: **end**
 - 7: calculate an accumulate total of the quality/error for each position $Q(p)$ in all maps
 - 8: calculate an accumulate total of the number of data entries recognized by a position in all maps $D(p)$
 - 9: **for each unit position** (p)
 - 10: initialize the fused map (fus) by calculating the centroid (w') of the units of all maps in that position (p)
 - 11: **end**
 - 12: **for each map** (m) **in the ensemble**
 - 13: **for each unit position** (p) **of the map**
 - 14: calculate the vote weight of the neuron (p) in the map (m) by using Eq. 2
 - 15: feed, to the fused map (fus), the weights vector of the neuron (p) as if it was an input to the network, 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.
 - This causes the unit of the final map (w') to approximate the unit of the composing ensemble (w_p) accordingly to its adaptation.
 - 16: **end**
 - 17: **end**
-

5 WeVoS-CBR: A New Oil Spill Forecasting System

There have already been CBR systems created to solve maritime problems [11] in which different maritime variables have been used. In this occasion, the data used have been previously collected from different observations from satellites, and then pre-processed, and structured to create the case base. The created cases are the main elements to obtain the correct solutions to future problems, through the CBR system. The developed system determines the probability of finding oil slicks in a certain area after an oil spill has been produced. To properly obtain the solutions, the system uses square divisions of the area to analyze of approximately half a degree side. Then the system calculates the number of slicks in every square as long as the surface covered by them.

The structure of a case in the presented system is composed by the values obtained in each square for the following 14 variables: longitude, latitude, date, sea height, bottom pressure, salinity, temperature, meridional wind, zonal wind, wind strength, meridional current, zonal current, current strength and area of slicks.

The described system includes different AI techniques to achieve the objectives of every CBR phase. Every CBR phase uses one or more AI techniques in order to obtain its solution. Those phases with its related techniques are going to be explained next.

5.1 Creating the Case Base and Recovering the Most Similar Cases

The data used to train the system has been obtained after the Prestige accident, between November 2002 and April 2003, in a specific geographical area to the north west of the Galician coast (longitude between 14 and 6 degrees west and latitude between 42 and 46 degrees north).

When the case base is created, the WeVoS algorithm is used to structure it. The graphical capabilities of this novel algorithm are used in this occasion to create a model that represents the actual variability of the parameters stored in the cases. At the same time, the inner structure of the case base will make it easier to recover the most similar cases to the problem cases introduced in the system.

The WeVos algorithm is also used to recover the most similar cases to the problem introduced in the system. That process is performed once the case base is structured keeping the original distribution of the available variables.

5.2 Adaptation of the Recovered Cases

After recovering the most similar cases to the problem from the case base, those cases are used to obtain a solution. *Growing RBF networks* [12] are used to generate the predicted solution corresponding to the proposed problem. The selected cases are used to train the GRBF network. This adaptation of the RBF network lets the system grow during the training phase in a gradual way increasing the number of elements (prototypes) which work as the centres of the radial basis functions. In this occasion the creation of the GRBF is automatically done, which implies an adaptation of the original GRBF system. The error definition for every pattern is shown below:

$$e_i = l/p \sum_{k=1}^p \|t_{ik} - y_{ik}\| \quad (3)$$

where t_{ik} is the desired value of the k^{th} output unit of the i^{th} training pattern, y_{ik} the actual values of the k^{th} output unit of the i^{th} training pattern. After the creation of the GRBF network, it is used to generate the solution to the introduced problem. The solution will be the output of the network using as input data the retrieved cases.

5.3 Revision and Retention of the Proposed Solution

In order to verify the precision of the proposed solution, *Explanations* are used [13]. To justify and validate the given solution, the retrieved cases are used once again. To create an explanation, different possibilities have been compared. The selected cases have their own future associated situation. Considering the case and its solution as two vectors, a distance between them can be measured by calculating the evolution of the situation in the considered conditions. If the distance between the proposed problem and the solution given is not bigger than the distances obtained from the selected cases, then the proposed solution considered as a good one, according to the structure of the case base.

Once the proposed prediction is accepted, it can be stored in the case base in order to serve to solve new problems. It will be used equally than the historical data previously stored in the case base. The *WeVoS* algorithm is used again to introduce new elements in the case base. After introducing a new case in the case base, the structure formed by the information stored in the case base, also change, to be adapted to the new situation created.

6 Results

To create the case base, data obtained from different satellites have been used. Temperature, salinity, bottom pressure, sea height, number and area of the slicks, as long as the date have been involved in the case base creation. All these data define the problem case and also the solution case. The problem solution is defined by the variables related to an area, but with the values of the different variables changed to the prediction obtained from the CBR system related to a future point.

The *WeVoS*-CBR system has been checked with a subset of the available data that has not been previously used in the training phase. The predicted situation was contrasted with the actual future situation as it was known (historical data was used to train the system and also to test its correction). The proposed solution was, in most of the variables, close to 90% of accuracy.

Table 1 shows a summary of the obtained results. In this table different techniques are compared. The table shows the evolution of the results along with the increase of the number of cases stored in the case base. All the techniques analyzed show an improvement in their results when the number of cases stored in the case base is increased. The “*RBF*” column corresponds to a simple Radial Basis Function Network trained with all the available information. The RBF network generates an output that is considered a solution to the problem giving it, as input, the problem to be solved. The “*CBR*” column represents a simple CBR system, with no artificial intelligence

techniques included. The cases are stored in the case bases and recovered considering the Euclidean distance. The most similar cases are selected and after applying a weighted mean depending on the similarity, a solution is proposed. It is a *mathematical* CBR. The “*RBF + CBR*” column corresponds to an approximation that uses a RBF system combined with a CBR methodology. The cases are recovered from the case base using the Manhattan distance to retrieve the most similar cases to the introduced problem. The RBF network is applied in the reuse phase, modifying the selected cases to generate the new solution. The results of the “*RBF+CBR*” column are, normally, better than those of the “*CBR*”, mainly due to the removal of worthless data to obtain the solution. Last, the “*WeVoS-CBR*” column shows the results obtained by the proposed system, being better than the three previous solutions analyzed. The solution proposed do not generate a trajectory, but a series of probabilities in different areas, what is far more similar to the real behaviour of the oil slicks.

Table 1. Percentage of good predictions obtained with different techniques

<i>Number of cases</i>	RBF	CBR	RBF + CBR	WeVoS-CBR
100	45 %	39 %	42 %	43 %
500	46 %	41 %	44 %	47 %
1000	49 %	46 %	55 %	63 %
2000	56 %	55 %	65 %	72 %
3000	58 %	57 %	66 %	79 %
4000	60 %	63 %	69 %	84 %
5000	60 %	62 %	73 %	88 %

7 Conclusions and Future Work

A new hybrid intelligent predicting system, based on the CBR methodology, to forecast the probability of finding oil slicks in an area after an oil spill is presented in this paper. The explained system uses information recovered from different orbital satellites. All that information has been used to create the CBR system. The available data is first structure and classified to create the case base, where the knowledge used by the system to generate its predictions is stored. To achieve the different tasks related with the phases of the CBR cycle, the developed system uses diverse artificial intelligence techniques. A new voting algorithm, the Weighted Voting Superposition algorithm is used both to organize the case base and to retrieve the most similar cases to the one introduced as a problem to the system. The great organization capabilities of that new algorithm allow the system to create a valid structure to the case base and to easily recover similar cases from the case base.

Growing Radial Basis Function Networks has been used to generate a prediction using the cases retrieved from the case base. Using this evolution of the RBF networks a better adaptation to the structure of the case base is obtained. The results using Growing RBF networks instead of simple RBF networks are about a 4% more accurate, which is a quite good advance.

It has been proved that the system can predict in already known conditions, showing better results than previously used techniques. The use of the blend of techniques

integrated in the CBR structure makes possible to obtain better results than using the CBR alone (17% better), and also better than using the techniques isolated, without the integration feature produced by the CBR (11% better).

The next step is generalising the learning, acquiring new data to create a base of cases big enough to have solutions for every season. Another improvement is to create an on-line system that can store the case base in a server and generate the solutions dynamically to different requests. This on-line version will include real time connection to data servers providing weather information of the current situations in order to predict *real future* situations.

Acknowledgements

This research has been partially supported by projects BU006A08 and SA071A08 of the JCYL and TIN2006-14630-C03-03 of the MEC.

References

1. Watson, I.: Case-Based Reasoning Is a Methodology Not a Technology. *Knowledge-Based Systems* 12, 303–308 (1999)
2. Baruque, B., Corchado, E.: WeVoS: A Topology Preserving Ensemble Summarization Algorithm. *Data Mining and Knowledge Discovery* (submitted)
3. Palenzuela, J.M.T., Vilas, L.G., Cuadrado, M.S.: Use of Asar Images to Study the Evolution of the Prestige Oil Spill Off the Galician Coast. *International Journal of Remote Sensing* 27, 1931–1950 (2006)
4. Solberg, A.H.S., Storvik, G., Solberg, R., Volden, E.: Automatic Detection of Oil Spills in Ers Sar Images. *IEEE Transactions on Geoscience and Remote Sensing* 37, 1916–1924 (1999)
5. Aamodt, A.: A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning. *Knowledge Engineering and Image Processing Group*. University of Trondheim (1991)
6. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7, 39–59 (1994)
7. Kohonen, T.: The Self-Organizing Map. *Neurocomputing* 21, 1–6 (1998)
8. Yin, H.: Data Visualisation and Manifold Mapping Using the Visom. *Neural Networks* 15, 1005–1016 (2002)
9. Baruque, B., Corchado, E., Yin, H.: Visom Ensembles for Visualization and Classification. In: Sandoval, F., Gonzalez Prieto, A., Cabestany, J., Graña, M. (eds.) *IWANN 2007*. LNCS, vol. 4507, pp. 235–243. Springer, Heidelberg (2007)
10. 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)
11. Corchado, J.M., Fdez-Riverola, F.: Fsfrr: Forecasting System for Red Tides. *Applied Intelligence* 21, 251–264 (2004)
12. Karayiannis, N.B., Mi, G.W.: Growing Radial Basis Neural Networks: Merging Supervised and Unsupervised Learning with Network Growth Techniques. *IEEE Transactions on Neural Networks* 8, 1492–1506 (1997)
13. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in Case-Based Reasoning—Perspectives and Goals. *Artificial Intelligence Review* 24, 109–143 (2005)

Clustering Likelihood Curves: Finding Deviations from Single Clusters

Claudia Hundertmark¹ and Frank Klawonn²

¹ Department of Cell Biology, Helmholtz Centre for Infection Research, Inhoffenstr. 7,
D-38124 Braunschweig, Germany

² Department of Computer Science, University of Applied Sciences
Braunschweig/Wolfenbuettel, Salzdahlumer Str. 46/48, D-38302 Wolfenbuettel, Germany

Abstract. For systematic analyses of quantitative mass spectrometry data a method was developed in order to reveal peptides within a protein, that show differences in comparison with the remaining peptides of the protein concerning their regulatory characteristics. Regulatory information is calculated and visualised by a probabilistic approach resulting in likelihood curves. On the other hand the algorithm for the detection of one or more clusters is based on fuzzy clustering, so that our hybrid approach combines probabilistic concepts as well as principles from soft computing. The test is able to decide whether peptides belonging to the same protein, cluster into one or more group. In this way obtained information is very valuable for the detection of single peptides or peptide groups which can be regarded as regulatory outliers.

Keywords: Clustering, iTRAQ™, LC-MS/MS, likelihood curve.

1 Introduction

Comparative analyses between normal and pathological states of biological systems are an important basis for biological and medical research. Therefore, quantitative analyses of biological components, e.g. proteins, are of particular importance. Proteins are the basic components of cells and responsible for most of the processes in organisms. The basic structure of proteins are amino acid chains, which can be digested into smaller chains termed peptides. The totality of proteins, called the proteome, is highly dynamic and can vary significantly concerning its qualitative and quantitative composition due to changed conditions. A common technology for the analysis of the proteome is called liquid-chromatography mass spectrometry (LC-MS/MS).

In the meantime besides protein identification, mass spectrometry enables relative peptide quantitation by LC-MS/MS. One of the most popular technologies for this purpose is called iTRAQ™, which is based on chemical labelling of peptides. iTRAQ™ allows comparative analyses of up to eight proteinogenic samples in parallel (see [1], [2]). After iTRAQ™-labelling of peptides from different samples and subsequently LC-MS/MS analysis a so-called mass spectrum is available for every detected peptide. The obtained mass spectrum contains information on the

amino acid sequence as well as information on the relative concentrations of the actually measured peptide in all analysed samples. Concentration of a peptide is given by a so-called “intensity”. In order to compare the concentrations of a peptide under different conditions, a ratio is calculated by dividing the intensities measured from the different samples (or conditions). The resulting ratio is termed “expression ratio” and “regulation factor”, respectively.

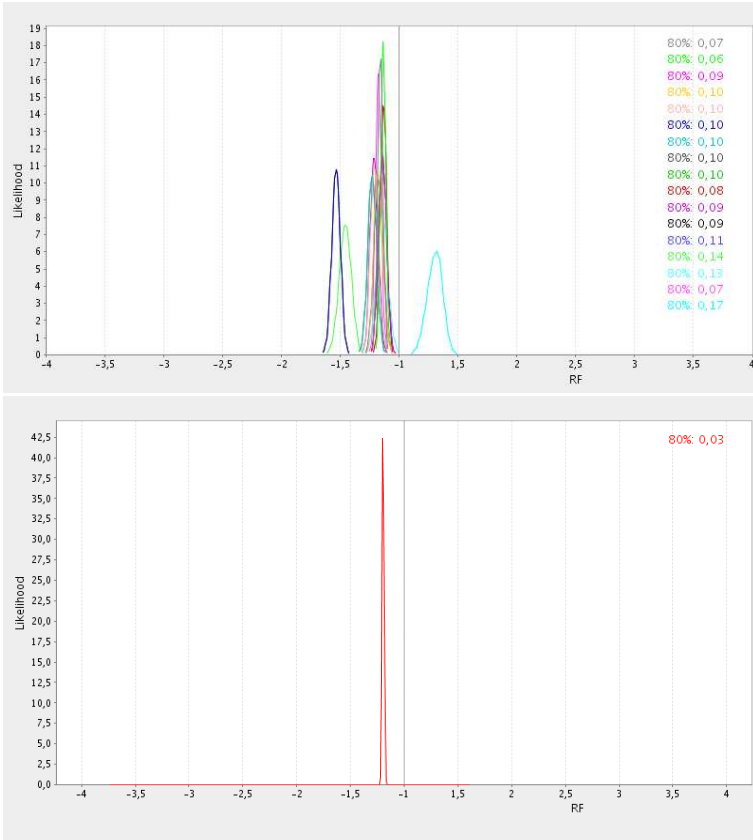


Fig. 1. Likelihood curves represent the most suitable regulation factor of peptides (top) and proteins (bottom), respectively as well as robustness of regulatory information. High iTRAQ™ intensities correlate positively with the robustness and thus result in narrow likelihood curves on the one hand and in small IR values on the other hand.

In several studies noise was observed in iTRAQ™ analysed data. Particularly if a peptide is detected with low intensities, impreciseness of the calculated ratio is significantly higher than in the case of peptides, that were measured with high intensities. This effect is called “intensity-dependent noise”. Based on this observation we developed a mathematical model for the estimation of noise inherent in quantitative data analysed from iTRAQ™-labelled peptides analysed by

LC-MS/MS (for details see [3]). Subsequently, we were able to derive several applications from our noise model, e.g. likelihood curves for the calculation of the most suitable regulation factor for single peptides and total proteins as a collective of all contributing peptides ([4]). Furthermore, likelihood curves provide evaluation of robustness of the calculated regulation factor, which again is strongly depending on measured intensities.

Depending on the aim of performed analysis various views on a protein may be useful. Peptides can be visualised separately by individual likelihood curves or they can be combined and visualised by a shared protein likelihood curve. Fig. 1 shows both peptide view (top) and protein view (bottom) for the same protein. Likelihood curves give the likelihoods for regulation factors, which are deviating from the calculated ratio to a greater or lesser extent. X-axis refers to regulation factors, y-axis refers to the likelihood.

Robustness of the underlying data is proportional with both the height and the slope of the produced curve. Furthermore, robustness is represented numerically by the interval of robustness (IR), which is given on the upper right hand side within each plot. IR gives the length of the minimal interval that is covered by 80% of the area beyond each area-normalised likelihood curve (normalised to $\int = 1$). All peptide curves are fairly robust (IR between 0.06 and 0.17) and most of them are fluctuating near -1.2 (no regulation). Robustness of the resulting protein curve of all peptides is extremely high (IR = 0.03) and the regulation factor approximates the regulation factor of the majority of peptide curves.

2 Distance Measure

Studying regulatory information of peptides, which is a common practice in biology (especially in proteomics), requires the detection of peptides, that differ in their regulatory characteristics from the remaining peptides of the same protein. Studies like this are important in order to reveal mismatched (by software) peptides or for the investigation of special peptide modifications. For the analysis of large protein samples resulting in very large datasets (several hundreds of proteins) cluster analysis for the automatic identification of proteins consisting of more than one group of peptides concerning their regulatory behaviour is a very helpful means. In the majority of cases all peptides, that belong to the same protein, fluctuate near the same regulation factor and therefore build one cluster. Hence, distinguishing between proteins containing one cluster of peptides and those, which have more than one peptide cluster is the main focus in this approach.

For the following cluster analysis a distance measure d_{ij} giving the distance of two elements i (prototype) and j (peptide j) is to be defined. In order to compare area-normalised likelihood curves i and j the size of the overlapping area a is the distance measure if there is an overlap of curves. Then the distance d_{ij} is given by

$$d_{ij} = \begin{cases} 0 \leq 1 - a \leq 1 & \text{partial overlap of curves } i \text{ and } j \\ 0 & \text{total overlap of curves } i \text{ and } j \end{cases} \quad (1)$$

with a = size of overlapping areas of curves i and j .

In the case of non-overlapping curves i, j d_{ij} is defined by the distance with regard to the scaling of the x-axis which is given by the distance of the highest calculated regulation factor x_{max} of the lower regulated element el_1 and the lowest calculated regulation factor x_{min} of the higher regulated element el_2 . Since $d_{ij} = 1$ for curves without overlap, this is the minimum value for non-overlapping curves and must be added to the calculated distance. Therefore, in the case of non-overlapping curves i and j the distance measure d_{ij} is given by

$$d_{ij} = x_{min}^{(el_2)} - x_{max}^{(el_1)} + 1 \quad (2)$$

3 Testing for the Existence of a Single Cluster

In search of proteins with significant differences in regulation of related peptides proteins are out of interest for further analyses, which consist of only one cluster. Those proteins are to be identified and can be discarded for following investigation. Therefore, we developed a method for the detection of consistently regulated proteins in order to obtain the remaining proteins consisting of differentially regulated peptides. For these purposes we created a hybrid approach. We combined the probabilistic proceeding for the calculation of protein and peptide likelihood curves with a prototype based fuzzy clustering method .

Our approach for the detection of one-cluster-proteins is based on the generation of a prototype for all regarded peptide likelihood curves and subsequently removing the most distant peptide curve in terms of the prototype curve. In this way the set of likelihood curves is reduced little by little and the distances of the removed curves are plotted by the means of a scatterplot. Analysis of entries within the scatterplot returns the result whether there is only one cluster or not. In detail the procedure is as follows:

First of all, a prototype curve representing all likelihood curves of the protein is calculated. Since the distance measure d_{ij} is based on area overlaps the prototype curve must consist of those parts of all peptide likelihood curves, where most of the peptide curves are overlapping. According to likelihood curves the prototype curve has to consist of connected areas and the likelihood must always be greater than zero at every discretised regulation factor. In detail, calculation of the prototype is done by the detection of the number of overlapping curves for every area that is located below one curve at least. Afterwards, areas, that are part of the highest number of likelihood curves are added to the prototype curve subsequently. This process is finished as soon as the total area of the prototype curve reaches 1.

The second step consists in repeatedly calculating the distances d_{ij} of every peptide likelihood curve and the prototype by application of (1) and (2) and removing the most distant object until there are no likelihood curves left. The respective distances of the removed curves are plotted against the number of iterations within a scatterplot.

Finally, the scatterplot shows whether the protein consists of only one or more different regulatory peptide clusters. In the case of one cluster the entries are arranged homogeneously and close to each other. In the case of multiple clusters on the other hand, there are groups of entries. After removing the last curve of a cluster a

significant step is observable in the scatterplot. For automatic detection of steps a threshold is defined: A step is regarded to be significant if the absolute value between the new entry e_{t+1} and the previous entry e_t is greater than the fivefold mean difference $\overline{\Delta}_c$ from the mean value \overline{x}_c within the actual regarded (or lost) cluster. Hence,

$$step = \begin{cases} \text{significant} & \text{if } |e_{t+1} - e_t| > 5\overline{\Delta}_c \\ \text{not significant} & \text{else} \end{cases} \quad (3)$$

It should be noted that the distance of two curves within a cluster is always smaller than 1.

The following figures illustrate this effect: Fig. 2 shows six peptide likelihood curves for a protein, which are clearly clustering into two groups. The corresponding scatterplot (Fig. 2 insert) shows two groups as well. According to the likelihood plot, every group is composed of three entries. The step between the third and the fourth entry identifies the loss of the last curve from one cluster and is equivalent to the distance of both clusters.

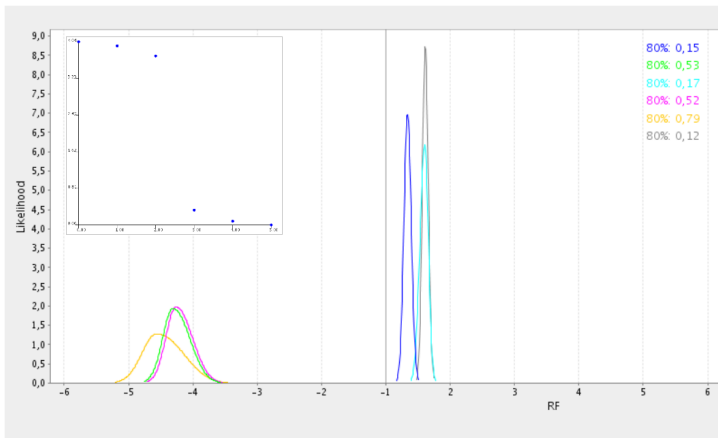


Fig. 2. Likelihood plot consisting of six peptide curves, which are clustering into two groups. Insert: The corresponding scatterplot clearly indicates that the protein is composed of more than one peptide cluster.

Fig. 3 gives an example for a protein, whose peptides are regulated very similarly. Therefore, our method returns that only one cluster is found, which is indicated by one group of entries that are located close to each other. For comparability the scaling of y-axis is the same as in Fig. 2.

Unfortunately, the test is not able to differentiate between different clusters, whose distances to the main cluster are similar. Since the test is using distances without consideration of the localisation of the likelihood curves (lower or higher regulated than the main cluster) two similar distant clusters result in a common step in the

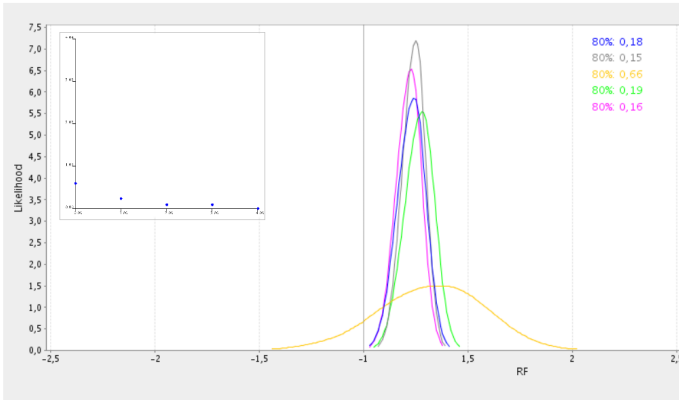


Fig. 3. Likelihood plot consisting of five peptide curves, which are clustering into one group. Insert: The corresponding scatterplot clearly indicates that the protein is composed of one peptide cluster.

scatterplot. Therefore, the test is not suitable for the determination of the number of clusters, but only for the discrimination of one cluster on the one hand and multiple clusters on the other hand. An example is given in Fig. 4: Next to the main cluster (in the middle) are additional clusters on the left as well as on the right. Both additional clusters are located very closely to the main cluster but they are not overlapping. Regarding the inserted scatterplot, it can be observed clearly that four curves are removed in the beginning and after a step the remaining curves are considered. Curves removed first originate from both additional clusters (the left cluster consists of three curves which can not be kept apart visually in the likelihood plot).

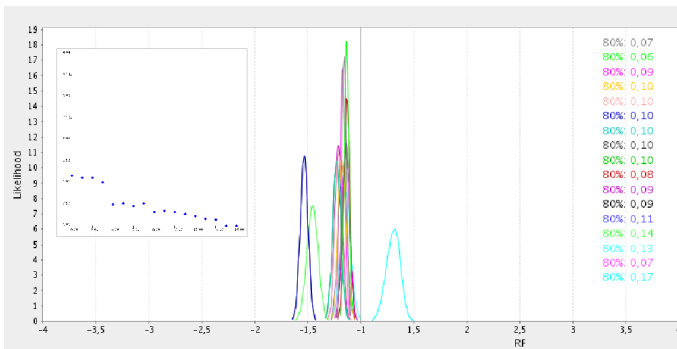


Fig. 4. Likelihood plot consisting of 17 peptide curves, which are clustering into three groups. The leftmost cluster is composed of three curves, the cluster in the middle consists of 13 peptide curves and the rightmost cluster contains one curve. Distances between the main cluster in the middle and the leftmost and the rightmost cluster respectively are more or less equal. Insert: The corresponding scatterplot clearly indicates that the protein is composed of more than one peptide cluster.

4 Conclusion

We have presented an approach for clustering complex data types in the form of likelihood curves. Our method has the advantage that it can cope with small datasets as well as with a small number of cluster, even one cluster. Determining the number of clusters based on validity measures as it is for instance done in fuzzy cluster analysis (see for instance [5,6]) is not suitable for such datasets. Although our approach as it is presented here is only designed to distinguish between one cluster and more than one cluster, our method can also be applied to determine the number of clusters in the following. In case more than one cluster is detected in the dataset, the main cluster corresponding to the likelihood curves after the last significant drop in the scatter plot like in Fig. 2 is removed from the dataset and our method is applied again to the remaining data. A similar idea of subtractive clustering based on removing distant data points, however for standard datasets, but not for likelihood curves, has also been proposed in [7,8].

Acknowledgments. We are very grateful for the support provided by Dr. Lothar Jänsch and Dr. Tobias Reinl.

References

1. Ross, P.L., Huang, Y.N., Marchese, J.N., Williamson, B., Parker, K., Hattan, S., Khainovski, N., Pillai, S., Dey, S., Daniels, S., Purkayastha, S., Juhász, P., Martin, S., Bartlett-Jones, M., He, F., Jacobsen, A., Pappin, D.J.: Multiplexed protein quantitation in *Saccharomyces cerevisiae* using amine-reactive isobaric tagging reagents. *Mol. Cell. Proteomics* 3, 1154–1169 (2004)
2. Pierce, A., Unwin, R.D., Evans, C.A., Griffiths, S., Carney, L., Zhang, L., Jaworska, E., Lee, C.-F., Blinco, D., Okoniewski, M.J., Miller, C.J., Bitton, D.A., Spooncer, E., Whetton, A.D.: Eight-channel iTRAQ enables comparison of the activity of 6 leukaemogenic tyrosine kinases. *Mol. Cell. Proteomics* (2007)
3. Klawonn, F., Hundertmark, C., Jänsch, L.: A Maximum Likelihood Approach to Noise Estimation for Intensity Measurements in Biology. In: Proc. Sixth IEEE International Conference on Data Mining Workshops ICDM Workshops 2006, pp. 180–184 (2006)
4. Hundertmark, C., Fischer, R., Reinl, T., May, S., Klawonn, F., Jänsch, L.: MS-specific noise model reveals the potential of iTRAQTM in quantitative proteomics (in preparation)
5. Bezdek, J.C., Keller, J., Krishnapuram, R., Pal, N.R.: *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Kluwer, Boston (1999)
6. Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: *Fuzzy Cluster Analysis*. Wiley, Chichester (1999)
7. Georgieva, O., Klawonn, F.: Cluster Analysis via the Dynamic Data Assigning Assessment Algorithm. *Information Technologies and Control*, 14–21 (2006)
8. Klawonn, F., Georgieva, O.: Identifying Single Clusters in Large Data Sets. In: Wang, J. (ed.) *Encyclopedia of Data Warehousing and Mining*, pp. 180–183. Idea Group, Hershey (2006)

Unfolding the Manifold in Generative Topographic Mapping

Raúl Cruz-Barbosa^{1,2} and Alfredo Vellido¹

¹ Universitat Politècnica de Catalunya, Jordi Girona, 08034, Barcelona, Spain

² Universidad Tecnológica de la Mixteca, Huajuapán 69000, Oaxaca, México
{rcruz, avellido}@lsi.upc.edu

Abstract. Generative Topographic Mapping (GTM) is a probabilistic latent variable model for multivariate data clustering and visualization. It tries to capture the relevant data structure by defining a low-dimensional manifold embedded in the high-dimensional data space. This requires the assumption that the data can be faithfully represented by a manifold of much lower dimension than that of the observed space. Even when this assumption holds, the approximation of the data may, for some datasets, require plenty of folding, resulting in an entangled manifold and in breaches of topology preservation that would hamper data visualization and cluster definition. This can be partially avoided by modifying the GTM learning procedure so as to penalize divergences between the Euclidean distances from the data to the model prototypes and the corresponding geodesic distances along the manifold. We define and assess this strategy, comparing it to the performance of the standard GTM, using several artificial datasets.

1 Introduction

Amongst density-based methods, Finite Mixture Models have established themselves as flexible and robust tools for multivariate data clustering [1]. In many practical data mining scenarios, though, the available knowledge concerning the cluster structure of the data may be quite limited. In these cases, data exploration techniques are valuable tools and, amongst them, multivariate data visualization can be of great help by providing the analyst with intuitive cues about data structural patterns.

In order to endow Finite Mixture Models with data visualization capabilities, certain constraints must be enforced. One alternative is forcing the mixture components to be centred in a low-dimensional manifold embedded into the usually high-dimensional observed data space. Such approach is the basis for the definition of Generative Topographic Mapping (GTM) [2], a flexible manifold learning model for simultaneous data clustering and visualization whose probabilistic nature makes possible to extend it to perform tasks such as missing data imputation [3], robust handling of outliers, and unsupervised feature selection [4], amongst others.

The density estimation performed by GTM does not necessarily account for the intrinsic geometric structure of the data. The standard Gaussian GTM orients the density mass in the neighbourhood of a data point along all the directions in the input

space rather than only along the manifold and it may grant too much probability to irrelevant directions of space and not enough to directions along the manifold.

The approximation of certain datasets of complex structure may require plenty of folding from the GTM model, resulting in an unduly entangled embedded manifold and breaches of topology preservation that would hamper both the visualization of the data and the definition of clusters the model is meant to provide. Following an idea proposed in [5], in turn inspired in [6], the learning procedure of GTM is here modified to avoid, at least partially, this undesired effect by penalizing the divergences between the Euclidean distances from the data to the model prototypes and the corresponding approximated geodesic distances along the manifold. In this paper, we define and assess the resulting Geo-GTM model, which incorporates the data visualization capabilities that the model proposed in [5] lacks, and compare its performance to that of the standard Gaussian GTM, using several artificial datasets.

2 Manifolds and Geodesic Distances

Manifold methods such as ISOMAP [7] and Curvilinear Distance Analysis [8] use the geodesic distance as a basis for generating the data manifold. This metric favours similarity along the manifold, which may help to avoid some of the distortions that the use of a standard metric such as the Euclidean distance may introduce when learning the manifold. In doing so, it can avoid the breaches of topology preservation that may occur due to excessive folding.

The otherwise computationally intractable geodesic metric can be approximated by graph distances [9], so that instead of finding the minimum arc-length between two data points lying on a manifold, we would set to find the shortest path between them, where such path is built by connecting the closest successive data points. In practice, this is achieved by using either the ϵ -rule or the K -rule. The ϵ -rule allows connecting data points x and y whenever $\|x-y\| < \epsilon$, for some $\epsilon > 0$. The K -rule allows connecting the K -nearest neighbours. A weighted graph is then constructed by using the data and the set of allowed connections. The data are the vertices, the allowed connections are the edges, and the edge labels are the Euclidean distances between the corresponding vertices. If the resulting graph is disconnected, some edges are added using a minimum spanning tree procedure in order to fully connect it. Finally, the distance matrix of the weighted undirected graph is obtained by repeatedly applying Dijkstra's algorithm [10], which computes the shortest path between all data samples.

3 GTM and Geo-GTM

3.1 The GTM Model

The standard GTM is a non-linear latent variable model defined as a mapping from a low dimensional latent space (usually 2-dimensional, as in this study, for visualization purposes) onto the multivariate data space. The mapping is carried through by a set of basis functions generating a constrained mixture density distribution. It is defined as a generalized linear regression model:

$$\mathbf{y}=\Phi(\mathbf{u})\mathbf{W}, \tag{1}$$

where Φ are R basis functions, Gaussians in the standard formulation; \mathbf{W} is a matrix of adaptive weights w_{rd} ; and \mathbf{u} is a point in the 2-dimensional latent variable space. To avoid computational intractability, a regular grid of M points \mathbf{u}_m can be sampled from the space defined by the latent variables of the model. Each of them, which can be considered as the representative of a data cluster, has a fixed prior probability $p(\mathbf{u}_m) = 1/M$ and is mapped, using Eq. 1, into a low dimensional manifold non-linearly embedded in the data space. This latent space grid is similar in design and purpose to that of the visualization space of the SOM. A probability distribution for the multivariate data $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ can then be defined, leading to the following expression for the log-likelihood:

$$L(\mathbf{W}, \beta|\mathbf{X}) = \sum_{n=1}^N \ln \left\{ \frac{1}{M} \sum_{m=1}^M \left(\frac{\beta}{2\pi} \right)^{D/2} \exp \left\{ -\frac{\beta}{2} \|\mathbf{y}_m - \mathbf{x}_n\|^2 \right\} \right\}, \tag{2}$$

where \mathbf{y}_m , usually known as *reference* or *prototype* vectors, are obtained for each \mathbf{u}_m using Eq. 1; and β is the inverse of the noise variance, which accounts for the fact that data points might not strictly lie on the low dimensional embedded manifold generated by the GTM. The Expectation Maximization (EM) algorithm is a straightforward alternative to obtain the Maximum Likelihood estimates of the adaptive parameters of the model, namely \mathbf{W} and β . In the E-step, the responsibilities z_{mn} (the probability of cluster m membership for data point \mathbf{x}_n) are computed as

$$z_{mn} = p(\mathbf{u}_m|\mathbf{x}_n, \mathbf{W}, \beta) = \frac{p(\mathbf{x}_n|\mathbf{u}_m, \mathbf{W}, \beta)p(\mathbf{u}_m)}{\sum_{m'} p(\mathbf{x}_n|\mathbf{u}_{m'}, \mathbf{W}, \beta)p(\mathbf{u}_{m'})}, \tag{3}$$

where $p(\mathbf{x}_n|\mathbf{u}_m, \mathbf{W}, \beta) = N(\mathbf{y}(\mathbf{u}_m, \mathbf{W}), \beta)$.

3.2 Geo-GTM

The Geo-GTM model is an extension of GTM that favours the similarity of points along the learned manifold, while penalizing the similarity of points that are not contiguous in the manifold, even if close in terms of the Euclidean distance. This is achieved by modifying the standard calculation of the responsibilities in Eq. 3 proportionally to the discrepancy between the geodesic (approximated by the graph) and the Euclidean distances. Such discrepancy is made operational through the definition of the exponential distribution

$$\varepsilon(d_g|d_e, \alpha) = \alpha^{-1} \exp \left\{ -\frac{d_g(\mathbf{x}_n, \mathbf{y}_m) - d_e(\mathbf{x}_n, \mathbf{y}_m)}{\alpha} \right\}, \tag{4}$$

where $d_e(\mathbf{x}_n, \mathbf{y}_m)$ and $d_g(\mathbf{x}_n, \mathbf{y}_m)$ are, in turn, the Euclidean and graph distances between data point \mathbf{x}_n and the GTM prototype \mathbf{y}_m . Responsibilities are redefined as:

$$z_{mn}^{geo} = p(\mathbf{u}_m | \mathbf{x}_n, \mathbf{W}, \beta) = \frac{p'(\mathbf{x}_n | \mathbf{u}_m, \mathbf{W}, \beta) p(\mathbf{u}_m)}{\sum_m p'(\mathbf{x}_n | \mathbf{u}_m, \mathbf{W}, \beta) p(\mathbf{u}_m)}, \quad (5)$$

where $p'(\mathbf{x}_n | \mathbf{u}_m, \mathbf{W}, \beta) = N(\mathbf{y}(\mathbf{u}_m, \mathbf{W}), \beta) \mathcal{E}(d_g(\mathbf{x}_n, \mathbf{y}_m)^2 | d_e(\mathbf{x}_n, \mathbf{y}_m)^2, 1)$.

When there is no agreement between the graph approximation of the geodesic distance and the Euclidean distance, the value of the numerator of the fraction within the exponential in Eq. 4 increases, pushing the exponential and, as a result, the modified responsibility, towards smaller values, i.e., punishing the discrepancy of metrics. Once the responsibility is calculated in the modified E-step, the rest of the model's parameters are estimated following the standard EM procedure.

The only additional computational effort incurred by Geo-GTM is the result of building a graph and the computation of its corresponding distance matrix, which is calculated only once before the EM algorithm is run. The dominant computational complexity of Geo-GTM is therefore similar to that of GTM.

3.3 Data Visualization Using Geo-GTM

The GTM was originally defined as a probabilistic alternative to the SOM. As a result, the data visualization capabilities of the latter are fully preserved and even augmented by the former. The main advantage of GTM and any of its extensions over general finite mixture models consists precisely on the fact that both data and results can be intuitively visualized on a low dimensional representation space.

Each of the cluster representatives \mathbf{u}_m in the latent visualization space is mapped, following Eq. 1, into a point \mathbf{y}_m belonging to a manifold embedded in data space. Given that the posterior probability of every Geo-GTM cluster representative for being the generator of each data point, or responsibility z_{mn}^{geo} , is calculated as part of the modified EM algorithm, both data points and cluster prototypes can be visualized as a function of the latent point locations as the mean of the estimated posterior distribution: $\mathbf{u}_n^{mean} = \sum_{m=1}^M \mathbf{u}_m z_{mn}^{geo}$.

4 Experiments

The experimental settings and visualization results for four artificial datasets are reported in this section.

4.1 Experimental Settings

Geo-GTM was implemented in MATLAB®. For the experiments reported next, the adaptive matrix \mathbf{W} was initialized, following a procedure described in [2], as to minimize the difference between the prototype vectors \mathbf{y}_m and the vectors that would be generated in data space by a partial Principal Component Analysis (PCA). The inverse variance β was initialised to be the inverse of the 3rd PCA eigenvalue. This initialization ensures the replicability of the results. The latent grid was fixed to a

square layout of approximate size $(N/2)^{1/2} \times (N/2)^{1/2}$, where N is the number of points in the dataset. The corresponding grid of basis functions was equally fixed to a 5×5 square layout for all datasets.

4.2 Results and Discussion

The main goal of the experiments reported next was assessing whether the proposed Geo-GTM model could capture and visually represent the underlying structure of datasets generated from smooth curves and surfaces of convoluted geometry better than the standard GTM.

Four artificial 3-dimensional datasets, represented in Fig. 1, were used in the experiments. Datasets of higher dimensionality were not used at this stage in order to better illustrate the effect of the proposed model on the visualization of the data. The first one is the *Swiss-Roll* dataset, consisting on 1,000 randomly sampled data points generated by the function: $(x_1, x_2, x_3) = (t \cos(t), t_2, t \sin(t))$, where t and t_2 follow $U(3\pi/2, 9\pi/2)$ and $U(0,21)$, respectively. The second dataset, herein called *Two-Spirals*, consists of two groups of 300 data points each that follow a similar distribution to that of *Swiss-Roll* although, this time, the first group follows the uniform distribution $U(3\pi/4, 9\pi/4)$, while the second group was obtained by rotating the first one by 180 degrees in the plane defined by the first two axes and translating it by 2 units along the third axis. The third dataset, herein called *Helix*, consists of 500 data points that are images of the function $\mathbf{x} = (\sin(4\pi t), \cos(4\pi t), 6t-0.5)$, where t follows $U(-2, 2)$. The fourth and final dataset, *Two-Helix*, is formed by two sub-groups of 300 data points each that are images of the functions $\mathbf{x}_1 = (\sin(4\pi t), \cos(4\pi t), 6t-0.5)$ and $\mathbf{x}_2 = (-\sin(4\pi t), -\cos(4\pi t), 6t-0.5)$, where t follows $U(-1, 1)$. Noise was added to all datasets but *Two-Helix*: Gaussian noise of zero mean and standard deviation (σ) of 0.05 was added to *Swiss-Roll* and *Two-Spirals*. A higher level of Gaussian noise ($\sigma = 0.20$) was added to *Helix*.

The posterior mean distribution visualization maps for all datasets are displayed in Figs. 2 to 5. Geo-GTM, in Fig. 2, is shown to capture the spiral structure of *Swiss-Roll* far better than standard GTM, which misses it at large and generates a poor data visualization with huge overlapping between non-contiguous areas of the data. A similar situation is reflected in Fig. 3: The two segments of *Two-Spirals* are neatly separated by Geo-GTM, whereas the standard GTM suffers a breach of topology preservation due to folding that manifests itself in the lack of contiguity of the segment represented by circles and also by the overlapping of part of the data of both segments. The results are even more striking for *Helix*, as shown in Fig. 4: the helicoidal structure is preserved by Geo-GTM, whereas it is mostly missed by the standard GTM. The former also faithfully preserves data contiguity, whereas many breaches of contiguity hinder the visualization generated by the latter. Remarkably, Geo-GTM is much less affected by noise than the standard GTM, as it recovers, in an almost perfect manner, the underlying noise-free helix. This is probably due to the fact that this model favours directions along the manifold, minimizing the impact of off-manifold noise.

The visualization of *Two-Helix* in Fig. 5 fully confirms the previous results: Geo-GTM faithfully represents the underlying helicoids as two symmetric and

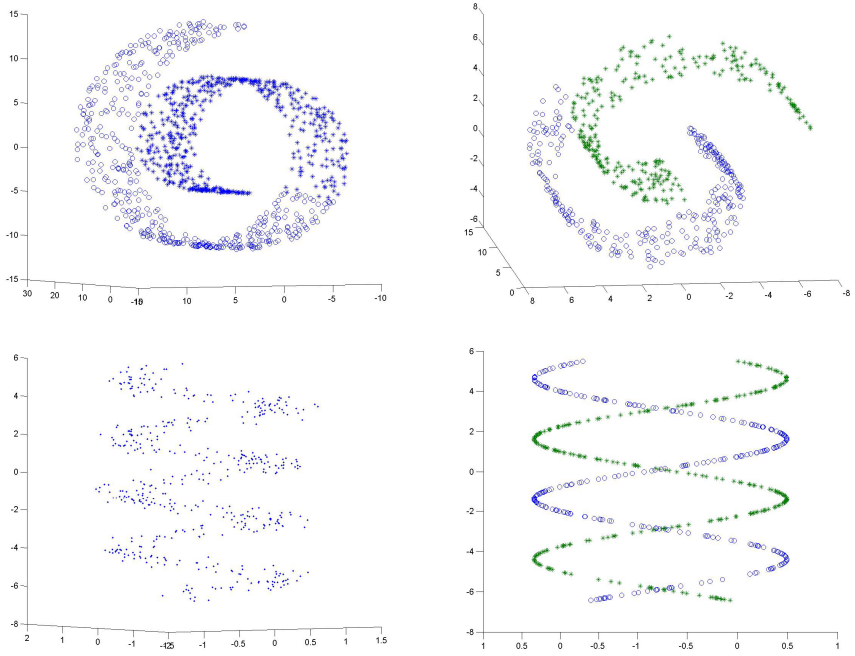


Fig. 1. The four datasets used in the experiments. (Top left): *Swiss-Roll*, where two contiguous fragments are identified with different symbols in order to check manifold contiguity preservation in Fig. 2. (Top right): *Two-Spirals*, again with different symbols for each of the spiral fragments. (Bottom left): *Helix*. (Bottom right): *Two-Helix*.

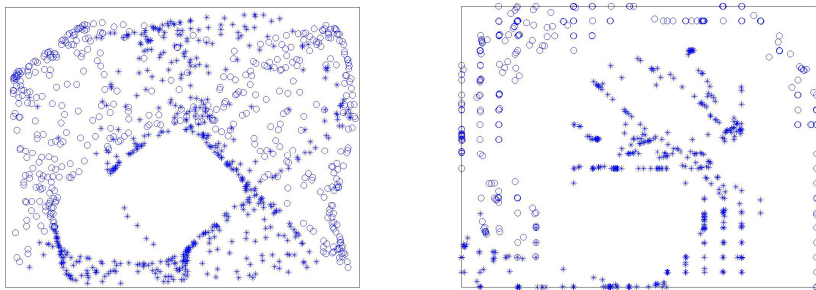


Fig. 2. Data visualization maps for the *Swiss-Roll* set. (Left): standard GTM; (right): Geo-GTM.

contiguity-preserving smooth curves. On the contrary, the standard GTM represents these data as several unconnected curves, a result of the breaches of contiguity produced by incorrect folding.

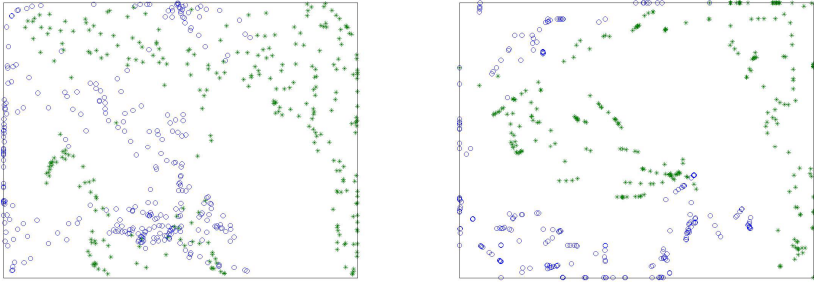


Fig. 3. Visualization maps for the *Two-Spirals* set. (Left): standard GTM; (right): Geo-GTM.

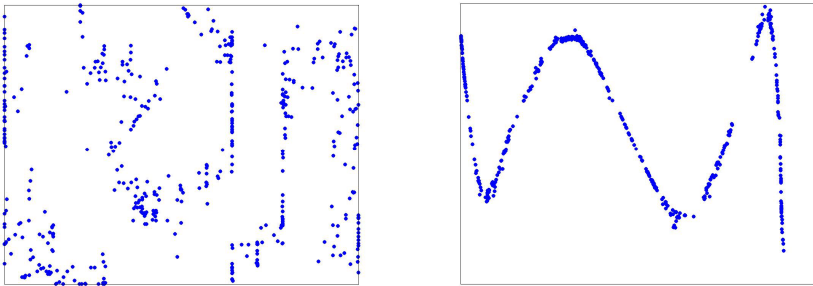


Fig. 4. Data visualization maps for the *Helix* set. (Left): standard GTM; (right): Geo-GTM.

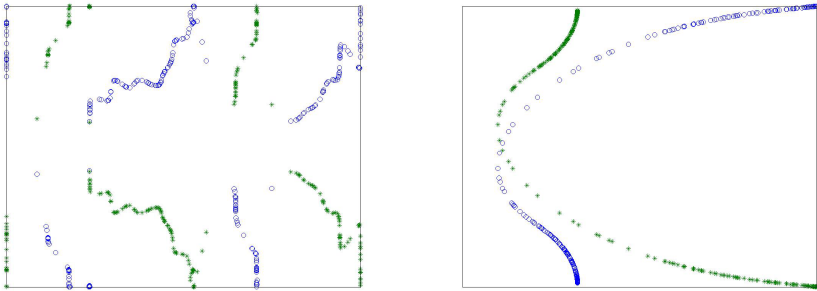


Fig. 5. Data visualization maps for the *Two-Helix* set. (Left): standard GTM; (right): Geo-GTM.

5 Conclusion

In this brief paper, we have introduced a novel variation of the manifold learning GTM model, namely Geo-GTM, which is able to faithfully recover and visually represent the underlying structure of datasets generated from smooth curves and surfaces of convoluted geometry. It does so by limiting the effect of manifold folding through

the penalization of the discrepancies between inter-point Euclidean distances and the approximation of geodesic distances along the model manifold. Formal quantitative measures of topography preservation will be considered in the next stages of the research.

The performance of Geo-GTM has been favorably assessed in several experiments with control artificial data. Moreover, Geo-GTM has been shown to recover the true underlying data structure even in the presence of noise. This capability of the model will also be investigated in detail in future research. Also, future research will involve more extensive experimentation, including a wider variety of datasets, both real and artificial, and of higher dimensionality.

Acknowledgements

Alfredo Vellido is a researcher within the Ramón y Cajal program of the Spanish MEC and acknowledges funding from the MEC I+D project TIN2006-08114. Raúl Cruz-Barbosa acknowledges SEP-SESIC (PROMEP program) of México for his PhD grant.

References

1. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(3), 381–396 (2002)
2. Bishop, C.M., Svensén, M., Williams, C.K.I.: The Generative Topographic Mapping. *Neural Computation* 10(1), 215–234 (1998)
3. Vellido, A.: Missing data imputation through GTM as a mixture of t-distributions. *Neural Networks* 19(10), 1624–1635 (2006)
4. Vellido, A., Lisboa, P.J.G., Vicente, D.: Robust analysis of MRS brain tumour data using t-GTM. *Neurocomputing* 69(7-9), 754–768 (2006)
5. Archambeau, C., Verleysen, M.: Manifold constrained finite Gaussian mixtures. In: Caibestany, J., Gonzalez Prieto, A., Sandoval, F. (eds.) *IWANN 2005*. LNCS, vol. 3512, pp. 820–828. Springer, Heidelberg (2005)
6. Vincent, P., Bengio, Y.: Manifold parzen windows. In: Thrun, S., Becker, S., Obermayer, K. (eds.) *NIPS*, vol. 15, pp. 825–832. MIT Press, Cambridge (2003)
7. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)
8. Lee, J.A., Lendasse, A., Verleysen, M.: Curvilinear Distance Analysis versus Isomap. In: *Proceedings of European Symposium on Artificial Neural Networks (ESANN)*, pp. 185–192 (2002)
9. Bernstein, M., de Silva, V., Langford, J., Tenenbaum, J.: Graph approximations to geodesics on embedded manifolds. Technical report, Stanford University, CA (2000)
10. Dijkstra, E.W.: A note on two problems in connection with graphs. *Numerische Mathematik* 1, 269–271 (1959)

Evaluation of Subspace Clustering Quality

Urszula Markowska-Kaczmar and Arletta Hurej

Wroclaw University of Technology, Institute of Applied Informatics,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
urszula.markowska-kaczmar@pwr.wroc.pl

Abstract. Subspace clustering methods seek to find clusters in different subspaces within a data set instead of searching them in full feature space. In such a case there is a problem how to evaluate the quality of the clustering results. In this paper we present our method of the subspace clustering quality estimation which is based on adaptation of *Davies-Bouldin Index* to subspace clustering. The assumptions which were made to build the metrics are presented first. Then the proposed metrics is formally described. Next it is verified in an experimental way with the use of our clustering method IBUSCA. The experiments have shown that its value reflects a quality of subspace clustering thus it can be an alternative in the case where there is no expert's evaluation.

1 Introduction

Data clustering in the full feature space, based on measuring the distance between objects using values of all attributes, works well in analysis of data sets with a small number of features. It is crucial to notice that features which distribution of values over the data set is nearly uniform, have no special influence on partitioning the objects into clusters. In traditional clustering methods like: k-means, hierarchical clustering, DBSCAN, etc. [3,1,5] they are taken into account during calculating the distance between instances. This phenomenon may have negative impact on the clustering results, and is called *curse of dimensionality*.

This problem becomes very important in the analysis of data sets where objects form clusters in different subspaces, as for instance in microarray gene expression results. Traditional data clustering algorithms are unable to limit the subspace to only these attributes that are important with regard to a cluster existence. These drawbacks have led to the concept of *subspace clustering* algorithms which are able to simultaneously find clusters existing in various subspaces of attributes. In such a case it is difficult to use the same metrics evaluating quality of clustering as in the full feature space clustering. The difficulties in creation such metrics are signaled in the literature [8]. It would be especially useful in the case when an expert is not available or it can be used to compare results of different subspace clustering methods.

To fulfill this gap we have proposed the metrics SCQE (*Subspace Clustering Quality Estimate*) which is presented in this paper. The metrics is calculated in a heuristic way. This heuristics applied to subspace clustering can be perceived as a hybrid combination of both methods.

The paper is organized as follows. In the next section short description of IBUSCA subspace clustering method is presented, the clustering results of which were evaluated with the use of the proposed SCQE metrics. Then our subspace quality clustering estimate SCQE metrics is introduced. In the subsequent section experimental results are shown and analyzed. The conclusion summarizes the paper and shows some future works.

2 IBUSCA – An Example of Subspace Clustering Method

To show the problems that arrive in evaluation of the subspace clustering results in this section we will focus on the example of subspace clustering method – IBUSCA [4]. The reader interested in other subspace clustering methods is referred to the comparative study of the subspace clustering algorithms, for instance [7] and [5].

A bottom-up approach is used in IBUSCA to find dense regions in single attributes and merge them into the new higher dimensional units that may form a cluster. One-dimensional dense regions are found using data histogram analysis. It is important to compute histograms that bring out and reflect data distribution in a given attribute faithfully. IBUSCA evaluates the bin width for a histogram of numerical attribute as directly proportional to the standard deviation of values and inversely proportional to the number of instances in the data set [9].

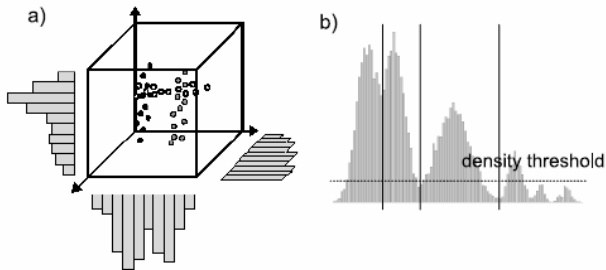


Fig. 1. IBUSCA – a) calculation of histograms for each dimension; b) histogram of numerical attribute with more splitting points

After computing a histogram for each attribute (fig. 1. a)) the data distribution in each attribute is analyzed to find meaningful differences in bins density, which separate clusters included in the attribute. Dense units in IBUSCA are created by dividing the domains of attributes into dense units by looking for the boundaries with the lowest density located between high density regions (fig. 1. b)).

The measure of relevance of density differences between histogram bins is defined on the basis of χ^2 statistical significance test to find splitting points between dense regions and separating domain into dense units.

An example of the histogram for numerical attribute with multiple splitting points is shown in fig. 1. b). Two consecutive dotted vertical lines are the cutting planes going through the splitting points and dividing the attribute domain into three dense units. The dashed horizontal line is used to filter dense units of possible low density –

histogram count of the highest bin in the dense unit has to be greater than a density threshold. The density threshold is equal to the hypothetical average the histogram count if the attribute values were uniformly distributed, modified by parameter *sensitivity*. It is crucial parameter for IBUSCA that influences on the clustering results. This approach allows IBUSCA to obtain adaptive, more accurate grid (fig. 1. b)).

For each one-dimensional dense unit a threshold of minimal instances count is set on the basis of the algorithm parameter *sensitivity* – dense units created by merging lower dimensional units have to include the number of points equal to the greatest threshold from units which build the new cell.

One-dimensional dense units created during histogram analysis are then merged in greater dimensions. Dense units in n dimensions are obtained by merging these dense units of $n-1$ dimensions which share any of the $n-2$ dimensions. This process is running until no new dense units of greater dimensionality can be created. The explosion of dense units number can be reduced by keeping only units with the greatest or one of the greatest dimensionality. Finally, hyper-rectangular clusters are formed by dense units from the resulting set which can also be described as decision rules making the results easily understandable to a human. But the approach applied in subspace clustering methods results in clusters defined in various subspaces, with different dimensions, where some of them can be common in lower dimensional subspaces. This causes many difficulties in evaluation of results.

3 Cluster Quality Estimate in Subspace Clustering Methods

In this section we present our method of subspace clustering quality evaluation. The method has come into being as an experiment result of *Davies-Bouldin Index* [2] adaptation to subspace clustering. Each instance in data set is characterized by a large number of attributes, which can be divided into two categories: numerical and nominal. The instance is represented by a point in feature space. Different character of the attributes causes the problem with a distance calculating. A distance between two data points for numerical attribute is proportional to the arithmetical difference of these values divided by a full range of this attribute. The range means the difference between *maximal* and *minimal* values of this attribute. For nominal attributes a distance takes two values only: 0 for the same values or 1 for different values. We have assumed the method should estimate clustering quality without any knowledge of real dependences between data points as well as without verification whether nominal values could be graded. A definition of a good subspace clustering result has been an essential problem in order to evaluate it. It is very important to determine which clusters are better or worse than others. We have taken three assumptions into consideration:

1. We expect compact and well-separated clusters in common subspace.
2. High-dimensional clusters should be judged better than low-dimensional ones.
3. Only attributes creating subspaces of clusters should be taken into consideration during evaluation of clustering. Unimportant attributes should be omitted.

A dispersal of a cluster C_i in the subspace created by attributes from a collection D is calculated as an arithmetic average of Euclidean distances between each cluster data point and cluster centroid. During this calculation we take into account only

attributes from the collection D . A centroid is represented by the vector of mean values of each numeric attribute which belongs to C_i . For nominal attributes a centroid contains the most frequent value in the cluster C_i . The distance between two clusters in common subspace is represented by the Euclidean distance between their centroids in this subspace.

Evaluation of clustering result starts with a single cluster evaluation. It begins with finding common attributes of this cluster and each from the others in turn. Next, for each pair of clusters with disjoint sets of attributes the dispersal of these clusters is calculated, as well as their mutual distance in a common subspace. The sum of dispersals in proportion to the other clusters distance is calculated for each cluster. Then, the maximal value is chosen which estimates actual cluster quality in regard to the common subspaces of attributes. In the next step the dispersal in its full subspace is calculated. After a comparison the highest from these two values is chosen. It expresses the quality of a given cluster.

To accomplish 2nd and 3rd assumptions, in the metrics expressing the quality of clustering we have introduced a factor referring to the cluster dimensionality. So, finally the estimate of all subspace clustering quality is calculated as arithmetical average of clusters estimation multiplied by the cluster dimensionality factor for all clusters in subspace clustering. This metrics is inversely proportional to the clustering quality.

Going to the more formal specification of the metrics a distance between two data points x and y is calculated according to eq. 1.

$$d(x, y, D) = \sqrt{\sum_{i \in D} R(x_i, y_i)^2} \tag{1}$$

where: D - is a set of attributes that create the cluster subspace, i - is an attribute index, $R(x_i, y_i)$ - is the difference of i -th attribute values for data x and y , which is expressed by eq. 2.

$$R(x_i, y_i) = \begin{cases} \frac{x_i - y_i}{\max_i - \min_i} & \text{for numerical attributes} \\ 1 & \text{for nominal attributes when } x_i \neq y_i \\ 0 & \text{for nominal attributes when } x_i = y_i \end{cases} \tag{2}$$

where: \max_i is the maximal value of i -th attribute in data set, \min_i is the minimal value of i -th attribute in data set. As it can be noticed it is defined specifically for each type of considered types of attributes.

The dispersal Δ of cluster C_i in subspace created by attributes from the collection D is calculated as follows:

$$\Delta(C_i, D) = \frac{1}{|C_i|} \sum_{x_p \in C_i \text{ and } p \in D} d(s_{C_i, p}, x_p, D) \tag{3}$$

where: $s_{C_i, p}$ - is the average of p -th attribute values in cluster C_i , x_p is the value of p -th attribute of instance x which is a member of cluster C_i and $|C_i|$ is a number of data points (instances) in a cluster C_i .

The distance between clusters C_i and C_j in subspace of attributes for both clusters from common set D of attributes, is equal to the distance between their centroids:

$$\delta(C_i, C_j, D) = d(s_{C_i}, s_{C_j}, D) \tag{4}$$

The quality estimate QEC of the cluster C_i is the maximal value calculated as a sum of dispersal of the cluster C_i and other cluster C_j divided by the distance between their centroids.

$$QEC(C_i) = \max_j \left\{ \frac{\Delta(C_i, D) + \Delta(C_j, D)}{\delta(C_i, C_j, D)} \right\} \tag{5}$$

In generality a cluster C_i can be found as one from many others in common subspace. Then its quality can be expressed by QEC (eq. 5) but it can happen that the same cluster C_i can create a single cluster in a higher subspace. In this case as a quality estimate for this cluster the higher value from both values is assumed (eq. 6).

$$QEC(C_i) = \max\{QEC(C_i), \Delta(C_i, D)\} \tag{6}$$

As we have explained we would like to promote the clusters which are created in higher subspaces. To realize this assumption we have introduced a factor $k_{dim}(C_i)$ referring to the cluster C_i dimensionality:

$$k_{dim}(C_i) = 1 - \frac{nDim(C_i)}{nDim_{max} + 1} \tag{7}$$

where: $nDim(C_i)$ is the dimension of i -th cluster, $nDim_{max}$ is the maximal dimension between all clusters.

The proposed metrics - $SCQE$ of subspace clustering quality estimation can be finally expressed as an average of quality estimate of clusters QEC weighted by dimensionality factor:

$$SCQE = \frac{1}{L} \sum_{i=1}^L (k_{dim}(C_i) \cdot QEC(C_i)) \tag{8}$$

where L is the number of clusters in this clustering. The less is this value the better is the clustering.

4 Experimental Studies

We have tested our metrics evaluating the quality of subspace clustering results for artificially created data sets and data sets from UCI Machine Learning Repository *Labor* and *Iris* [6]. For clustering we have used IBUSCA algorithm with different values of *sensitivity* parameter. The aim of experiments with artificially created data was testing the conformity of the results with the assumption 1. and 2. The results have shown the metrics promotes clusters in higher dimensional subspaces. They have also confirmed that as distance between clusters in a common subspace is decreased the value of metrics is decreased as well. The experiments with UCI data sets will be presented wider. For these data sets the class attribute has been used only to estimate the quality of received clusters after clustering. During grouping process these attributes were ignored.

Labor dataset contains 57 elements described by 16 attributes: 8 numeric and 8 nominal. These elements are divided into 2 classes: *bad* (20 elements) and *good* (37 elements).

Table 1. Clusters for *Labor* dataset and the value of metrics *SCQE*

cluster	subspace dimensions	attributes: range	class	
			<i>bad</i>	<i>good</i>
<i>sensitivity = 1.4; SCQE = 0.05187</i>				
#1	2	working-hours: [37,83667; 40,00400) shift-differential: [0,00000; 4,16708)	8	8
#2	1	duration: [1,66677; 2,33353)	10	17
#3	1	wage-increase-first-year: [2,00000; 4,50035)	20	23
#4	1	statutory-holidays: [9,00000; 11,40060)	18	19
<i>sensitivity = 1.2; SCQE = 0.06007</i>				
#1	2	working-hours: [37,83667; 40,00400) shift-differential: [0,00000; 4,16708)	8	8
#2	1	duration: [1,66677; 2,33353)	10	17
#3	1	wage-increase-first-year: [2,00000; 4,50035)	20	23
#4	1	statutory-holidays: [9,00000; 11,40060)	18	19
#5	1	wage-increase-second-year: [2,00000; 4,00028)	14	14

Clustering result (Table 1) for *sensitivity* = 1.2 contains all clusters from clustering obtained with *sensitivity* = 1.4 and one cluster more (#5). Both of this clustering results consist of several clusters which are single in their subspaces (they do not have any common attributes). So the only criterion of clustering quality are clusters dispersals: Δ (#1)=0.07331, Δ (#2)=0, Δ (#3)=0.17233, Δ (#4)=0.10226, Δ (#5)=0.13928. The dispersal of cluster #2 is equal to 0 because the value of *duration* attribute for all elements of this cluster is equal to 2.0. Because value of our

Table 2. Clusters for *Iris* dataset and the value of metrics *SCQE*

cluster	subspace dimensions	attributes: range	class		
			<i>Setosa</i>	<i>Versicolor</i>	<i>Virginica</i>
<i>sensitivity = 1.6; SCQE = 0.02493</i>					
#1	2	petallength: [1,00000; 2,18014) petalwidth: [0,10000; 0,58005)	49	0	0
#2	1	sepalwidth: [2,60011; 3,20022)	16	32	35
<i>sensitivity = 1.5; SCQE = 0.05077</i>					
#1	2	sepalwidth: [2,60011; 3,20022) petallength: [2,77021; 6,90069)	0	32	35
#2	2	petallength: [1,00000; 2,77021) petalwidth: [0,10000; 0,58005)	49	0	0
#3	1	sepallength: [5,32880; 5,84320)	10	19	5
<i>sensitivity = 1.1; SCQE = 0.09826</i>					
#1	3	petallength: [2,77021; 6,90069) sepalwidth: [2,60011; 3,50028) petalwidth: [0,82007; 2,50025)	0	34	40
#2	2	petallength: [1,00000; 2,77021) petalwidth: [0,10000; 0,82007)	50	0	0
#3	1	sepallength: [4,81440; 6,87199)	34	48	35

metrics is inversely proportional to clustering quality, the clustering results for sensitivity = 1.4 are better than for sensitivity = 1.2. It is worth to notice that the additional cluster (#5) groups the same number of elements of each class, which lie near each other so its relative value of dispersal to the distance between cluster centroids is high.

Iris data set contains 150 elements divided into 3 classes, 50 elements each. The elements are described by 4 numerical attributes. Table 2 presents the results. Clustering result for sensitivity = 1.6 consists of two clusters. The first one contains elements only from the class *Setosa*, which is linearly separable from other classes (Fig. 2). The elements of *Versicolor* and *Virginica* classes are the most part of second cluster.

This cluster is built in one-dimensional subspace created by *sepalwidth* attribute. Clustering result for sensitivity = 1.5 encloses three groups of elements. The second one is the same as cluster #1 for sensitivity = 1.6. The first cluster corresponds to the cluster #2 for sensitivity = 1.6 but here there is an additional subspace dimension (*petallength*) which limits data points in this cluster to the elements of two classes. The third cluster contains elements of all classes. This cluster is the reason for worse quality of this clustering than quality of clustering for sensitivity = 1.6. The set of clusters for sensitivity = 1.1 is the worst. The main reason for this is its third cluster which contains comparable number of elements of each class.

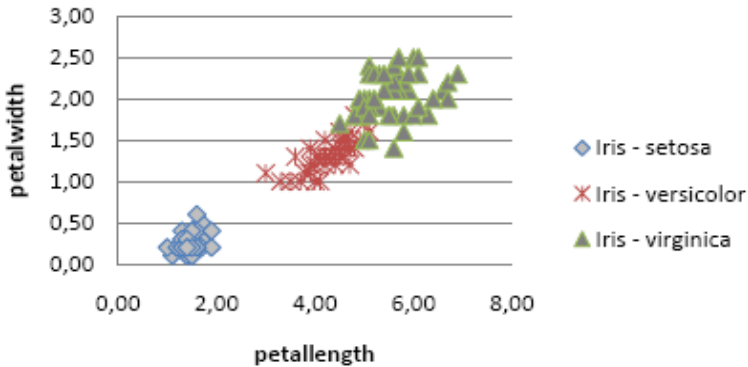


Fig. 2. Visualisation of *Iris* dataset

We have also examined the effect of attributes which do not contain any clusters on the value of our quality measure. For this purpose we have extended described data sets: *Labor* and *Iris* by adding 100 additional attributes. Each of these attributes have uniform distribution of values with 5% noise. The results of experiments have shown that the attributes which do not create any cluster do not make an effect on *SCQE* value.

5 Conclusions

The results of performed experiments have shown that the *SCQE* metrics (subspace clustering quality estimate) reflects the real quality of received cluster groups. Nevertheless to fully evaluate it more experiments are needed with artificially created data

sets including specifically generated clusters what can lead to valuable evaluations and further improvements. Taking into account nominal attributes in the SCQE we have applied a straightforward solution. In the future we plan to consider nominal values that can be semantically or structurally similar.

References

1. Bouveyron, C., Girard, S., Schmid, C.: High-dimensional data clustering. *Computational Statistical and Data Analysis* 52(1), 502–519 (2007)
2. Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE Trans. Pattern Analysis and Machine Intelligence* 1, 224–227 (1979)
3. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *KDD*, pp. 226–231 (1996)
4. Glomba, M., Markowska-Kaczmar, U.: IBUSCA: A Grid-based Bottom-up Subspace Clustering Algorithm. In: *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA 2006)*. IEEE Computer Society, Los Alamitos (2006)
5. Han, J., Kamber, M.: *Cluster Analysis*. In: *Data Mining: Concept and Techniques*, pp. 335–393. Morgan Kaufman Publishers/Academic Press (2001)
6. Newman, S., Hettich, D., Blake, C., Merz, C.: Uci repository of machine learning databases, <http://www.ics.uci.edu/mllearn/MLRepository.html>
7. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.* 6(1), 90–105 (2004)
8. Patrikainen, A., Meila, M.: Comparing subspace clusterings. *IEEE Transactions on Knowledge and Data Engineering* 18(7), 902–916 (2006)
9. Wand, M.P.: Data-based choice of histogram bin width. *The American Statistician* 51(1), 59–64 (1997)

Clustering by Chaotic Neural Networks with Mean Field Calculated Via Delaunay Triangulation

Elena N. Benderskaya and Sofya V. Zhukova

St. Petersburg State Polytechnical University, Faculty of Computer Science,
Russia, 194021, Saint-Petersburg, Politechnicheskaya 21
bender@sp.ru, sophya.zhukova@gmail.com

Abstract. High quality clustering is impossible without a priori information about clustering criteria. The paper describes the development of new clustering technique based on chaotic neural networks that overcomes the indeterminacy about number and topology of clusters. Proposed method of weights computation via Delaunay triangulation allows to cut down computing complexity of chaotic neural network clustering.

Keywords: clustering, cluster analysis, chaotic neural network (CNN), chaotic map lattices, Delaunay triangulation.

1 Introduction

Clustering problem is considered to be one of the most complicated tasks among those defined in pattern recognition. The set of elements division into non-overlapping groups (clusters) is provided via criterion of similarity that predetermines the result. To choose the proper measure of similarity between objects is very hard without a priori information about clusters topology and amount of clusters. To overcome the uncertainty expert evaluation is used at various clustering stages. Thus all clustering methods remain to be automated, but not automatic. In one's turn this fact restrains greatly pace of development of automatic decision making systems.

This paper describes an improvement of a perspective clustering method that comprises the main ideas of synergetics and last results of chaos theory and pattern recognition. Proposed model is a hybrid system. It joins both the advantages of neural networks distributed parallel processing and wide abilities of nonlinear systems with chaotic dynamics. Parallel processing here allows overcoming the curse of averaging out when centers of clusters are not formed. Bringing in neural networks functioning chaotic component due to specific transfer functions helps to reveal unique implicit laws in any dataset even without a priori information about its topology.

1.1 Metrics and Their Influence on Clustering Results

In general clusterization quality is satisfactory if following prerequisites are fulfilled: within a cluster objects are strongly alike; objects from different clusters are dissimilar to each other; with other things being equal distributions of objects within groups

should be uniform. First two requirements express the standard concept of compactness of classes; last requirement consists in that the criterion did not impose association objects into groups.

Thus there are two types of metrics [1-2]: type 1 - similarity measure between objects within a cluster (Euclidean, Cityblock, Mahalanobis, Minkowski, Cosine, Chebyshev, Supremum norm); type 2 - similarity (or dissimilarity) measure between the clusters themselves (Single linkage, Complete linkage, Median clustering, Centroid clustering, Ward's method, Statistical clustering). Numerous clustering techniques are named according to the concrete metric or group of metrics.

Despite of great amount of methods clustering quality may be even unsatisfactory if the hypothesis about input image topology is wrong. In case of clustering problem we do not know the target clusterization. Under these conditions a whole set of methods are used to generate variants of solution. In the end all of the variants are analyzed and the winning clusterization (the answer) is chosen as the most frequent one [3]. This induces greatly the increase of computational resources expenses. So there is a great need of a method that is capable of adaptation to the peculiarities of input dataset.

1.2 Synergetics and Hybrid Systems for Pattern Recognition

Due to the direction of science integration new solutions of 'old' complex problems are being found on the intersections of various scientific fields. We improve a method based on using inhomogeneous coupled map lattices (CML) with chaotic dynamics. L. Angelini [5-8] and his colleagues for the first time used such a model for data clustering and named it as chaotic neural network (CNN). The method comprises knowledge from: cybernetics, self organization paradigm [4], chaos theory [5], synchronization theory [5-7], and pattern recognition [1, 8]. We insist that it is necessary to add to this method a topological component.

Forming clusters from previously separate elements turns out to be universal concept in animate nature and in abiocoen. Self-organization occurs in various phenomena such as structures creation, cooperative behavior, etc [4, 9]. Clusters built up from atoms, molecules, neurons are examined in many scientific fields.

Computer science development predetermined great abilities of computer modeling. It became possible to study complex nonlinear dynamics. Great evidence for rich behavior of artificial chaotic systems was accumulated and thus chaos theory came into being [5, 11, 12]. Dynamics exponential unpredictability of chaotic systems, their extreme instability generates variety of system's possible states that can help us to describe all the multiformity of our planet. It is assumed to be very advantageous to obtain clustering problem solution using effects produced by chaotic systems interaction.

We deal with structural hybridism. In this paper the hybrid neural network itself produces chaos not abstract but unique to each input dataset. So we do not have to use the concept of averaging out. The structure of neural network is like Hopfield's [10] only without self-links, but transfer functions of neurons are logistic maps that guarantee chaotic fluctuations of outputs. Such recurrent neural network obtains an ability not only to solve classification problem but more complex one – clustering problem.

2 CNN Model

Primary results on modeling high dimensional chaotic map lattices were published by K. Kaneko [11-12]. These works showed up the fact that globally coupled chaotic map lattices exhibit formation of ensembles synchronously oscillating elements. These ensembles were called clusters serving as system's attractors. If there appear to be several clusters then the system is characterized by multistability, when several attractors coexist in the phase space at the same parameters values.

Synchronization as a universal concept is comprehensively described in [5-6]. To summarize the conditions that cause inner synchronous motions among groups of nonlinear systems it is necessary to mark out the following:

- (a) large amount of globally coupled nonlinear elements;
- (b) weak coupling strength to exclude the possibility of several elements to suppress individual dynamics of all others;
- (c) instability dynamics of each nonlinear element;
- (d) feedbacks to provide own element's dynamics tuning to the neighbors' fluctuations.

To apply oscillatory clustering phenomenon L. Angelini and his colleagues proposed [13-15] to hand on information about input dataset to logistic map network by means of inhomogeneous weights assignment

$$W = \{w_{ij}\} = \exp\left(-\frac{d_{ij}^2}{2a}\right), d_{ij} = |x_i - x_j|, i, j = \overline{1, N}, \quad (1)$$

where N – number of elements, w_{ij} - strength of link between elements i and j , d_{ij} - Euclidean distance between neurons i and j , a – local scale, depending on k -nearest neighbors. The value of a is fixed as the average distance of k -nearest neighbor pairs of points in the whole system. So we can see that here the authors used modified Euclidean metric

Each neuron is responsible for one object in the dataset, but the image itself is not given to inputs, because CNN does not have classical inputs – it is recurrent neural network with one layer of N neurons. Instead, the image (input dataset) predetermines the strength of neurons interactions (as at Hopfield's network [10]).

As long as $d_{ii} = 0$ ($i = \overline{1, N}$), then there is no loops in Angelini's model. Evolution of each neuron is governed by

$$y_i(t+1) = \frac{1}{C_i} \sum_{i \neq j}^N w_{ij} f(y_i(t)), t = 1..T, \quad (2)$$

$$f(y(t)) = 1 - 2y^2(t) \quad (3)$$

where $C_i = \sum_{i \neq j} w_{ij}$, $i, j = \overline{1, N}$, T – time interval, N – number of elements. Neurons state is dependent on the state of all other elements. In [11, 12] the system's functioning is

divided into two parts: transition regime and stationary regime. The stationary regime of difference equation system (4) corresponds to a macroscopic attractor which is independent of the initial conditions. During transition period T_p the system converges to the macroscopic attractor. Stationary regime is characterized by gathering statistics about dynamics of each variable y_i ($i = \overline{1, N}$). Then observation matrix $\{NO_{ij}\}$ is formed:

$$NO_{ij} = \begin{cases} 1, & y_i > 0, \\ 0, & y_i \leq 0 \end{cases}, \quad i = \overline{1, N}, j = \overline{1, T_n} \quad (4)$$

where y_i – output of variable i , T_n – time interval to gather dynamics statistics in stationary regime. Each row in the observation matrix corresponds to the sequence of bits that can be interpreted as messages and thus Shannon entropy for each variable is calculated. Then information matrix $I = \{I_{ij}\}$, ($i, j = \overline{1, N}$) is fixed. Each element in this matrix is the mutual information that i th element carries about j th element. Information is determined on the base of joint entropy. If maps i and j evolve independently, then $I_{ij} = 0$; if the two maps are exactly synchronized, then the mutual information achieves its maximum value [5], in the present case $\ln 2 \approx 0.69$. Each time values $\{I_{ij}\}$ are compared with concrete threshold θ_i and list of clusters is formed. Thus the hierarchy of solutions is obtained.

Neurons are joined in clusters several times, because a priori it is unknown the veritable threshold θ that corresponds to real clustering. The value of θ_i controls the resolution at which the data is clustered. Thresholds are chosen with some step in the range of minimum and maximum values in the information matrix. Neural network in which weight coefficients are calculated in compliance with (1) and evolution is given by (2) was called chaotic neural network (CNN) to stress the chaotic functioning of the system, guaranteed by transfer function (3).

3 CNN Modeling

For a start 2D test clustering problem illustrated in Fig.1 were solved in terms of Angelini's model. This gives us opportunity to reveal the influence of every parameter on the result. Since we know the answer for a test clustering problem then it is correct to order inputs in the dataset by their belongings to clusters. It is important to stress that this operation will not change CNN functioning, because initial conditions are set in a random way in the range $[-1, 1]$ as before. This will help us to watch CNN parameters and dynamics. CNN parameters are: T_n , T_p , k and θ . Preliminary experiments showed that $T_n = 1000$ is quite enough for system (with proper strengths of elements interactions) to achieve macroscopic attractor. And T_p should be taken more than 100 to gather representative statistics. As we have no a priori information about input dataset this means that we don't know the target range of θ . So we generated 15 output clusterizations changing θ with step 0.05 from 0 up to maximum value in information matrix, calculated via (4).

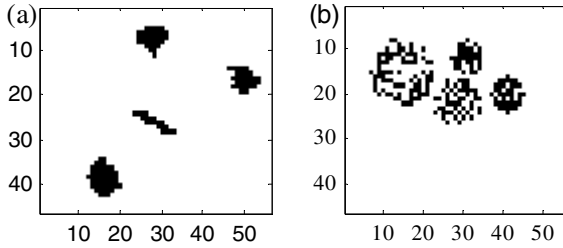


Fig. 1. Clustering problems: (a) - simple clustering problem (134 objects, each described by two coordinates ($p=2$), four clusters are compact and stand apart from each others, clusters comprise correspondingly 49, 36, 30, 19 objects-points); (b) – more complex clustering problem (210 objects arranged with lower density in four clusters close to each other in future space, population is correspondingly 85, 50, 40, 35 points)

Though it is said in [14] that number of k -nearest neighbors have no influence on clustering results in wide range of values, this parameter is of great importance. In Fig. 2a, Fig. 2c, Fig. 2e are visualized weight coefficients matrixes calculated for different numbers of k -nearest neighbors and in Fig. 2b, Fig. 2d, Fig. 2f is shown corresponding CNN dynamics. Number of nearest neighbors influence on the mean field that have an effect on all of the neurons. If the field is too inhomogeneous then it prevents synchronization of elements even within the same cluster as we can see in Fig. 2a, Fig. 2b. On the opposite, if the field is too homogeneous (number of neighbors is comparable with number of points in the image) complete synchronization predetermines only one cluster and impossibility of the division into groups as we have in Fig. 2e, Fig. 2f.

As it is shown on Fig. 2c, Fig. 2d suitable synchronous dynamics occurs if k -nearest neighbors' value approximately fits the smallest cluster population. In this case weights field is homogeneous within each of the clusters and is zero beyond their bounds. Besides, neurons dynamics dependence on initial conditions exist no more due to stationary regime (in terms of Kaneko to the ordered phase). The macroscopic attractor contains trajectories of all the variables in feature space. All trajectories remains to be chaotic due to the transfer function (4). Thus, a trajectory in the phase space of CNN undoubtedly depends on initial conditions. Yet, mutual trajectories disposition in N -dimensional phase space stands to be invariable to start point. That means that mutual synchronization is insensitive to initial conditions and as a result is insensitive the obtained solution of clustering problem. And this is exactly what we need.

At present in general case solving high dimension system of difference equation does not always succeed. The solution is frequently obtained by means of computer modeling. Though this process can be automated at great extent nevertheless it requires large computational resources and expert assistance at the final stage of CNN parameters definition. Number of clusters and their structure constancy independence from initial conditions serve as a criterion for unique and proper clustering result.

A priori uncertainty about amount and topology of clusters now is replaced by uncertainty about CNN parameters.

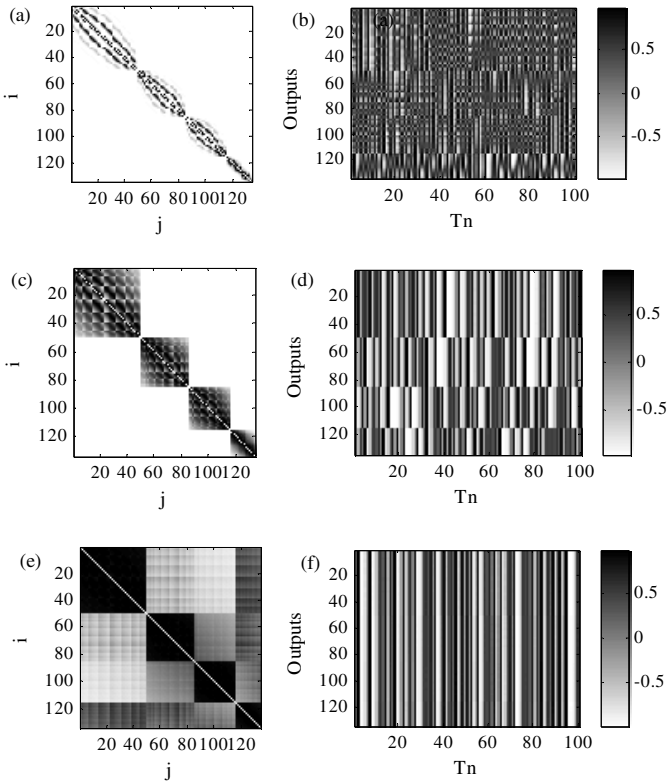


Fig. 2. Mean field and correspondent dynamics: (a, b) – number of nearest neighbors $k=2$; (c, d) – number of nearest neighbors $k=30$, (e, f) – number of nearest neighbors $k=100$

4 Mean Field Via Delaunay Triangulation

Mathematical analysis of Angelini model proposed in [14] helps to reveal the one of the impediments that prevents CNN application in solving complex clustering problems without expert's participation.

In CNN model weights matrix W is calculated under the condition that previously was determined local scale a on the base of a priori unknown value of k -nearest neighbors. Generalization of classical clustering methods brings to a conclusion that only geometrical criterion of least distances [1-3] values does not always provide proper clustering results especially if the assumption about the metric happens to be wrong. To fix the parameters that ensure stable clustering results CNN has been run over and over again from different initial conditions and k -nearest neighbors' values. To reduce number of frequentative experiments we propose to calculate weights coefficients using Delaunay triangulation [16].

In this paper we propose a new clustering method on the base of CNN as a development of Angelini's model. The method comprises new approach to analytical determination of local scale a , and a different CNN's dynamics interpretation.

The values range of local scale a ensuring solution insensibility to the changes in initial conditions becomes shorter as far as clusters topology complexity grows. The less is the range the more identical experiments are needed to find this range. Local scale depends on a priori unknown number of k -nearest neighbors.

To find the way of automatic k -nearest value calculation were tried different variants. We took into account average pairwise dispersion between objects, or only those neighbors of i th point that lay at some distance, and other approaches based on statistical analysis of input image. But all of them did not bring to desired being not unified. Lack of universality can be explained by consideration only nearest neighbors in the sense of distance but not the topology of the neighbors.

To define nearest neighbors taking into account image topology we propose to use Delaunay triangulation. Triangulation is a set of lines connecting each point to its natural neighbors as it is shown in Fig. 3. These lines form a loop-free graph with triangles as component parts. There are many ways to find triangulation. If for each triangle is true the condition that the unique circle circumscribed about the triangle contains no data points then we deal with Delaunay triangulation.

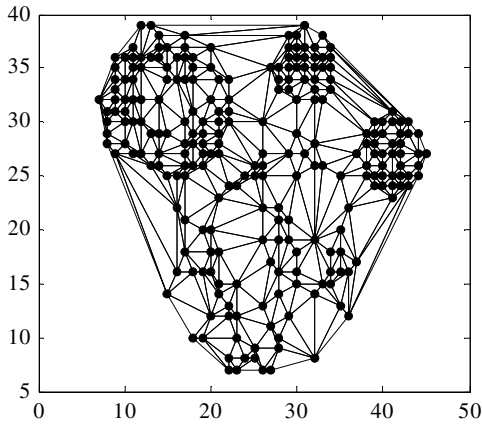


Fig. 3. Delaunay triangulation for image in Fig 1b

Delaunay triangulation [16] gives us all the nearest neighbors of each point from all directions. The value of a is now fixed as the average distance of Delaunay-nearest neighbor pairs of points in the whole system. Thus we form the proper mean field that contributes greatly to convergence of CNN dynamics to macroscopic attractor.

Some results on experimental and analytical evaluation of local scale a you can see in Fig. 4.

In Fig. 4a for both images local scale depends directly proportional on k -nearest neighbors (corner of an inclination depends on clusters topology). In Fig. 4b we can see that the range of local scale a ensuring proper solution (in this paper number of coinciding variants R must be more than 7 from 15 various thresholds θ) when clustering a simple image in Fig. 1a ([2,5; 9.8]) becomes shorter ([4.5; 6.1]) when clustering more complex problem in Fig. 1b . Local scale a calculated for both of images

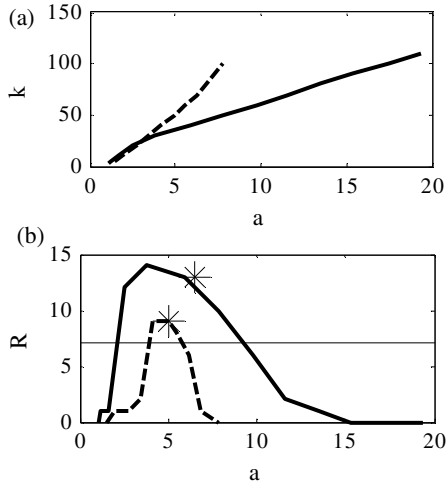


Fig. 4. Experiment results: (a) – experimentally modeled local scale a dependence on k nearest neighbors (solid line – when clustering image on Fig.1a, dashed line - when clustering image on Fig. 1b), (b) – number of right clusterizations (from 15 available) depending on local scale a . With asterisk are marked values of a obtained via proposed method based on Delaunay triangulation. Both of them belong to the experimentally fixed ranges of a values, that correspond to macroscopic attractor ($a = 6.5 \in [2.5; 9.8]$ for Fig. 1a, $a = 9 \in [4.5; 6.1]$ for Fig. 1b).

(and many others from our collection) via Delaunay triangulation always belonged to experimentally got range. On Fig. 4b analytically computed values of local scale are marked with asterisks.

Thus proposed way of local scale determination allows to remove a priori ambiguity about value of k -nearest neighbors in Angelini model. Successful experiments on the wide set of images confirmed the effectiveness of suggested computational procedure.

5 Conclusion

It was shown that it is possible to cut down computational complexity of the CNN clustering method via proposed algorithm of mean field calculation. So the method consists of three phases: formalization of input dataset (and dynamics parameters estimation), evolution of system itself and outputs processing. We also tried to exclude second phase and try to cluster images formalized via (1) with our method of local scale calculation by the help of set of classical clustering methods, but the results were negative. The proposed technique is effective with both taking into account topology of clusters and CNN dynamics.

The main advantage of proposed technique is the ability to produce adequate clusterization even under the condition of absence of any a priori information about the topology and amount of groups. Other clustering techniques like single linkage, complete linkage, median clustering, centroid clustering, ward's method, statistical clustering, k-means, self-organizing maps [10] on the same set of complex clustering

problems (the clusters are not compact in the feature space) can not produce mistakes-free answers without an expert that preliminary classify the problems to the methods.

References

1. Kumar, B.V.K.V., Mahalanobis, A., Juday, R.D.: Correlation Pattern Recognition, p. 402. Cambridge University Press, Cambridge (2006)
2. Valente de Oliveira, J., Pedrycz, W.: Advances in Fuzzy Clustering and its Applications, p. 454. Wiley, Chichester (2007)
3. Dimitriadou, E., Weingessel, A., Hornik, K.: Voting-Merging: An Ensemble Method for Clustering. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) ICANN 2001. LNCS, vol. 2130, pp. 217–224. Springer, Heidelberg (2001)
4. Haken, H.: Synergetics. Introduction and Advanced Topics. In: Physics and Astronomy Online Library, p. 758. Springer, Heidelberg (2004)
5. Mosekilde, E., Maistrenko, Y., Postnov, D.: Chaotic synchronization. World Scientific Series on Nonlinear Science, Series A 42, 440 (2002)
6. Pikovsky, A., Rosenblum, M., Kurths, J.: Synchronization: A Universal Concept in Nonlinear Sciences. Cambridge University Press, Cambridge (2003)
7. Osipov, G.V., Kurths, J., Zhou, C.: Synchronization in Oscillatory Networks. Springer Series in Synergetics, p. 370. Springer, Heidelberg (2007)
8. Han, J., Kamber, M.: Data Mining. Concepts and Techniques. The Morgan Kaufmann Series in Data Management Systems, p. 800. Morgan Kaufmann, San Francisco (2005)
9. Schweitzer, F.: Self-Organization of Complex Structures: From Individual to Collective Dynamics, p. 620. CRC Press, Boca Raton (1997)
10. Haykin, S.: Neural Networks. A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River (1998)
11. Kaneko, K.: Phenomenology of spatio-temporal chaos. Directions in chaos, pp. 272–353. World Scientific Publishing Co., Singapore (1987)
12. Kaneko, K.: Chaotic but regular posinega switch among coded attractors by clustersize variations. Phys. Rev. Lett. N63(14), 219–223 (1989)
13. Angelini, L., Carlo, F., Marangi, C., Pellicoro, M., Nardullia, M., Stramaglia, S.: Clustering data by inhomogeneous chaotic map lattices. Phys. Rev. Lett. N85, 78–102 (2000)
14. Angelini, L., Carlo, F., Marangi, C., Pellicoro, M., Nardullia, M., Stramaglia, S.: Clustering by inhomogeneous chaotic maps in landmine detection. Phys. Rev. Lett. N86, 89–132 (2001)
15. Angelini, L.: Antiferromagnetic effects in chaotic map lattices with a conservation law. Physics Letters A 307(1), 41–49 (2003)
16. Preparata, F.R., Shamos, M.I.: Computational Geometry. An Introduction, Monographs in Computer Science, p. 398. Springer, Heidelberg (1993)

Image Fusion Algorithm Using RBF Neural Networks

Hong Zhang^{1,2}, Xiao-nan Sun², Lei Zhao², and Lei Liu^{1*}

¹ College of Computer Science and Technology, Jilin University,
2699 Qianjin Street, 130012, Changchun, China
hong168z@126.com

² State Key Laboratory on Integrated Optoelectronics, College of Electronic Science
and Engineering, Jilin University, 2699 Qianjin Street, 130012, Changchun, China
zhong@jlu.edu.cn

Abstract. This paper presents the image fusion algorithm using Radial Basis Function (RBF) neural networks. The clustering of every original image pixel is obtained by RBF neural networks combined with nearest neighbor clustering method. For each cluster center, the membership of every fused image pixel is adopted as the weighting coefficient of the weighted strategy, which is used to obtain the fusion image. The membership is obtained by maximum rule. The original data set is chosen as the candidate set of nearest neighbor clustering algorithm, and the center set of hidden units are dynamically established. In this experiment, the fusion results of various widths of hidden unit are compared with the results obtained by self-organizing feature map (SOFM) neural networks method. The influence of the various widths is discussed in this paper. The experiment results show that the proposed method can achieve better performance of the fused image.

Keywords: RBF Neural networks; Image Fusion; Membership; Nearest Neighbor Clustering Algorithm.

1 Introduction

Image fusion is to integrate complementary information from the same view point under different focal sensors. Because of the different properties of image sensors, these images might provide totally different information. Image fusion process is required to give the new image which is more suitable for the purpose of human visual perception and computer-processing tasks. It has been widely applied to many areas such as computer vision, automatic target recognition, remote sensing, robotics and medical image processing etc.

Many neural network models [1-3] have been proposed for image fusion such as BP and SOFM neural networks. BP algorithm has been mostly used. However, the convergence of BP networks is slow and the global minima of the error space may not be always achieved. As an unsupervised network, SOFM network clusters input

* Corresponding author.

sample through competitive learning. But the number of output neurons should be set before constructing neural networks model.

RBF neural network [4] can approximate objective function at any precise level if enough hidden units are provided. The advantages of RBF network training include no iteration, few training parameters, high training speed, simply process and memory functions. This paper explores the way that using RBF neural networks combined with nearest neighbor clustering method to cluster, and membership weighting is used to fuse. Experiments show this method can obtain the better effect of cluster fusion with proper width parameter.

The following sections of this paper are organized as follows. Section 2 presents the proposed algorithm. Experimental results compared with the SOFM algorithm are presented in Section 3. Finally Section 4 summarizes the conclusions.

2 RBF Neural Networks Fusion Algorithm

The flow chart of the proposed algorithm is shown in Fig.1. The two original images are merged into an image by weighted fusion strategy. Firstly, the nearest neighbor clustering algorithm is used to choose the centers of the RBF hidden units. Secondly, RBF neural network is used to cluster the image pixels. According to the maximum membership rule, the fused image membership, which is the weighting coefficients of the weighted strategy, is obtained. The membership is the percentage content of every pixel to a different cluster. Finally, the fused image is obtained by the weighting coefficients.

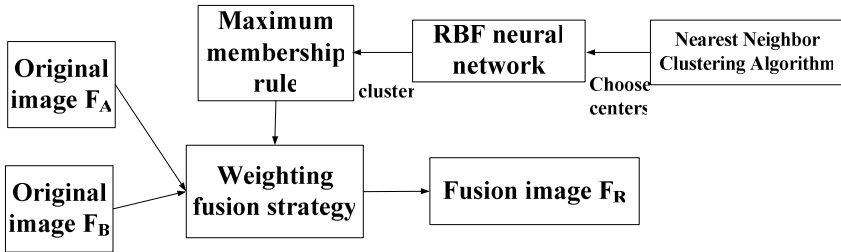


Fig. 1. The flow chart of the RBF neural networks algorithm

2.1 Image Clustering Using RBF Algorithm

RBF neural network is presented by J.moody and C.darken [5-7]. Generally, a RBF network is a feedforward architecture consisting of three layers: input layer, hidden layer and output layer. The hidden layer composed of the respective fields plays an important role. The respective fields are called radial basis functions. The radial basis function is typically chosen as Gaussian function, as

$$y(x) = \frac{\exp(-\|x - \omega_i\|^2)}{2r^2} \quad i = 1, 2, \dots, l \tag{1}$$

where x represents any pattern vector in the training set. ω_i and r are called the center of the kernel function of the i th hidden unit and its width respectively. l denotes the number of hidden units.

For clustering, RBF network construction includes two important decisions: the number of radial basis functions and the location of the cluster center[8-9]. In practice, the number of radial basis functions is related to the initialization methods. The standard way is to define the number of hidden units beforehand. Then the center is chosen randomly or determined by supervised or unsupervised way. It is still in use nowadays due to its simplicity in clustering. One drawback of this technique is that it depends on experience. Sometimes, inappropriate choice of centers will bring bad node and waste network source.

To get rid of the drawback, we combine RBF neural network with the nearest neighbor clustering algorithm [10-11]. This algorithm is self-adaptive, and the number of hidden units doesn't need to know beforehand. It establishes the centers dynamically according to the input samples. There is a simple linear relation between calculation amount and the number of data samples. Thus, little calculation amount and few control parameters are required.

Assuming the size of source image is $n_1 \times n_2$, and p_k denotes the gray scale of the k th data point, is the data sample. The RBF algorithm using nearest neighbor clustering algorithm consists of the following steps:

Step 1: Initialize the center vectors ω_i , and the learning parameter of RBF network. Choose an appropriate width r of the Gaussian function.

Step 2: Consider the first data sample p_1 . Set p_1 to be the first clustering center, that is $\omega_1 = p_1$. The RBF network has only one hidden node whose center is ω_1 .

Step 3: For the second data sample p_2 , calculate the distance between p_2 and ω_1 . If $\|p_2 - \omega_1\| \leq r$, p_2 and ω_1 belong to the same nearest neighbor cluster, so there is still only one cluster. If the distance is larger than r , a new cluster shall be set. Let $\omega_2 = p_2$ where ω_2 represents the center of the new hidden unit.

Step 4: When the k th sample p_k is considered, assuming the number of hidden units is M , calculate the distances between p_k and all the existing centers $\omega_1, \omega_2, \dots, \omega_M$. If the distances between p_k and the existing centers are all larger than r , set p_k to be a new center, that is $\omega_{M+1} = p_k$. Therefore a new hidden unit is added in the construction of RBF network. Otherwise, if $\|p_k - \omega_j\|$ is the minimum among all the distances and $\|p_k - \omega_j\| \leq r$, ω_j is the nearest neighbor cluster center of p_k and there is no new cluster center.

Step 5: Repeat the above processes until the last data sample $p_{n_1 \times n_2}$ has been traveled.

Finally, the number of the cluster centers comes to be K and $\omega_1, \omega_2, \dots, \omega_K$ are the values of center. If a pixel of source images belongs to a special cluster g , the

membership degree of μ is $\mu_k = 1 (k = g)$ or $\mu_k = 0 (k \neq g)$. Assuming p_k belongs to the h th cluster and the center value is ω_h , it can be expressed as follows:

$$\|p_k - \omega_h\| = \min_{1 \leq j \leq K} \{\|p_k - \omega_j\|\} \quad k = 1, 2, \dots, n_1 \times n_2 \tag{2}$$

2.2 Membership of Image Pixel

The improved membership degrees [12] are introduced to change the value range of μ from the specific value of 0 or 1 to the range between 0 and 1. That is to say, the pixels will no longer belong to a special cluster but belong to a different cluster according to the percent content. $\mu_j(p_k)$ denotes the membership function of p_k assigned to j th cluster center as follows:

$$\mu_j(p_k) = \begin{cases} 0 & , p_k \notin [\omega_j, \omega_{j+1}] \\ (\omega_{j+1} - p_k) / (\omega_{j+1} - \omega_j), p_k \in [\omega_j, \omega_{j+1}] \end{cases} \tag{3}$$

where $j = 1, 2, \dots, K$; $k = 1, 2, \dots, n_1 \times n_2$; ω_j and ω_{j+1} stand for the center of the j th and the $j+1$ th cluster respectively. And the membership function to $j+1$ th cluster can be calculated by:

$$\mu_{j+1}(p_k) = 1 - (\omega_{j+1} - p_k) / (\omega_{j+1} - \omega_j) = 1 - \mu_j(p_k) \tag{4}$$

Namely, when the grey level p_k is between $[\omega_j, \omega_{j+1}]$, formula (3) and formula (4) can be used to get the memberships of this pixel to ω_j and ω_{j+1} . Thus we get

$$\sum_{j=1}^K \mu_j(p_k) = 1.$$

2.3 Image Fusion

We assume there are N original images to be fused. $p_k(I)$ indicates the gray scale of pixel k of the I th image, and $\mu_j(p_k(I))$ denotes the membership degree of pixel k of the I th image to the j th cluster. Then the membership degree of the pixel k of fused image F to the j th cluster is expressed as:

$$\mu_j(p_k^F) = \max_{1 \leq I \leq N} \left| \mu_j(p_k(I)) \right| \tag{5}$$

The grey scale p_k^F of pixel k of the fused image F is obtained by the weighted mean method:

$$p_k^F = \frac{\sum_{j=1}^K \left(\mu_j(p_k^F) \times \omega_j \right)}{\sum_{j=1}^K \mu_j(p_k^F)} \tag{6}$$

The fused image can be constructed by outputs of formula (6).

3 Experimental Results

To illustrate the RBF algorithm, two sets of source and fused images are presented in this section. The RBF algorithm is performed on two data sets with different width parameter r , the remaining two data sets tackled by SOFM algorithm are properly chosen for comparison. Root Mean Square Error (RMSE) and peak signal-to-noise ration (PSNR) [13] are applied in all the cases to evaluate the result. RMSE indicates the error between the fused image and the standard image, so the smaller the value, the better the fusion effect. PSNR indicates better image quality and more information if it is larger.

As shown in fig. 2 and 3, fig.2(a), 2(b) and 3(a), 3(b) are the source images of different focus respectively. Their sizes are all 512*512 bits. Fig.2(d) and 3(d) are the results of RBF algorithm using 3 as width parameter r . Fig 2(c)and 3(c) are the results of SOFM algorithm. According to the fig 2 and 3, the results of RBF algorithm always outperform the results of SOFM algorithm.

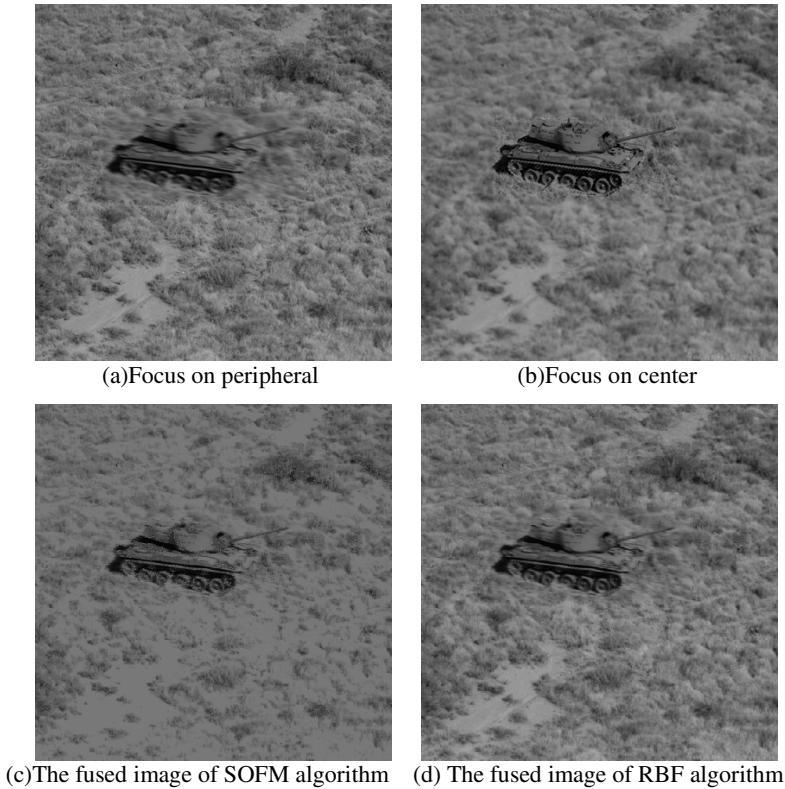


Fig. 2. The fused images of the tank images



(a) Plane head is covered



(b) Plane tail is covered.



(c)The fused image of SOFM algorithm



(d) The fused image of RBF algorithm

Fig. 3. The fused images of the plane images

Table 1 shows the evaluation value of two sets of images. In RBF algorithm, with the increment of r , the values of RMSE become larger and the values of PSNR become smaller. Compared with the results of SOFM algorithm, the results of RBF algorithm are satisfactory. However in plane image, the RMSE value of the SOFM algorithm is smaller than the result of the RBF algorithm using 25 as width parameter r .

The fusion results of RBF algorithm with four sets of width parameter r obtain the different evaluation value. Overall, smaller radius can decrease the value of RMSE as well as increase the value of PSNR, so the fusion effect is better. If the r is smaller, the number of clusters is more, and the more character details are preserved. At the same time, the noise points are clearly distinguished by clustering, and the fusion can eliminate their effects. If the r is too big-scale, the number of clusters is few, some image information can be lost, the image quality can be influenced. It can be concluded that the width parameter r is important to the quality of fusion image. In fact, the process of optimizing is equated with the process of image characteristic definition. It influences not only the number of clusters, but also quality of fused image.

Table 1. Comparison of RBF algorithm and SOFM algorithm

			RMSE	PSNR
Tank images	RBF algorithm	r=3	0.0134	27.6937
		r=6	0.0144	26.2401
		r=14	0.0188	20.6621
		r=25	0.0307	11.0259
		SOFM algorithm	0.0432	0.6621
Plane images	RBF algorithm	r=3	0.0418	7.0891
		r=6	0.0433	6.3608
		r=14	0.0466	4.8930
		r=25	0.0560	1.0307
		SOFM algorithm	0.0541	-0.0026

4 Conclusion

This paper proposes a novel image fusion algorithm using RBF neural networks combined with nearest neighbor clustering method. The membership is used to highlight fusion method. It needs no iteration and establishes center set based on input sample. The process of training RBF neural network becomes faster and more efficient. In addition, we analyze the proper choice of width parameter with many experiment and detailed description. The better clustering leads to the better effect of fusion, thus create an effective approach for image fusion.

References

1. Huang, W., Jing, Z.: Multi-focus image fusion using pulse coupled neural network. *Pattern Recognition Letters* 28, 1123–1132 (2007)
2. Li, S., Kwok, J.T., Wang, Y.: Multifocus image fusion using artificial neural networks. *Pattern Recognition Letters* 23, 985–997 (2002)
3. Wu, Y., Liu, C., Liao, G.: Multi-focus Image Fusion Based on SOFM Neural Networks and Evolution Strategies. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 1–10. Springer, Heidelberg (2005)
4. Zhen-zhen, S., Kun, F., Yi-rong, W.: The High-Resolution SAR Image Terrain Classification Algorithm Based on Mixed Double Hint Layers RBFN Model. *Acta electronica sinica* 31(12A), 2040–2044 (2003)
5. Mark, J., Orr, L.: Introduction to radial basis function networks. Center for Cognitive Science, University of Edinburgh, Buccleuch Place, Edinburgh EH8 9LW, Scotland, pp. 54–60 (1996)
6. Chen, S., Cowan, C.: Orthogonal least squares learning for radial basis function network. *IEEE Transactions on Neural Networks* 2(2), 302–309 (1991)
7. Huang, D.S.: Radial basis probabilistic neural networks: model and application. *International Journal of Pattern Recognition and Artificial Intelligence* 13(7), 1083–1101 (1999)
8. Zhuo-sheng, Z.: Radial Basis Function Networks Based on Adaptive Projective Learning Algorithm and Its Applications. *Acta Electronica Sinica* 28(9), 120–122 (2000)
9. Ming-xing, Z., De-long, Z.: Study on the Algorithms of Selecting the Radial Basis Function Center. *Journal of Anhui University Natural Science Edition* 24(1), 72–78 (2000)

10. Hu, Q., Yu, D., Xie, Z.: Neighborhood classifiers. *Expert Systems with Applications* 34, 866–876 (2008)
11. Chen, Y.-S., Hung, Y.-P., Yen, T.-F., Fuh, C.-S.: Fast and versatile algorithm for nearest neighbor search based on a lower bound tree. *Pattern Recognition* 40, 360–375 (2007)
12. Qin, Z., Bao, F., Li, A.: A Novel Image Fusion Method Based on SGNN. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) *ISNN 2005. LNCS*, vol. 3497, pp. 747–752. Springer, Heidelberg (2005)
13. Shi, W., Zhu, C., Tian, Y., Nichol, J.: Wavelet-based image fusion and quality assessment. *International Journal of Applied Earth Observation and Geoinformation* 6, 241–251 (2005)

Behaviour of Texture Features in a CBIR System

César Reyes¹, María Luisa Durán², Teresa Alonso¹, Pablo G. Rodríguez²,
and Andrés Caro²

¹ SICUBO S.L.TM

{creyes, talonso}@sicubo.com

² Depto. Ingeniería de Sistemas Informáticos y Telemáticos (DISIT)

Universidad de Extremadura, Spain

{mlduran, pablogr, andresc}@unex.es

Abstract. Searching and processing in databases of general and non-specific images are highly subjective. The process of texture feature extraction from images produces results of highly theoretical and mathematical character that have little to do with human perception. We present a method to select from low-level texture features, statistics and numerical groupings and to transform them into other high-level features, with visual meaning. We also aim to facilitate their use within CBIR systems. The detailed study of the composition and behaviour of the texture characteristics has enabled us to abstract and use them in an automated manner, similarly to how an observer would do.

Keywords: Texture features, semantic features, CBIR.

1 Introduction

Texture is a key component of human visual perception. Like colour or shape, texture is an essential feature to consider when querying image databases. Everyone can recognize texture but it is more difficult to define it. Unlike colour, texture occurs over a region rather than at a point or pixel. Texture features are usually defined purely by grey levels and they have qualities such as periodicity and scale; they can be described in terms of direction, coarseness, contrast, etc. That is what makes particularly interesting to work with textures and their extraction and selection processes.

The texture features of images play an important role in CBIR (*Content-Based Image Retrieval*) systems, especially if they are related to and used in conjunction with other features such as colour, shape, etc. Among the characteristics of texture, some of the most commonly used groups are second order statistics [2, 7] that base their performance on the grouping of pixels or on the repetitions of patterns, resulting in certain matrix values on which to do several specific calculations.

In a CBIR or MIR (*Multimedia Information Retrieval*) system, the process of feature extraction is as important as the selection process to achieve better information retrieval. Therefore, it is necessary to select an optimal group among features extracted from images either by discarding some irrelevant or unnecessary ones, or by giving them a perceptual meaning in order to achieve a better interpretation of the content of the images.

The selection of groups of textural features is a much more subjective task than, for example, the use of colour characteristics or the recognition of a shape and its position within an image. It is, therefore, a process where the observer must make a greater effort to automate the perception of the meaning of certain physical characteristics.

This paper proposes a way to combine different low level texture features obtained from an image by statistical methods, according to perceptual properties and behaviour. The purpose is to match each group of low level features with one given semantic characteristic.

2 Feature Extraction

The features obtained from the images for this paper's proposes are based on second order statistics. These statistics are structured according to three groups:

Grey Level Cooccurrence Matrix (GLMC). The spatial dependence of grey levels characterizes textures through a cooccurrence matrix of grey levels. Resolving the cooccurrence of grey levels may lead to the capability to describe the spatial relationships of grey levels within a pattern of texture. Likewise, this procedure can also be done so that it is invariant to monotonous changes of grey levels [6, 8].

Grey Level Run Length Matrix (GLRLM). A coarse texture is characterized by a large number of pixels in a neighbourhood with the same level of grey, whereas a fine texture is characterized by small neighbourhoods with the same level of grey. The length (measured in number of pixels) of these texels in different directions serves as a texture descriptor. A primitive texture can be regarded as a set of consecutive pixels with the same level of grey which we can call "run of grey level", and the length of this run is the number of pixels that form it [3, 5].

Neighbouring Grey Level Dependence Matrix (NGLDM). Introduced by Sun et al. [11] and completed by Berry et al. [1]. This method is robust to translations, as it relies on the direct relationship between an element and all its neighbouring pixels, rather than growing in one direction. It is based on the assumption that the matrix of spatial dependence of a grey level image can adequately specify textural information. The NGLDM matrix is calculated from the grey level ratio between each element of the image and all its neighbours at a certain distance [10].

3 Feature Selection

Traditional methods for the selection of features are based on the analysis of the discriminating capacity of each feature, with maintenance of independence among them. Methods such as Fisher's discriminant factor or Principal Components Analysis [12] are applied to achieve this aim. Our proposed method is based on the analysis of the relevance of features via non-expert users in mostly perceptual terms.

The low level features used have been grouped in subsets. These are based on tests done on a specific database that enables the extraction of similar images according to a given physically perceivable property. Hereby an intuitive identification of images is achieved by means of texture features of a higher level. The choice of low-level

characteristics in each group is based on experiments that combine them, first in small quantities, and then in larger groups. Tests tell us that some important features are based on common factors that make their grouping meaningful. Other features are based on independent factors that lead to a grouping that either subtracts meaning or may yield conflicting results.

We have used the system QatrisImanager to carry out all the experiments. This tool is a software package designed for the management of large amounts of images. It qualifies as a CBIR system, based on feature vector storage. The information is indexed by a Q-tree that allows multiattribute searches [4], and the system is provided with classification tools.

With the use of QatrisImanager [9] as the core analytical system, we can carry out tests on all possible combinations of selected features, while they can be tested also independently. We have found that certain groups of characteristics bring meaning to image retrieval; however, adding a texture characteristic to one of such groups makes it inconsistent, deriving into retrievals that present little meaning to the users.

With this process of testing, assessment and correction, we find five groups, identified by means of a given property or feature visible in the images. We call these groups uniformity, contrast, shade, linearity, and perpendicularity.

4 Giving Semantic Content

Figure 1 displays some samples from a collection of 80 artificial images created for the experiments. We have studied the behaviour of individual characteristics; we then tried to add those features whose method of calculation could provide information to the search. The analysis processes characteristics based both on the detailed study of the mathematical processes behind them, and on the experiments done under the supervision of users, who have contributed with their personal perceptions.

We have relied on the opinion of twelve users from SICUBO and University staff. Six of them are not directly related to QatrisImanager development, only four people are expert in image processing and other two of them take part in the development of algorithms for classification and indexation. Every user has completed 59 tests (not in the same day). Each test consists of watching 80 images decreasingly sorted by every low-level feature (we have analyzed 59 textural features; therefore we have carried out 59 tests per user). Users must annotate which properties (maximum 2) they perceive or miss in the placed images both in the first positions of the ranking and in the last ones respectively. We studied the answers of the users and determined that several tests registered the same results. These coincidences showed us which features would form the groups as well as their denominations.

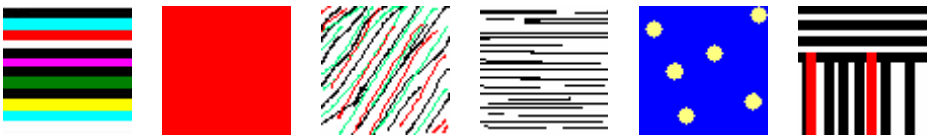


Fig. 1. Artificial images from the sample

Then we select those features that bring the same kind of meaning to the results obtained. We compare those results in relation to a high-level outcome feature that a given observer would emphasize when doing the same search for similar images. The goal is to achieve the results as observable and measurable data with which to assess the outcome accurately.

The measure of similarity is given by the distance between the images obtained by means of the function of the normalized measure of similarity:

$$\text{Similarity (i)} = \sum_{\forall c} \left(1 - \frac{\min(\text{ValueChar}(c)(i), \text{ValueChar}(c)(\text{query}))}{\max(\text{ValueChar}(c)(i), \text{ValueChar}(c)(\text{query}))} \right)$$

Where, $i=1, 2, \dots$ number of images in the collection, $c=1,2, \dots$ number of features. For each image of the collection (i), $\text{ValueChar}(c)(i)$ is the value for each of the texture features (c) we want to consider. And $\text{ValueChar}(c)(\text{query})$ is the value of the same feature calculated for the query image.

The retrieval of the images from the database is done with the values of the characteristics nearest to those of the characteristics of the original image, i.e., those images with the lowest values of the similarity function.

4.1 Set-Grouped Features

Uniformity. Composed of 10 low-level features; GLCM (Energy), NGLDM (NNU), GLRLM 0° - 45° - 90° - 135° (RLNU, RPC). The feature Energy obtained from the matrix of grey level cooccurrence is indicative of the homogeneity of an image; the greater the value of its energy, the greater uniformity of the image. A similar property is present in the NNU feature, belonging to the group of descriptors of grey levels for a neighbourhood and linked to the degree of an image roughness. Therefore, the values from these characteristics in different images indicate certain uniformity, a degree of homogeneity, and a similar size of patterning. We complement the group with RLNU and RPC, calculated from the run length matrix, which have low values for coarse and fine textures, respectively calculated for each of the direction spaces, 0° , 45° , 90° and 135° . These features increase the information of uniformity, adding directionality to homogeneity.

Contrast. Composed of 3 low-level features; GLCM (Contrast, Inertia, Dissimilarity). Contrast, Inertia and Dissimilarity, obtained from the grey levels of the cooccurrence matrix, are related to the continuity of values between adjacent pixels. High values indicate that there are sudden changes in tone in the image, i.e., abrupt changes in the intensity of the image. Therefore, these characteristics, on being grouped, enhance the meaning of the high-level feature contrast.

Shade. Composed of 20 low-level features; GLRLM 0° - 45° - 90° - 135° (GLNU, SRLGE, SRHGE, LRLGE, LRHGE). Continuing with features extracted from the run length matrix, there are images that, after removing their colour information, have the same "tone", i.e. they visually present similar organization changes in intensity. This perception is measurable through a set of characteristics extracted from GLRLM: SRLGE, SRHGE, LRLGE, and LRHGE. All are calculated in the four directions 0° ,

45°, 90° and 135°, which provide information about the spatial distribution of the linear and tonal continuity of the image.

Linearity. Composed of 8 low-level features; GLRLM 0°-45°-90°-135° (LRE, SRE). The characteristics SRE and LRE of the Grey Level Run Length Matrix emphasize respectively the presence of short and long rows in an image; therefore, they are a measure of the linearity of an image. LRE presents values raised for soft textures and SRE values raised for abrupt textures. The calculation of these characteristics for each of the directional matrix (0°, 45°, 90° and 145°) provides a quantification of the linearity of an image.

Perpendicularity. Composed of 16 low-level characteristics; GLRLM 0°-90° (LRE, SER, GLNU, RLNU, RPC, SRLGE, SRHGE, LRHGE). If we simply looked for information on the direction spaces 0° and 90°, we would get images that have a sharp linear nature in such directions. This property known as perpendicularity is used to calculate features drawn from the GLRLM, just in the horizontal and vertical directions. The resulting images exhibit high linearity in parallel to the limits of the image.

From all 59 analyzed characteristics, the five aforementioned high-level properties use only 41. Those are the ones with greater discriminating power and which modify values in greater proportion, similarly to the perception of an observer. The characteristics of the three groups of descriptors depend on the calculation of a distance either between neighbours (GLCM, NGLDM) or levels of grey (GLRLM).

Once obtained, these five high level features we have proceeded to its validation and implantation in a CBIR system, i.e., the QatrisImanager software developed by the SICUBOTM company [9].

5 Results

In order to validate the behaviour of these semantic features, we have used a collection of 2500 images with different textures extracted from major groups. Those groups are visually distinct, e.g., brick, wood, metal, clothes, floor, and construction. In them, the properties are visually recognizable, e.g., directionality, uniformity, etc.

Figure 2 shows some images per group. Figure 3 shows a snapshot of QatrisImanager doing a search on perpendicularity. The query image is shown on the top left side, while the feature selected for the search on similarity appears on the top right. The results are at the bottom, ordered by a lower value of the distance function. As it can be seen, the query image has a horizontal and vertical directionality, easily recognizable in the images returned. The user can change, in the graphical user interface, the priority of features by slice bars and buttons, so as to get the retrieved images that are most similar to the query.

The examples shown in Figure 4 reveal similarity in relation to the query image on the basis of one of the obtained features. In (a) the images are estimated as presenting a nearby variability in the changes of intensity levels. In (b), in spite of the different colours, the homogeneity of texture makes them similar in uniformity to the query image. In (c) the query image presents a high vertical linearity, as the images demonstrate. In (d) all the returned images present a similar tone.

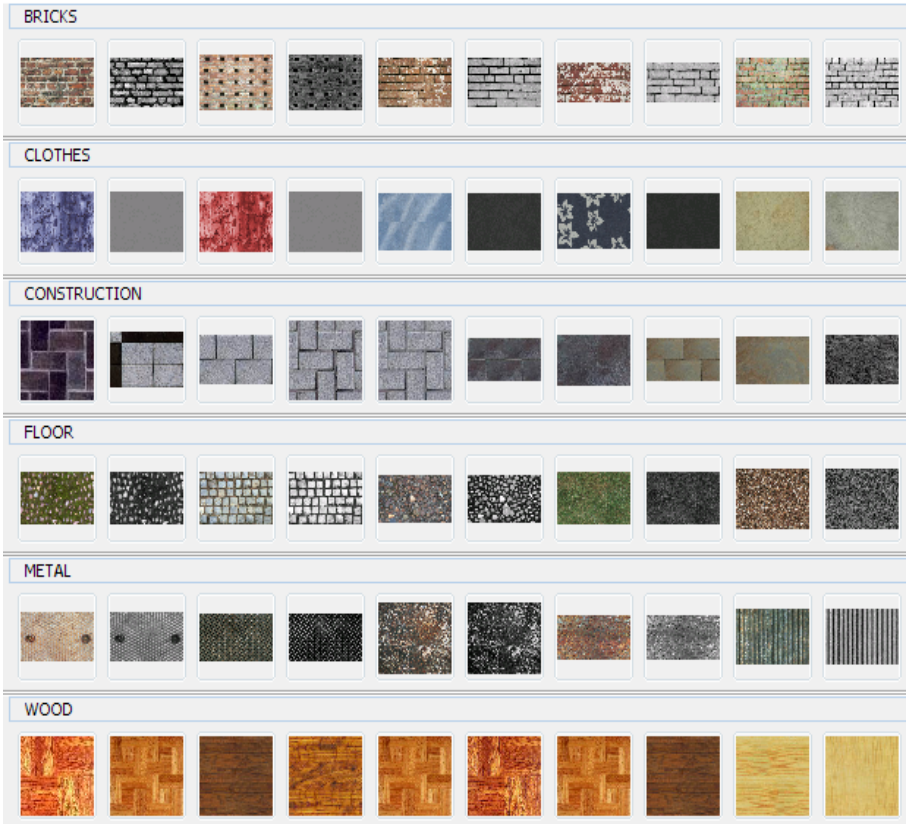


Fig. 2. Sample images

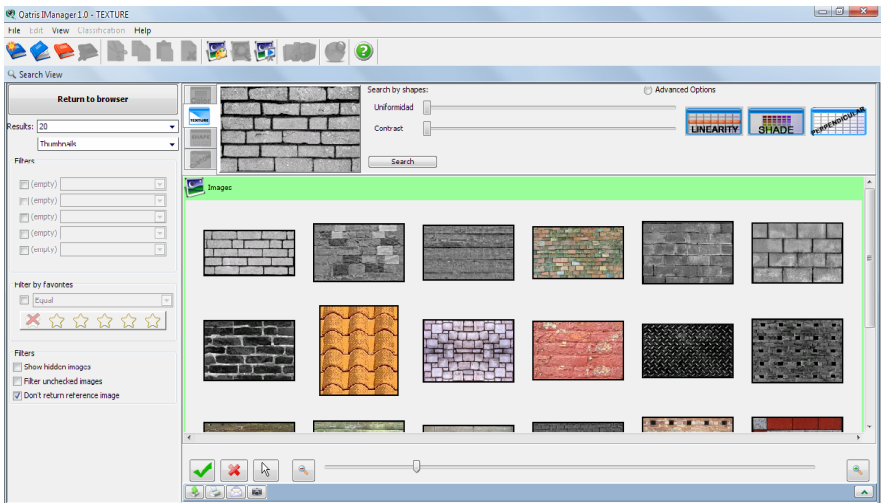


Fig. 3. Example of search for “Perpendicularity” in Qatrismanager

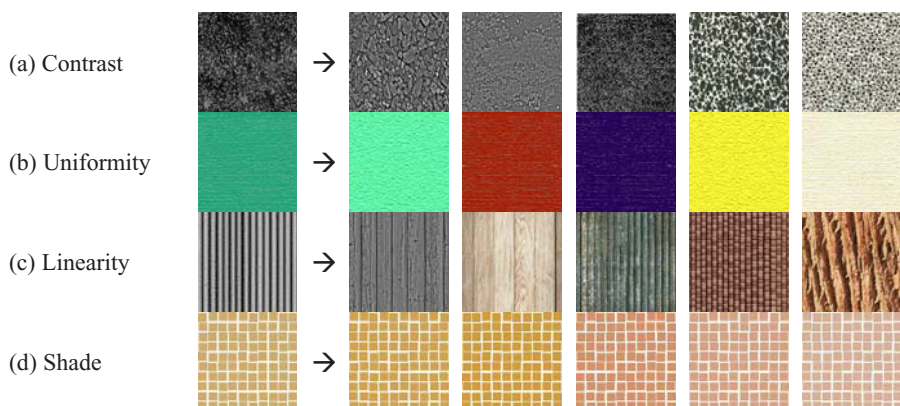


Fig. 4. Examples of searches for Contrast (a), Uniformity (b), Linearity (c) and Shade (d)

6 Conclusions

One further step is taken in this paper to fill the semantic gap of CBIR systems. The abstraction of the characteristics from these developments, nearing to human perception, facilitates the selection process and makes the automation of these systems closer to human behaviour.

Because texture features in an image inherently present complex mathematical components, an important obstacle rises and limits its verification by people in real time. Thus, more specific groupings and visual meaning provision in feature extraction from images tend to make the overall process of information retrieval and search more simple and intuitive.

The outcomes of our queries have in fact demonstrated that the semantic features obtained are similar to the results returned by an user that chooses this feature among all the images in a collection.

Acknowledgments. The authors wish to acknowledge and thank the Sicubo S.L.TM for their strong belief and firm support in this work. Our research has been funded by the Ministerio de Educación y Ciencia (Spain's Government Board – Research Project # TIN2005-05939) and the Junta de Extremadura (Regional Government Board – Research Project # 3PR05B027).

References

1. Berry, J.R., Goutsias, J.: A comparative study of matrix measures for maximum likelihood texture classification. *IEEE Trans. SMC* 21(1), 252–261 (1991)
2. Connors, R.W., Harlow, C.A.: A theoretical comparison of texture algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-2* 3, 204–222 (1980)
3. Chu, A., et al.: Use of Grey Value Distribution of Run Lengths for Texture Analysis. *Pattern Recognition Letters* 11, 415–420 (1990)

4. Jurado, E., Barrena, M.: Efficient Similarity Search in Feature Spaces with the Q-tree. In: Manolopoulos, Y., Návrat, P. (eds.) ADBIS 2002. LNCS, vol. 2435, pp. 177–190. Springer, Heidelberg (2002)
5. Galloway, M.M.: Texture Analysis using Grey Level Run Lengths. *Computer Graphics and Imag. Processing* 4, 172–179 (1975)
6. Haralick, R.M., Shanmugam, K., Dinstein, X.: Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics* (1979)
7. Haralick, R.M.: Statistical and structural approaches to texture. *Proc. IEEE* 67, 786–804 (1979)
8. Haralick, R.M., Shapiro, L.G.: *Computer and Robot Vision*. Addison-Wesley, Reading (1993)
9. <http://www.sicubo.com>
10. Siew, L.H., Hodgson, R.M., Wood, E.J.: Texture Measures for Carpet Wear Assessment. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 10(1), 92–104 (1988)
11. Sun, C., Wee, W.G.: Neighbouring grey level dependence matrix. *Computer Vision, Graphics, Image Processing* 23, 341–352 (1982)
12. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*. Academic Press, London (1999)

Object Tracking Using Grayscale Appearance Models and Swarm Based Particle Filter

Bogdan Kwolek

Rzeszów University of Technology, W. Pola 2, Rzeszów, Poland
bkwolek@prz.rzeszow.pl

Abstract. We propose a hybrid tracking algorithm consisting of two trackers built on grayscale appearance models. In a first tracker we employ an object template that consists of several grayscale image patches. Every patch votes for the possible positions of the object undergoing tracking. A grayscale appearance model that is learned on-line is used in a supplementing tracker. A particle swarm optimization algorithm is utilized to shift particles toward more promising regions in the probability density function. Experimental results show that the hybrid tracker outperforms each of the trackers.

Keywords: Natural computation, hybrid systems, particle swarm optimization.

1 Introduction

Natural computation refers to computational systems that use ideas and get inspiration from natural systems, including biological, sociological, ecological and physical systems [1]. The aim of such research is to develop methods for dealing with large, complex, and dynamic problems. Computing inspired by nature takes nature as inspiration for the development of more robust and effective techniques. This often leads to combination of natural patterns and behaviors, and the use of visual systems to gather the knowledge. The human visual system enables us to perform tasks such as object recognition and categorization. Understanding the internal processes that lead to such perceptual capabilities is one of aims of this emerging discipline.

The work [2] presents arguments for the hypothesis that the human visual system consists of a number of interacting but autonomous systems that process different modalities and complement each other. Such subsystems can serve mutually in process of learning and bootstrapping the object representations. Cognitive science states that complex entities are perceived as composition of simple elements. Objects are represented through such components and the relations between them [3].

Visual tracking is one of the central problems and has received considerable attention in past years. The goal of tracking is to automatically find the same object in adjacent frames in a video sequence. Despite attempts to make visual tracking resistant to losing the object undergoing tracking, most currently available algorithms inevitably fails under large visual perturbations including rapid unexpected motions, changes in ambient illumination, and occlusions. In this work we present a hybrid tracker that learns on-line within a co-training framework. To improve the tracking efficiency and effectiveness we employ particle swarm optimization and multi-part object representation.

2 Swarm Based Particle Filtering

2.1 Particle Filtering

Particle filtering [4] is recursive process in which particles are repeatedly selected, moved forward according to a probabilistic motion model that is dispersed by an additive random noise component, evaluated against the observation model, and re-sampled according to their weights to avoid degeneracy. The key idea underlying particle filtering is to approximate a probability distribution by a weighted particle set $S = \{(\mathbf{s}_n, \pi_n) \mid n = 1, \dots, N\}$. In such representation each particle \mathbf{s} stands for a hypothetical state of the object, whereas the corresponding non-negative weight π represents the sampling probability. The weights are normalized such that $\sum_{n=1}^N \pi_n = 1$. Two important components of each particle filter (PF) are motion model $p(\mathbf{z}_t \mid \mathbf{z}_{t-1})$ describing the state propagation and observation model $p(\mathbf{y}_t \mid \mathbf{z}_t)$ describing the likelihood that the state \mathbf{z}_t causes the observation \mathbf{y}_t . The evolution of the sample set takes place by drawing new samples from a suitably chosen proposal distribution, which may depend on the old state and the new measurements, i.e. $\mathbf{z}_{n,t} \sim q(\mathbf{z}_{n,t} \mid \mathbf{z}_{n,t-1}, \mathbf{y}_t)$, and then propagating each sample according to probabilistic motion model of the target. To give a particle representation of the posterior density the particles are set to $\pi_{n,t} \propto p(\mathbf{y}_t \mid \mathbf{z}_{n,t})p(\mathbf{z}_{n,t} \mid \mathbf{z}_{n,t-1})/q(\mathbf{z}_{n,t} \mid \mathbf{z}_{n,t-1}, \mathbf{y}_t)$. The particles should be re-sampled according to their weights to avoid degeneracy.

The particle filter converges to the optimal filter in some sense as the number of particles grows to infinity. The most important property of the particle filter is its ability to handle complex, non-Gaussian and multimodal posterior distributions. However, the number of particles required to adequately approximate the conditional density grows exponentially with the dimensionality of the state space. This poses practical difficulties in applications such as articulated body tracking [5]. In such applications, weakness of the particle filter consists in that the particles can not cluster around the true state of the object and instead they can migrate towards local maximas in the posterior distribution. If particles are too diffused the track of the object can be lost. When the observation likelihood lies in the tail of prior distribution, most of the particles will become insignificant weights. If the system model is inaccurate, the prediction based on the system model may not be a good one. Predictions of poor quality can also be caused by the simulation in the particle filtering, when the system noise is large and the number of particles is not sufficient. In order to cope with the mentioned effects several improvements to basic algorithm have been proposed, among others solutions combining extended Kalman filter/unscented Kalman filter with generic particle filter [4]. Such filters incorporate the current observation to generate the better importance density than the generic particle filter that uses the prior as the importance density.

2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic, population-based evolutionary algorithm for solving nonlinear, multimodal optimization problems [6]. This optimization

technique, which is based on swarm intelligence has been inspired by social behavior of fish schooling or bird flocking. PSO shares several similarities with evolutionary computation techniques. The swarm is initialized with a population of individuals representing random solutions and then it searches for optima by updating particle values. The particles iteratively evaluate the fitness of the candidate solutions and remember the locations $\hat{\mathbf{x}}_i$, where they achieved their best fit so far. Another important value for the swarm is the location $\hat{\mathbf{g}}$ corresponding to the best fitness, which was obtained so far by any particle. Therefore, while exploring the hyperspace spanned by the possible parameters the particles employ reasoning capabilities about their own best location and the knowledge of the global best one. Through changing the velocity of each particle toward best fit $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{g}}$ locations, the particle swarm finds $\hat{\mathbf{x}}^* \subseteq \mathcal{X} \subseteq \mathbb{R}^m$ such that $\hat{\mathbf{x}}^* = \arg \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) = \{ \mathbf{x}^* \in \mathcal{X} : f(\mathbf{x}^*) \leq f(\mathbf{x}) \ \forall \mathbf{x} \in \mathcal{X} \}$, and $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is a fitness function.

Let there be N particles, each with associated locations $\mathbf{x}_i \in \mathbb{R}^m$ and velocities $\mathbf{v}_i \in \mathbb{R}^m$. The optimization algorithm takes the following form:

- Initialize \mathbf{x}_i and \mathbf{v}_i . Then $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$, $\hat{\mathbf{g}} = \arg \min_{\mathbf{x}_i} f(\mathbf{x}_i)$
- While stopping criteria are not satisfied
- For each particle
 - Update the particle locations: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$
 - Update particle velocities: $\mathbf{v}_i \leftarrow \omega \mathbf{v}_i + c_1 \mathbf{r}_1 \circ (\hat{\mathbf{x}}_i - \mathbf{x}_i) + c_2 \mathbf{r}_2 \circ (\hat{\mathbf{g}} - \mathbf{x}_i)$
 - Update the local best values, if $f(\mathbf{x}_i) < f(\hat{\mathbf{x}}_i)$, then $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$
 - Update the global best, if $f(\mathbf{x}_i) < f(\hat{\mathbf{g}})$, then $\hat{\mathbf{g}} \leftarrow \mathbf{x}_i$

The symbol ω represents an inertial constant, \circ stands for array multiply, c_1 and c_2 are constants, which balance the influence of the particle’s local best and the global best, respectively, \mathbf{r}_1 and \mathbf{r}_2 are vectors of uniform random numbers between 0 and 1.

The above algorithm was inspired by animal behavior. A few simple rules result in complex action. It has been demonstrated that in many problems PSO achieves better results in a faster way in comparison with other methods. PSO has been successfully applied in wide range of applications. Another reason that PSO is attractive is that it needs a few parameters to adjust.

2.3 Swarm Based Particle Filter

PSO algorithm and PF employ different schemes to concentrate particles near best locations. PSO shifts particles towards the optimum by updating position and velocity of each particle using its best previous value and global best value of all particles. Particle filter utilizes general prediction-update framework in which probabilistic motion and observation models are used alternately during approximating the conditional density. The algorithm for swarm-based particle filter is as follows:

1. Initialization. Sample $\mathbf{z}_{1,0}, \dots, \mathbf{z}_{N,0}$ i.i.d from initial density p_0
2. Importance Sampling/Propagation. Sample $\mathbf{z}_{i,t}$ from $p(\mathbf{z}_t | \mathbf{z}_{i,t-1})$, $i = 1, \dots, N$

3. Initialize PSO. Assign initial values \mathbf{v}_i , then $\mathbf{x}_i \leftarrow \mathbf{z}_{i,t}$, $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$, $\hat{\mathbf{g}} = \arg \min_{\mathbf{x}_i} f(\mathbf{x}_i)$

3a. While not stop

For each particle

Update the particle locations: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$

Update particle velocities: $\mathbf{v}_i \leftarrow \omega \mathbf{v}_i + c_1 \mathbf{r}_1 \circ (\hat{\mathbf{x}}_i - \mathbf{x}_i) + c_2 \mathbf{r}_2 \circ (\hat{\mathbf{g}} - \mathbf{x}_i)$

Update the local best values, if $f(\mathbf{x}_i) < f(\hat{\mathbf{x}}_i)$, then $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$

Update the global best, if $f(\mathbf{x}_i) < f(\hat{\mathbf{g}})$, then $\hat{\mathbf{g}} \leftarrow \mathbf{x}_i$

3b. $\mathbf{z}_{i,t} \leftarrow \hat{\mathbf{x}}_i$

4. Updating. Compute $\hat{p}(\mathbf{z}_t | \mathbf{y}_{1:t}) = \sum_{i=1}^N \pi_{i,t} \delta(\mathbf{z} - \mathbf{z}_{i,t})$ using normalized weighs:

$$\pi_{i,t} = p(\mathbf{y}_t | \mathbf{z}_{i,t}), \quad \pi_{i,t} = \pi_{i,t} / \sum_{j=1}^N \pi_{j,t}, \quad i = 1, \dots, N$$

5. Resampling. Sample $\mathbf{z}_{1,t}, \dots, \mathbf{z}_{N,t}$ i.i.d from $\hat{p}(\mathbf{z}_t | \mathbf{y}_{1:t})$

6. $t \leftarrow t+1$, go to step 2.

The algorithm was compared with generic particle filter on example from work [4]:

$$\begin{aligned} x_{k+1} &= 1 + \sin(0.04\pi k) + \rho x_k + u_k, \quad x_0 = 0, \\ y_k &= \begin{cases} 0.2x_k^2 + v_k & k \leq 30 \\ 0.5x_k - 2 + v_k & k > 30 \end{cases} \end{aligned} \quad (1)$$

where ρ is a system parameter. The system noise u_k follows a Gamma distribution $\Gamma(3, \theta)$, where θ is the scale parameter. The observation noise follows a Gaussian distribution $\mathcal{N}(0, 1.0e-5)$. We simulated 100 independent runs of length 60 with random initializations. To compare filters we calculated the root mean square (RMS) error of the estimated states for each run, and the means of the RMS of the 100 runs. Using the generic particle filter built on 200 and 100 particles, respectively, we obtained the following RMS: 0.07, 0.2, in time 4.4 and 2.2 sec., respectively. Using the swarm-based particle filter built on 50 particles and performing 3 and 5 iterations in PSO, respectively, we obtained the following RMS: 0.11, 0.07 in time 2.2 and 3.1 sec., respectively. From the above results we can observe that in time 2.2 sec. the generic particle filter produces estimates with RMS equal to 0.2, whereas the swarm based particle filter gives RMS equal to 0.11. The generic particle filter needs 4.4 sec. to yield estimates with RMS equal to 0.07, whereas the swarm-based particle filter takes 3.1 sec. The better results are due to swarm intelligence, which considers the distribution both from local and global perspective.

3 Appearance Based Person Tracking

3.1 Adaptive Appearance Models

The intensity model has been inspired by the *WSL* model [7]. The model consists of three components called wandering, stable and lost. During model learning each component votes according to its level of creditability. The model can adapt to slowly changing appearance and provides robustness to partial occlusions. It has been shown

to yield reliable tracing of pedestrians, where a second AdaBoost-based tracker with a different failure mode was used to support the learning of the pedestrian’s appearance [8].

The appearance model that is assigned to a single gray observation d_t consists of three components, namely, the W -component characterizing variations between two consecutive frames, S -component expressing stable component structures within the former observations, and C -component representing pixel values from the initial object template. Such a model exhibits the object appearances in frames up to time $t - 1$. It is a mixture of Gaussians with centers $\{\mu_{i,t} | i = w,s,c\}$, their corresponding variances $\{\sigma_{i,t}^2 | i = w,s,c\}$, and mixing probabilities $\mathbf{m}_t = \{m_{i,t} | i = w,s,c\}$. The mixture probability density for a new data d_t conditioned on the former observations is given by:

$$p(d_t | d_{t-1}, \mu_{s,t-1}, \sigma_{s,t-1}^2, \mathbf{m}_{t-1}) = m_{w,t-1} p_w(d_t | \mu_{w,t-1}, \sigma_w^2) + m_{s,t-1} p_s(d_t | \mu_{s,t-1}, \sigma_{s,t-1}^2) + m_{c,t-1} p_c(d_t | \mu_{c,0}, \sigma_c^2). \tag{2}$$

The mean $\mu_{w,t-1}$ in the wandering component is set to the observation d_{t-1} and the variance σ_w^2 is fixed. The stable component expresses the appearance properties that are relatively stable over time. A Gaussian density function with slowly accommodated parameters $\mu_{s,t-1}, \sigma_{s,t-1}^2$ expresses the evolution of such temporally stable image observations. The fixed component accounts for data holding information from the initial object appearance. The mean value is the observation d_0 taken from the initial frame and the variance is fixed at σ_c^2 . The model is learned on-line using the EM algorithm. Details of the learning algorithm can be found in [8].

A swarm particle filter build on the learned on-line appearance models has been constructed and then used to track people in surveillance videos. The state transition model is a random walk in which a new predicted state is composed through adding to the previous state a zero mean Gaussian noise with a covariance Σ . A more complex dynamic model can be employed if relevant. The observation likelihood is calculated according to the following formula:

$$p(\mathbf{y}_t | \mathbf{z}_t) = \prod_{j=1}^M \sum_{i=w,s,c} \frac{m_{i,t}}{\sqrt{2\pi\sigma_{i,t}^2(j)}} \exp\left[-\frac{d_t(j) - \mu_{i,t}(j)}{2\sigma_{i,t}^2(j)}\right] \tag{3}$$

where d_t denotes the value of gray pixel, M is the number of pixels in the appearance model.

3.2 Multi-patch Object Tracking

In work [9] it has been demonstrated that multi-part object representation leads to better tracking. The authors proposed a method consisting in dividing the object to be tracked into non-overlapping rectangular regions and then computing in each sub-region a color histogram. The observation model has been constructed under assumption that image data in different sub-regions are independent. The object likelihood was proportional to the exponential of negative sum of squared distances between histograms. In

[10], rectangle sub-regions that are defined within the extent of the object to be tracked are employed to calculate the averaged color of pixels. The object tracking is achieved through an exhaustive search in a confidence region. An occlusion model has been developed to discriminate between good and spurious measurements.

Cognitive science states that complex entities are perceived as composition of simple elements. Objects are represented through such components and the relations between them [3]. One of the disadvantages of color histograms is the loss of spatial information. To incorporate such information in the object representation we divide the object template into adjoining cells regularly spaced within the extent of the target to be tracked. We compute histograms within such regularly spaced patches using a fast method that has been proposed in [11]. Given the estimated position in the previous frame and the histograms within object at the estimated position we employ the chi-square test between such histograms and histograms extracted from cells within the template at the candidate position. The χ^2 test is given by: $\chi^2 = \sum_i ((h_{e,i} - h_{c,i})^2 / (h_{e,i} + h_{c,i})^2)$, where $h_{e,i}$ and $h_{c,i}$ represent the number of entities in the i -th bin of the histograms, and a low value for χ^2 indicates a good match. Such values are transformed into likelihoods through the usage of the exponential function. We seek for the object in the new frame within a search window, which center is located at the previous object position. At each candidate position we compare the corresponding histograms and the result is utilized to vote for the considered position. Every cell votes in its own map. Then we combine the votes in a relevance map, which encodes hypothesis where the target is located in the image. In order to cope with partial occlusions we employ a simple heuristics aiming at detecting outliers. If the difference between corresponding histograms is below a certain level, then in such case the score in the relevance map remains unchanged. The level for each cell is determined individually using the distances between histograms from the initial template and corresponding histograms from few first frames. Similar test is performed with respect to actual medians.

3.3 Nature Inspired Cooperative Tracking of the Object

Obtaining a collection for on-line unsupervised learning is a complex task. In work [2] it has been argued that the human visual system consists of a number of interacting but still autonomously operating subsystems that process different object representations. Within such a mechanism, subsystems can serve mutually in process of learning and bootstrapping of object representations. This motivated us to construct an object tracker consisting of two independent trackers with different failure modes, complementing each other and operating in co-training framework. The co-training approach has been utilized in work [12], where a tracker starts with a small training set and increases it by co-training of two classifiers, operating on different features. In work [8] a co-training mechanism supports boosting of features as well as unsupervised learning of adaptive appearance models.

In the multi-patch based object tracker an accommodation of the histograms over time takes only place if there is a significant overlap between the object templates, i.e. between the cell-based object template and the adaptive appearance-based template. The multi-patch object tracker takes the advantages of the collaborative tracker that keeps the object location far more precisely. Owing to this the multi-patch tracker is

prevented from the template drift, in which the template gradually slips away from the object until the tracker is following something else entirely. The second tracker can fail in case of partial occlusions or when the target appearance undergoes major changes. The cells that are marked as possibly occluded indicate pixels, which are not employed in learning of the stable components.

4 Experiments

The tests were done on a sequence¹ of images 288 high and 384 wide. The tracker has been initialized in frame #700, see Fig. 1 that depicts some results obtained by multi-patch based tracker. Frames #888 - #939 show the tracking during temporal occlusions. We can see that in frame #929 the track has been temporally lost. The tracker recovered the track in frame #939. Frames #1554 and #1566 illustrate the tracking during another occlusion. The tracker failed in frame #1645. The results were achieved using four patches. The size of the search window was 15 x 15. Due to considerable jitter of the tracked window the histogram has not been adapted in the course of the tracking.



Fig. 1. Person tracking using multi-patch algorithm. Frames #700, #888, #889, #899, #900, #909, #928, #929, #939, #1554, #1566, #1645.

Figure 2 depicts some results that were obtained using swarm-particle filter built on adaptive appearance models. The tracking has been done using only 20 particles. As we can observe the tracker keeps the specific region of the person far more precisely. However, it fails in frame #1566. The generic particle filter loses the person earlier. Additionally, the jitter of the tracking window in such a tracker is considerable.



Fig. 2. Person tracking using swarm particle filter built on adaptive appearance models. Frames #888, #889, #899, #900, #909, #928, #929, #939, #1554, #1566.

The hybrid tracker outperforms each of the trackers. In particular, the jitter of the window is comparable to the jitter in the swarm-particle filter based tracker. This allows us to carry out the adaptation of the histogram in the multi-patch based tracker. Both adaptation of the histogram and learning of the appearance models is done only in case of significant overlap between windows determined by each of the trackers. The person has been tracked through the whole sequence, see Fig. 3. The deterministic,

¹ Downloaded from site at: <http://groups.inf.ed.ac.uk/vision/CAVIAR/WalkByShop1cor>



Fig. 3. The location of the tracking window determined by the hybrid tracker. Frames #888, #889, #899, #900, #909, #928, #929, #939, #1554, #1566.

multi-patch based tracker is several times slower than the particle based tracker but it is more resistant to partial occlusions. The hybrid tracker takes the advantages of both trackers. In consequence it is able to precisely track the object with small window jitter, see frames #888 - #939 as well as to cope with occlusions, see frames #1554 and #1566.

5 Conclusions

The main contribution of this work is a hybrid algorithm consisting of two trackers that have different failure modes and complement each other. This cooperative framework leads to better tracking. The main ingredient of the tracking algorithm is swarm-based particle filter that has been tested in simulation as well as on real video data. The multi-patch tracker acknowledged its great usefulness in tracking objects undergoing occlusions.

References

1. de Castro, L.N.: Fundamentals of Natural Computing: An Overview. *Physics of Life Reviews* 4(1), 1–36 (2007)
2. Zeki, S.: Localization and Globalization in Conscious Vision. *Annual Review Neuroscience* 24, 57–86 (2001)
3. Ommer, B., Buhmann, J.M.: Learning Compositional Categorization Models. In: *European Conf. on Computer Vision*, vol. III, pp. 316–329 (2006)
4. van der Merve, R., de Freitas, N., Doucet, A., Wan, E.: The Unscented Particle Filter. *Advances in Neural Information Processing Systems* 13, 584–590 (2001)
5. Schmidt, J., Fritsch, J., Kwolek, B.: Kernel Particle Filter for Real-Time 3D Body Tracking in Monocular Color Images. In: *IEEE Int. Conf. on Face and Gesture Rec.*, pp. 567–572 (2006)
6. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: *Proc. IEEE Int. Conf. on Neural Networks*, pp. 1942–1948. IEEE Press, Piscataway (1995)
7. Jepson, A.D., Fleet, D.J., El-Maraghi, T.: Robust Online Appearance Models for Visual Tracking. *IEEE Trans. on PAMI* 25(10), 1296–1311 (2003)
8. Kwolek, B.: Learning-Based Object Tracking Using Boosted Features and Appearance Adaptive Models. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2007. LNCS*, vol. 4678, pp. 144–155. Springer, Heidelberg (2007)
9. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002. LNCS*, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
10. Fieguth, P., Terzopoulos, D.: Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates. In: *Proc. IEEE Conf. on Comp. Vision and Patt. Rec.*, pp. 21–27 (1997)
11. Porikli, F.: Integral Histogram: A Fast Way to Extract Histogram in Cartesian Spaces. In: *IEEE Computer Society Conf. on Pattern Rec. and Computer Vision*, pp. 829–836 (2005)
12. Levin, A., Viola, P., Freund, Y.: Unsupervised Improvement of Visual Detectors Using Co-Training. In: *Proc. Int. Conf. on Comp. Vision*, pp. 626–633 (2004)

Extraction of Geometrical Features in 3D Environments for Service Robotic Applications*

Paloma de la Puente, Diego Rodríguez-Losada, Raul López, and Fernando Matía

Intelligent Control Group - Universidad Politécnica de Madrid, José Gutiérrez Abascal 2
28006 Madrid, Spain
paloma.puente.yusty@alumnos.upm.es, diego.rlosada@upm.es

Abstract. Modeling environments with 3D feature based representations is a challenging issue in current mobile robotics. Fast and robust algorithms are required for applicability to navigation. We present an original and effective segmentation method that uses computer vision techniques and the residuals from plane fitting as measurements to generate a range image from 3D data acquired by a laser scanner. The extracted points of each region are converted into plane patches, spheres and cylinders by means of least-squares fitting.

Keywords: Laser-range finder, 3D point cloud, range-image based segmentation, least-squares fitting.

1 Introduction

In the mobile robotics community significant research efforts are lately focusing on acquiring and processing information about the 3D nature of the environments found in real world scenarios. Overcoming the limitations imposed by the 2D models is of great importance regarding safe navigation but also to provide further knowledge to be used in other robotic tasks. This is the main field of application towards which this work is orientated. Yet it could be of interest in some other areas such as building modeling or industrial reconstruction and identification, being there several publications that address the issues of 3D data acquisition, segmentation and fitting in these contexts [1], [2]. The most remarkable particular aspect of the 3D feature-based navigation problem is the need for real-time algorithms, especially when performing *Simultaneous Localization and Mapping* (SLAM). To comply with this requirement, a practical approach may go for obtaining and storing the data in an organized way. If methods that consider the extra information hence available are applied subsequently compact models that replace the initial large and scarcely meaningful point cloud can be built in reasonable time.

Although there are other means to obtain 3D data from the environment, up to now the laser range scanner is the most popular stereoeptive sensor used in mobile robotics. The newly developed Swiss Ranger Camera from the CSEM (Swiss Center for

* This work is funded by the Spanish Ministry of Science and Technology, DPI2007-66846-c02-01.

Electronics and Microtechnology), commercialized by Acroname Robotics, may become a good alternative, but it is yet not very suitable for mapping activities due to its reduced field-of-view (of about 45°) and the noisiness of its measurements [3]. Stereo vision systems are also less accurate than the laser range finder and they pose more difficulties to the data manipulation and interpretation processes.

Here we put forward our ideas to obtain 3D primitives frequently found by mobile robots in indoor environments out of distance measurements provided by such laser devices. Our main contribution is a novel method to solve the segmentation of the laser data by integrating vision based techniques. Taking advantage of the ordered nature of the data, real-time capabilities are achieved.

The paper is organized as follows. Section 2 constitutes a brief explanation of the hardware and system configuration we have employed in the 3D data acquisition. Section 3 introduces the segmentation problem, revises some related work and presents the range image based technique we propose to solve it keeping in mind the specific conditions of the feature based navigation goal. In Section 4 we expose how the final fitting to geometrical primitives is carried out. Section 5 shows some experimental results. Finally, Section 6 summarizes our conclusions and future work.

2 3D Data Acquisition

One of the simplest solutions to gather 3D data with a 2D laser range finder consists of displacing it along the perpendicular direction to its scanning plane, normally horizontal. Equivalently, instead of actually moving one single laser scanner up and down, adding a second laser vertically mounted is a practical option when the robot operates in strictly flat terrain [4][5]. Another possibility is to make the laser collect data in a vertical plane while it turns around its upright axis by means of an extra servo drive [6]. One laser rotating around its optical axis, with a yaw movement, generates 3D data too [7], but requires higher mechanical effort and leads to each individual 2D scan not having much intuitive sense.

The most natural approach seems to be that of employing a nodding system as is done in [3] [5] [8] [9]. We have mounted a SICK LMS200 laser device on top of a servo pan-tilt unit (PowerCube Wrist 070, by Amtec Robotics) positioned at the front of our Pioneer 3AT robot, Nemo (Fig. 1).

A data server running on an onboard laptop computer sends synchronous updated information about odometry, PW70 and laser readings at clients' cyclical requests within a capture procedure in a stop-and-go manner. The communication protocols between the laptop and each of both devices and the robot work via serial connection. The port's baud rate for the laser scanner is set at 500 kb (using an external USB to 422 interface) so as to gain velocity and permit a good precision in synchronization with the PW70. This is of utmost importance to avoid distortions when applying the relative transformations to calculate each point's $[x, y, z]$ coordinates.

The laser beams of a 2D scan are emitted between 0° and 180° at regular intervals $d\theta$ we have fixed in 1° . We obtain a 3D scan by varying the tilt angle.



Fig. 1. Our robot Nemo equipped with a laser mounted on a wrist for 3D data acquisition

3 Segmentation

Segmentation undertakes the partition of a 3D point cloud into smaller subsets representing objects of interest (different surfaces, here). Points identified as part of the same region are allotted the same tag so that they can be treated as raw data of a sought-after feature. This is a key subject in feature based navigation for mobile robots, being especially challenging the fact that it is not known a priori what stuff the scanned scene may contain apart from those elements we intend to recognize [3].

3.1 Related Work

In computer vision, the segmentation problem has received considerable attention for the last decades. Notwithstanding, the adaptation of developed techniques and algorithms to robotic applications with data coming not from cameras but from laser sensors has not been much exploited. Indeed, a great deal of recent research has focused on extracting enhanced levels of detail in complex scenes and better preserving the objects' exact shape, [10] [11]. Robots do not yet need that much knowledge about small items in the environment and this approach implies too complicated, time consuming solutions that are therefore not adequate for navigation.

Three main categories of existing methods are commonly used, namely edge-based segmentation, region-based segmentation and scanline-based segmentation [12].

- Edge-based segmentation consists of detecting those points where discontinuities in certain local properties are noteworthy and then grouping together points that fall inside the closed boundaries found.
- Region algorithms follow a merging schema. They usually start from some seed points to which near points are added in accordance to a certain similarity criterion.
- Scanline-based segmentation in first place analyses each scanline or row of a range image independently and then checks whether extracted features of consecutive 2D scans are combinable into the same 3D feature.

In [13] and [14] a series of range image segmentation methods within these three classes are thoroughly described and evaluated. We think that edge-based algorithms are more suitable here because they are faster and more direct and elegant than the other two approaches. Some hybrid techniques complementing it with any of the other two are interesting as well [15].

We have observed that a vast majority of segmentation algorithms are limited to planar surfaces, mainly among those few explicitly designed for mobile robotics applications. An assessment of four different line extraction algorithms to separate points from different planar surfaces in service robotics has recently been published, [16]. In [3] two region-based algorithms for planar segmentation of range images were presented; they are both capable of processing probabilistic data in order to do 3D SLAM based upon planar features. Reference [17] is the closest work to the approach presented in this paper. It addresses the segmentation of laser data in mobile robotics by applying common range image processing operations, but also sticks to planar patches. The edge-based algorithm there presented introduces a new measure, the so called *bearing angle*, to characterize the surfaces and then apply a standard *Sobel* border detector.

The algorithm reported in [12] inspired the key idea of the method we propose. It is a region growing strategy for unstructured point clouds from industrial environments.

3.2 Vision-Based Range Image Segmentation

The first step towards classifying points in a 3D scan through range image segmentation may be to fill an image like structure with the available raw measurements coming from the sensor and then apply computer vision edge detection algorithms. This allows finding the *jump* edges of the data, i.e. those associated to depth discontinuities between two physically separated (or not intersecting) surfaces. However, different objects and surfaces are usually touching one another (things standing on the floor are in permanent contact with it, walls join at corners...). In that case, boundaries are defined by *crease* or *roof* edges (changes in orientation) that are totally compatible with smooth variations in distance measurements. Similar conclusions were pointed out in [15], [17].

The solution we suggest comes from the fact that the residuals of fitting one point's neighborhood to a plane will have higher values at both jump and crease edges. Assuming the laser is correctly calibrated and errors in the measurements can be modeled as Gaussian noise with a not too high standard deviation (5mm for the LMS 200), most points inside a planar patch have tiny residuals. Points from other kind of primitives have residuals proportional to the local curvature. The basic features we are going to use for representation (planar patches, cylinders and spheres) have theoretical constant curvatures for all of their points. Nevertheless, segmentation would not be affected if that were not the case, for residuals would still change smoothly and have greater values at break points.

For each 3D point, we select its neighborhood with an eight connectivity criterion and compute the residual of the plane fitting by applying the equations in Subsection 4.1. Adequately scaling the obtained values, the image of edges is created. In order to make sure borders do not break improperly, a *closing* (dilate + erode) morphological operation is applied twice. To eliminate false borders provoked by noise the image is binarized so that only important edges are left. Once this has been accomplished, a floodfilling algorithm is used to assign the same color to all pixels inside each large enough region enclosed by the remaining borders. Different regions are given different colors, so points belonging to important features are labeled univocally. The following pseudocode is an outline of the whole algorithm:

```

Input: matrix A containing the 3D data
for every (i, j) element in A
    {neigh} ← {A(i-1, j-1), ..., A(i, j), ..., A(i+1, j+1)}
    image (i, j) = k * PlaneLSQresidual(neigh)(See 4.1)
end
image = Closing(image) × 2
image = Thresholding(image)
for each pixel in mask of image
    if (black pixel) %pixel not labeled yet
        new_color
        FloodFilling(image, new_color)
        r = NewRegion(new_color) %create new labeled region
        R ← r
    else
        for all existing regions
            if (color(pixel) == color(region))
                r ← 3D point associated to pixel (i, j)
                break;
            end
        end
    end
end
F = {∅} %the extracted features
for r ∈ R
    if (PlaneLSQresidual(r) < threshold)
        Plane p = PlaneLSQ(r)
        F ← p
    else if (CylinderLSQresidual(r) < threshold)(See 4.2)
        Cylinder c = CylinderLSQ(r)
        F ← c
    else if (SphereLSQresidual(r) < threshold)(See 4.3.)
        Sphere s = SphereLSQ(r)
        F ← s
    end
end

```

Fig. 2. Pseudocode for segmentation and fitting

4 Model Fitting

4.1 Plane Fitting

In the Hessian Normal Form of a plane the model parameters are the normal vector components (n_x, n_y, n_z) and the perpendicular distance to the origin, d . The parameters

of the plane that best fits a set of N points $\mathbf{p}_i = (x_i, y_i, z_i)$ in a least squares sense can be obtained by solving a 3×3 eigenvalue problem. We aim at minimizing the sum of squares of the orthogonal distances from the points to the estimated surface (the residual):

$$R(n_x, n_y, n_z, d) = \sum_{i=1}^N (n_x x_i + n_y y_i + n_z z_i - d)^2 . \tag{1}$$

Differentiation with respect to the four parameters yields a 4×4 linear system on the parameters, but the problem can be simplified. If the partial derivative with respect to d is obtained, it is easy to see that the best possible estimation of the plane passes through the center of gravity (o_x, o_y, o_z) of the points. The normal vector can then be obtained as the eigenvector associated to the smallest eigenvalue of:

$$A = \begin{bmatrix} \sum_{i=0}^N x_i'^2 & \sum_{i=0}^N x_i' y_i' & \sum_{i=0}^N x_i' z_i' \\ \sum_{i=0}^N x_i' y_i' & \sum_{i=0}^N y_i'^2 & \sum_{i=0}^N y_i' z_i' \\ \sum_{i=0}^N x_i' z_i' & \sum_{i=0}^N y_i' z_i' & \sum_{i=0}^N z_i'^2 \end{bmatrix} , \tag{2}$$

being $\mathbf{p}'_i = (x'_i, y'_i, z'_i) = (x_i - o_x, y_i - o_y, z_i - o_z)$.

The residual value will be $R = \sum_{i=1}^N (\mathbf{p}'_i \cdot \mathbf{n})^2$.

4.2 Sphere Fitting

A way to determine an estimation of the center coordinates (x_c, y_c, z_c) and the radius r is to minimize the following function for all the points p_i :

$$(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 - r^2 . \tag{3}$$

To make the equation linear the variable $t = (x_c^2 + y_c^2 + z_c^2) - r^2$ is introduced so that the problem comes to solving

$$\begin{bmatrix} -2x_1 & -2y_1 & -2z_1 & 1 \\ -2x_2 & -2y_2 & -2z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -2x_N & -2y_N & -2z_N & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ t \end{bmatrix} - \begin{bmatrix} -x_1^2 - y_1^2 - z_1^2 \\ -x_2^2 - y_2^2 - z_2^2 \\ \vdots \\ -x_N^2 - y_N^2 - z_N^2 \end{bmatrix} = 0 . \tag{4}$$

Or in matrix notation, $Mx = B$. So, $x = (M^T M)^{-1} B$ is the solution we sought.

4.3 Cylinder Fitting

A cylinder may be represented by a normalized vector along its axis (a, b, c) , a point on the axis (x_0, y_0, z_0) and the radius R . Equation 5 holds:

$$[c(y_i - y_0) - b(z_i - z_0)]^2 + [a(z_i - z_0) - c(x_i - x_0)]^2 + [b(x_i - x_0) - a(y_i - y_0)]^2 = R^2. \quad (5)$$

Developing the former expression and grouping the coefficients, it can be put as: $Ax_i^2 + By_i^2 + Cz_i^2 + Dx_iy_i + Ex_iz_i + Fy_iz_i + Gx_i + Hy_i + Iz_i + J = 0$. Dividing it by A, we get a linear system with nine unknowns that can be solved in a least squares fashion. The following approximations have proven accurate enough in our experiments.

If E/A , F/A and G/A are close to zero, B/A close to 1 implies $(a, b, c) = (0, 0, 1)$ and C/A close to 1 implies $(a, b, c) = (0, 1, 0)$. Otherwise, $A = 2/(1+B/A+C/A)$. If A and B are close to 1 $(a, b, c) = (-EA/2C, -F/2c, (1-C)^{1/2})$. If A is close to 1 but B is not close to 1 $(a, b, c) = (-D/2b, (1-B)^{1/2}, -F/2b)$. If A is not close to 1 $(a, b, c) = ((1-A)^{1/2}, -D/2a, -E/2A)$. If the resultant vectors are not normalized they must be divided by their magnitude.

Once (a, b, c) has been obtained, the point on the axis is computed by solving:

$$\begin{bmatrix} -2(b^2 + c^2) & 2ab & 2ac \\ 2ab & -2(a^2 + c^2) & 2bc \\ 2ac & 2bc & -2(a^2 + b^2) \\ a & b & c \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} G \\ H \\ I \\ 0 \end{bmatrix}. \quad (6)$$

The estimation of the radius value is done by:

$$R^2 = (b^2 + c^2)x_0^2 + (a^2 + c^2)y_0^2 + (a^2 + b^2)z_0^2 - 2bcy_0z_0 - 2bcy_0z_0 - 2bcy_0z_0 - J$$

5 Experiments

To capture the 3D data, the laser is tilted from -0.4 radians (upwards looking) to 0.3 radians (downwards looking) at a constant speed of 0.05 rad/s, which results in a 70x181 matrix of measurements. We process the central 70x131submatrix to avoid problems caused by the points being too near from each other at the extreme lateral angles of the view area. Fig. 3 shows some results of the image processing steps of our algorithm. They correspond to two 3D scans taken at our laboratory with different room configurations.

The segmentation process takes about 1 second. It is considerably below the time consumed by region growing algorithms, applied when not much speed is required, such as [12], which we implemented for evaluation. Our development has been in C++ using the OpenCv libraries.

Table 1. shows some information about a series of experiments conducted by moving the robot across the corridors and rooms of our laboratory. Under segmentation

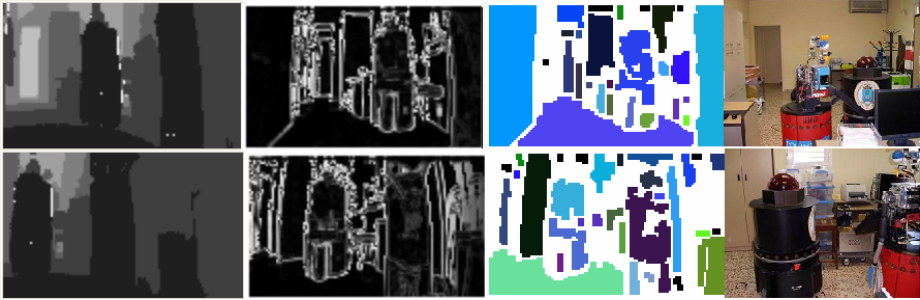


Fig. 3. Images on the left contain raw distance measurements. Next images show the plane fitting residuals followed by the region extraction output. Images on the right are the environments where the data for these experiments were taken, with some modifications.

Table 1. Summary of results from several different experiments

Experiment	Extracted features	Under segmentation	Over segmentation	False fittings	Processing time (s)
1	7	1 (14.3%)	0(0%)	0	1.06
2	13	0 (0%)	1(7.7%)	0	1.00
3	17	1(5.9%)	1(5.9%)	0	1.05
4	10	0(0%)	2(20%)	0	1.03
5	12	2(16.7%)	1(8.3%)	1	1.00
6	17	0(0%)	4(23.5%)	0	1.04
7	18	1(5.6%)	2(11.1%)	0	1.04
8	13	1(7.7%)	1(7.7%)	1	1.06
9	20	2(10%)	1(5%)	0	1.07
10	23	0(0%)	6(26.1%)	0	0.98

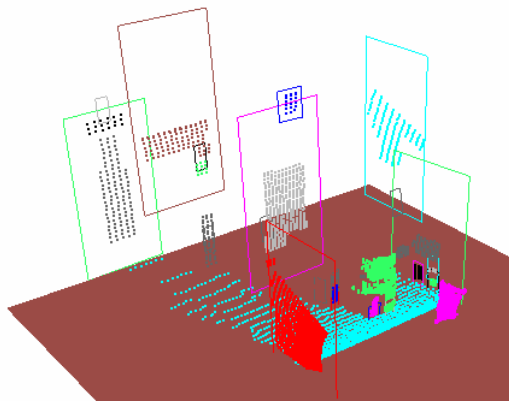


Fig. 4. Features extracted from the environment by applying the algorithms we propose

consists of having different surfaces belonging to the same extracted region. It may lead to obtaining a wrong plane model (false fitting). Over segmentation happens when several contiguous patches are acquired from a surface.

Some changes in the threshold values were made so as to adapt the segmentation process to the particular conditions of the experiments. Although at first it may leave out a greater number of points than other methods, ours allows for the extraction of the most important elements of the structure in cluttered environments, consuming less time. As already mentioned, surfaces other than planes can be obtained this way.

The final feature extraction for experiment number 10 is shown in Fig. 4. No false fittings were done and only few unimportant objects were missing. The outcome for other 3D scans was similar.

6 Conclusions and Future Working Lines

We have presented a novel range-image based algorithm employing computer vision techniques. The residuals of a least squares fitting process for each pixel's neighborhood are used to detect surface discontinuities in the data and achieve effective fast performance, as is needed in mobile robotics applications. The segmented 3D points are fitted to planar, cylindrical or spherical models to generate a compact representation of the environment.

One of the earliest improvements we plan to introduce is the combination of this segmentation results with the detection of edges at both sides of the image, to widen the field of view. We need to make the algorithm completely robust by applying local checking methods that will eliminate the under-segmentation cases. Furthermore, once the different regions have been found some close points left out after the image processing steps may be added in order to enlarge the patches. Finally, patches' boundaries should be determined and patches belonging to the same object ought to be merged (we think about convex hulls determination and applying boolean operations among polygons). This will directly deal with the over-segmentation cases.

A higher level goal we have is the integration of features to generate room and corridor models, aiming at building rich maps containing topological information.

References

1. Dorninger, P., Nothegger, C.: 3D Segmentation of Unstructured Point Clouds for Building Modelling. In: Stilla, U., et al. (eds.) PIA 2007. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 36 (3/W49A) (2007)
2. Delft University of Technology. Department of Earth Observation & Space Systems, <http://www.lr.tudelft.nl/live/pagina.jsp?id=14f981bc-23a0-4dd8-8653-3db5a5f6f627&lang=en>
3. Weingarten, J.: Feature-based 3D SLAM. PhD thesis, EPFL (2006)
4. Mahon, I., Williams, S.: Three Dimensional Robotic Mapping. In: Proc. of the Australasian Conference on Robotics & Automation (ACRA) (2003)
5. Hähnel, D., Burgard, W., Thrun, S.: Learning Compact 3d Models of Indoor and Outdoor Environments with a Mobile Robot. *Robotics and Autonomous Systems* 44(1), 15–27 (2003)

6. Brenneke, C., Wulf, O., Wagner, B.: Using 3D Laser Range Data for SLAM in Outdoor Environments. In: Proc. of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Las Vegas, Nevada (2003)
7. Kohlhepp, P., Pozzo, P., Walther, M., Dillmann, R.: Sequential 3DSLAM for Mobile Action Planning. In: Proc. of 2004 IEEE/RSJ International Conference, vol. 1, pp. 722–729 (2004)
8. Cole, D., Newman, P.: Using Laser Range Data for 3D SLAM in Outdoor Environments. In: Proc. of the IEEE International Conference on Robotics and Automation, Florida (2006)
9. Nüchter, A., Lingemann, K., Hertzberg, J., Surmann, H.: 6D SLAM—3D Mapping, Outdoor Environments. *Journal of Field Robotics* 24(8/9), 699–722 (2007)
10. Han, F., Tu, Z., Zhu, S.C.: Range Image Segmentation by an Effective Jump-diffusion Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(9), 1138–1153 (2004)
11. Bellon, O., Silva, L.: New Improvements to Range Image Segmentation by Edge Detection. *IEEE Signal Processing Letters* 9(2) (2002)
12. Rabbani, T., van den Heuvel, F.A., Vosselman, G.: Segmentation of Point Clouds Using Smoothness Constraint. In: ISPRS Commission V Symposium Image Engineering and Vision Metrology, Dresden, vol. XXXVI, Part 5 (2006)
13. Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B.: An Experimental Comparison of Range Image Segmentation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(7), 673–689 (1996)
14. Jiang, X., Bowyer, K., Morioka, Y., Hiura, S., Sato, K., Inokuchi, S., Bock, M., Guerra, C., Loke, R., du Buf, J.: Some Further Results of Experimental Comparison of Range Image Segmentation Algorithms. In: Proc. of the 15th International Conference on Pattern Recognition (2000)
15. Sappa, A.D., Devy, M.: Fast Range Image Segmentation by an Edge Detection Strategy. In: Third IEEE International Conference on 3D Digital Imaging and Modeling, p. 292 (2001)
16. Gächter, S., Nguyen, V., Siegart, R.: Results on Range Image Segmentation for Service Robots. In: Proc. of the 4th IEEE International Conference on Computer Vision Systems (2006)
17. Harati, A., Gächter, S., Siegart, R.: Fast Range Image Segmentation for Indoor 3D-SLAM. In: 6th IFAC Symposium on Intelligent Autonomous Vehicles (2007)

Hybrid GNG Architecture Learns Features in Images

José García-Rodríguez, Francisco Flórez-Revuelta,
and Juan Manuel García-Chamizo

Department of Computer Technology. University of Alicante. Apdo. 99. 03080 Alicante, Spain
{jgarcia, florez, juanma}@dtic.ua.es

Abstract. Self-organising neural networks try to preserve the topology of an input space by using their competitive learning. This capacity has been used, among others, for the representation of objects and their motion. In this work we use a kind of self-organising network, the Growing Neural Gas, to represent objects as a result of an adaptive process by a topology-preserving graph that constitutes an induced Delaunay triangulation of their shapes. In this paper we present a new hybrid architecture that creates multiple specialized maps to represent different clusters obtained from the multilevel multispectral threshold segmentation.

Keywords: Self-organising maps, topology preservation, topologic maps, thresholds, data clustering, Delaunay Triangulation.

1 Introduction

Self-organising neural networks, by means of a competitive learning, make an adaptation of the reference vectors of the neurons as well as the interconnection network among them; obtaining a mapping that tries to preserve the topology of an input space. Besides, they are capable of a continuous re-adaptation process even if new patterns are entered, with no need to reset the learning.

These capacities have been used for the representation of objects [1] (Figure 1) and their motion [2] by means of the Growing Neural Gas (GNG) [3] that has a learning process more flexible than other self-organising models, like Kohonen maps [4] and more flexible and faster than Topology Representing Networks [5].

In this work we introduce a new hybrid architecture based on the GNG that combines the graph obtained from the learning algorithm of the net with a threshold segmentation technique, using as input a single image or an image sequence and representing the different objects or features that appears in the images.

Segmentation using multilevel multispectral threshold [6] is used to obtain clusters for any different feature or object in the image, running a new specialized map for any different cluster obtained in the segmentation step even the background will be represented with a map. The novelty of this method is the use of GNG to represents and improves segmentation at the same time and also the specialization of the maps in different features.

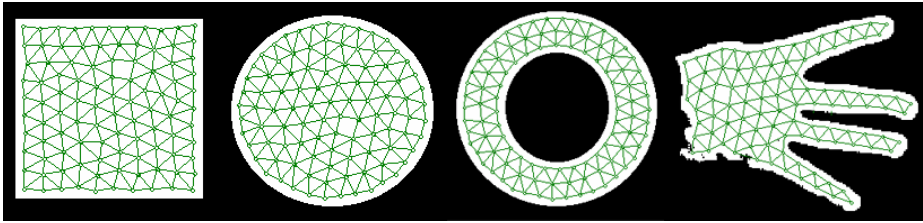


Fig. 1. Representation of two-dimensional objects with a self-organising network

The remainder of the paper is organized as follows: section 2 provides a detailed description of the topology learning algorithm GNG and in section 3 an explanation on how we can apply GNG to represent objects is given. Section 4 presents the hybrid GNG based architecture to learn and represent image features and some experimental results are presented, followed by our major conclusions and further work.

2 Topology Learning

One way of selecting points of interest in 2D shapes is to use a topographic mapping where a low dimensional map is fitted to the high dimensional manifold of the shape, whilst preserving the topographic structure of the data. A common way to achieve this is by using self-organising neural networks where input patterns are projected onto a network of neural units such that similar patterns are projected onto units adjacent in the network and vice versa. As a result of this mapping a representation of the input patterns is achieved that in post-processing stages allows one to exploit the similarity relations of the input patterns. Such models have been successfully used in applications such as speech processing [4], robotics [7,8] and image processing [9]. However, most common approaches are not able to provide good neighborhood and topology preservation if the logical structure of the input pattern is not known a priori. In fact, the most common approaches specify in advance the number of neurons in the network and a graph that represents topological relationships between them, for example, a two-dimensional grid, and seek the best match to the given input pattern manifold. When this is not the case the networks fail to provide good topology preserving as for example in the case of Kohonen's algorithm.

The approach presented in this paper is based on self-organising networks trained using the Growing Neural Gas learning method [3], an incremental training algorithm. The links between the units in the network are established through competitive hebbian learning [10]. As a result the algorithm can be used in cases where the topological structure of the input pattern is not known a priori and yields topology preserving maps of feature manifold [5].

2.1 Growing Neural Gas

With Growing Neural Gas (GNG) [3] a growth process takes place from minimal network size and new units are inserted successively using a particular type of vector quantisation [4]. To determine where to insert new units, local error measures are

gathered during the adaptation process and each new unit is inserted near the unit which has the highest accumulated error. At each adaptation step a connection between the winner and the second-nearest unit is created as dictated by the competitive hebbian learning algorithm. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology based on some measure. Also the ending condition could it be the insertion of a predefined number of neurons or a temporal constrain. In addition, in GNG networks learning parameters are constant in time, in contrast to other methods whose learning is based on decaying parameters.

In the remaining of this Section we describe the growing neural gas algorithm. The network is specified as:

- A set N of nodes (neurons). Each neuron $c \in N$ has its associated reference vector $w_c \in R^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons.
- A set of edges (connections) between pairs of neurons. These connections are not weighted and its purpose is to define the topological structure. An *edge aging scheme* is used to remove connections that are invalid due to the motion of the neuron during the adaptation process.

The GNG learning algorithm to approach the network to the input manifold is as follows:

1. Start with two neurons a and b at random positions w_a and w_b in R^d .
2. Generate a random input pattern ξ according to the data distribution $\mathcal{P}(\xi)$ of each input pattern. In our case since the input space is 2D, the input pattern is the (x, y) coordinate of the points belonging to the object. Typically, for the training of the network we generate 1000 to 10000 input patterns depending on the complexity of the input space.
3. Find the nearest neuron (winner neuron) s_1 and the second nearest s_2 .
4. Increase the age of all the edges emanating from s_1 .
5. Add the squared distance between the input signal and the winner neuron to a counter error of s_1 such as:

$$\Delta error(s_1) = \|w_{s_1} - \xi\|^2 \tag{1}$$

6. Move the winner neuron s_1 and its topological neighbours (neurons connected to s_1) towards ξ by a learning step \mathcal{E}_w and \mathcal{E}_n , respectively, of the total distance:

$$\Delta w_{s_1} = \mathcal{E}_w (\xi - w_{s_1}) \tag{2}$$

$$\Delta w_{s_n} = \mathcal{E}_n (\xi - w_{s_n}) \tag{3}$$

7. If s_1 and s_2 are connected by an edge, set the age of this edge to 0. If it does not exist, create it.
8. Remove the edges larger than a_{max} . If this results in isolated neurons (without emanating edges), remove them as well.

9. Every certain number λ of input signals generated, insert a new neuron as follows:
 - Determine the neuron q with the maximum accumulated error.
 - Insert a new neuron r between q and its further neighbour f :

$$w_r = 0.5(w_q + w_f) \tag{4}$$

- Insert new edges connecting the neuron r with neurons q and f , removing the old edge between q and f .
 - Decrease the error variables of neurons q and f multiplying them with a constant α . Initialize the error variable of r with the new value of the error variable of q and f .
10. Decrease all error variables by multiplying them with a constant β .
 11. If the stopping criterion is not yet achieved, go to step 2. (In our case the criterion is the number of neurons inserted)

In summary, the adaptation of the network to the input space takes place in step 6. The insertion of connections (step 7) between the two neurons nearer to each one of the input patterns establishes, finally, an induced Delaunay triangulation by the input space. The elimination of connections (step 8) eliminates the edges that no longer must comprise of this triangulation. This is made eliminating the connections between neurons that no longer are next or that they have nearer neurons. Finally, the accumulation of the error (step 5) allows to identify those zones of the input space where it is necessary to increase the number of neurons to improve the mapping.

3 Representation of 2D Objects with GNG

Given an image $I(\chi, y) \in \mathcal{R}$ we perform the transformation $\psi_{\mathcal{T}}(\chi, y) = \mathcal{T}(I(\chi, y))$ that associates to each one of the pixels its probability of belonging to the object, according to a property \mathcal{T} . For instance, in figure 2, this transformation is a threshold function.

If we consider $\xi = (\chi, y)$ and $\mathcal{P}(\xi) = \psi_{\mathcal{T}}(\xi)$, we can apply the learning algorithm of the GNG to the image I , so that the network adapts its topology to the object. This adaptive process is iterative, so the GNG represents the object during all the learning.

As a result of the GNG learning we obtain a graph, the Topology Preserving Graph $\mathcal{TPG} = \langle \mathcal{N}, \mathcal{C} \rangle$, with a vertex (neurons) set \mathcal{N} and an edge set \mathcal{C} that connect them (figure 1). This \mathcal{TPG} establishes a Delaunay triangulation induced by the object [6].

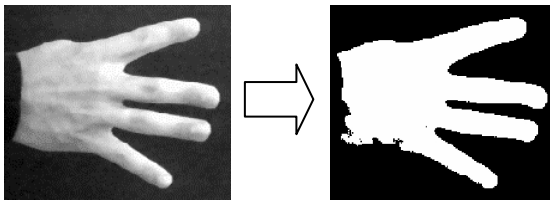


Fig. 2. Silhouette extraction

Also, the model is able to characterize the diverse parts of an object, or several present objects in the scene with the same values for the visual property \mathcal{T} , without initialize different data structures for each one of the objects. This is due to the capacity of GNG to divide itself when removing neurons.

3.1 Multilevel Multispectral Threshold Segmentation

Segmentation using multilevel multispectral threshold [6] is used to obtain clusters for different features or objects in the image running a new map for any different cluster obtained in the segmentation step even the background will be represented with a map.

Multilevel threshold classifies a point $p(x,y)$ with a visual property \mathcal{T} as belonging to one object class if $T1 < \mathcal{T} < T2$ with $T1$ and $T2$ different thresholds. For colour segmentation is necessary to use a colour model [11] with more than one variable to consider like HSI (Hue, Saturation, Intensity) model what is called multispectral threshold.

The prediction of colour pixels belonging to a target object in the segmentation of image sequences is a useful technique to segment, represent and track objects in realistic conditions, with changes in the illumination conditions.

4 Hybrid GNG-Based Architecture to Learns Features in Images

The hybrid GNG architecture creates a new map or cluster for any of the different ranges or features that we define to be considered in the learning process following a hybrid approach that combines multispectral multilevel threshold segmentation with a self-organising neural network graph representation. In figure 3 the architecture of the system is presented.

The process to obtain the representation of the image features would be based on the following scheme:

- Calculation of a priori transformation function in order to segment objects or features in the image
- Learning features with GNG updating thresholds and creating a new cluster (map) if necessary to represent a new feature in the image
- Repeat until every different feature in the image had been represented.

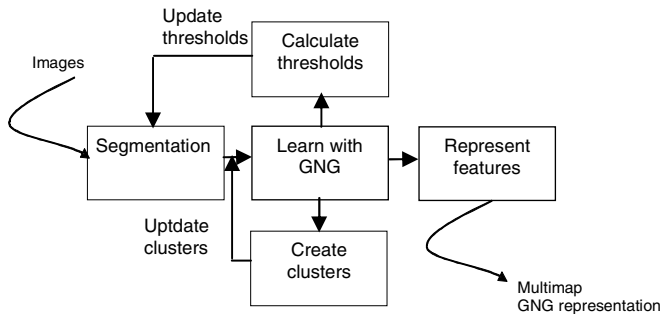


Fig. 3. Incremental GNG-based architecture

4.1 Experiments

We have tested the architecture to represent some images. In the example we show a synthetic image representing some geometric figures with different colours and another one with a hand gesture.

To apply that technique we have used a system of segmentation based on multi-level multispectral thresholds and using the HSI colour model simplified to Hue and Saturation variables.

Figure 4 shows the representation of a synthetic image with geometric figures in different colours (on the left). The system runs a new specialized map for any of the different features that appears in the image: circle, triangle, square and background (on the right). In this case no threshold has been predefined since the idea is to learn every different feature appearing in the image.

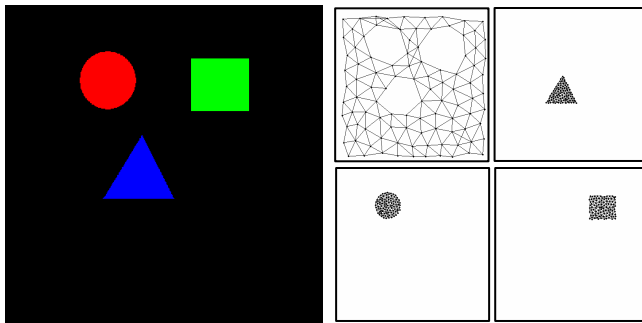


Fig. 4. Synthetic image representation with Hybrid GNG-based Architecture



Fig. 5. Hand gesture image representation with Hybrid GNG-based Architecture

In figure 5 a hand gesture is represented using as a feature to learn, skin colour of the hand. The HS Initial parameters are H ($T_1=100, T_2=180$) S ($T_1=0, T_2=50$) normalized between 0-255. Image on the left shows the original image with a complex background, image on the middle present the segmented image base on applied segmentation and image on the right shows the GNG graph representation.

The GNG parameters used for the simulations are: $\lambda= 1000, \epsilon_w = 0.1, \epsilon_n = 0.001, \alpha = 0.5, \beta= 0.95, \alpha_{max} = 250, N=50$. Topology preservation measures comparison with other representation models can be found in [12].

5 Conclusions and Further Work

In this paper, we have demonstrated the capacity of representation of bi-dimensional objects by a self-organising neural network. Establishing a suitable transformation function, the model is able to adapt its topology to images of the environment. Then, a simple, but very rich representation is obtained.

The model, by its own adaptation process, is able to divide itself so that it can characterize different fragments from an object or different objects with the same visual property.

A hybrid architecture based on GNG neural network combined with segmentation using multilevel multispectral threshold obtains clusters for different features or objects in the image running a new map for any different cluster obtained in the segmentation step even the background will be represented with a map.

Finally, the iterative and parallel performance of the presented representation model is the departure point for the development of high performance architectures that supply a characterization of an object depending on the time available.

References

1. Flórez, F., García, J.M., García, J., Hernández, A.: Representation of 2D Objects with a Topology Preserving Network. In: Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems (PRIS 2002), Alicante, pp. 267–276. ICEIS Press (2001)
2. Flórez, F., García, J.M., García, J., Hernández, A.: Hand Gesture Recognition Following the Dynamics of a Topology-Preserving Network. In: Proc. of the 5th IEEE ICAFG, pp. 318–323 (2001)
3. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. In: Tesauro, G., et al. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 625–632. MIT Press, Cambridge (1995)
4. Kohonen, T.: Self-Organising Maps. Springer, Berlin (2001)
5. Martinez, T., Schulten, K.: Topology Representing Networks. *Neural Networks* 7(3), 507–522 (1994)
6. Kerfoot, I.B., Bresler, Y.: Theoretical Analysis of Multispectral Image Segmentation Criteria. *IEEE Transactions on Image Processing* 8(6), 798–820 (1999)
7. Ritter, H., Schulten, K.: Topology conserving mappings for learning motor tasks. In: *Neural Networks for Computing*, AIP Conf. Proc. (1986)
8. Martinez, T., Ritter, H., Schulten, K.: Three dimensional neural net for learning visuomotor-condition of a robot arm. *IEEE Transactions on Neural Networks* 1, 131–136 (1990)
9. Nasrabati, M., Feng, Y.: Vector quantisation of images based upon kohonen self-organising feature maps. In: Proc. IEEE Int. Conf. Neural Networks, pp. 1101–1108 (1998)
10. Martinez, T.: Competitive hebbian learning rule forms perfectly topology preserving maps. In: *ICANN* (1993)
11. Skarbek, W., Koschan, A.: Colour Image Segmentation - A Survey, Technical Report 94-32, Technical University Berlin (October 1994)
12. García, J., Flórez, F., García, J.M.: Measuring GNG Topology Preservation in Computer Vision Applications. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4253(3), pp. 424–431. Springer, Heidelberg (2006)

Information-Theoretic Measures for Meta-learning

Saddys Segrera, Joel Pinho, and María N. Moreno

Department of Computer Science and Automatics, University of Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
{saddys, joelpl, mng}@usal.es

Abstract. Information-theoretic measures are suitable to characterize datasets with discrete attributes (or continuous which can be transformed). They can find information that can be decisive in order to analyze the behavior of different learning algorithms with specific datasets. The objective of the work presented in this paper is to study by means of three similar datasets from UCI Repository Machine Learning, the possible reasons for which breast-cancer-wisconsin dataset, in comparison with other 20 datasets, showed in a previous research that Stacking by Meta-Decision Trees (MDT) was significant better than all other multiclassifier models, including Stacking by Multi-Response Linear Regression (MLR). In our experiments the proportion of missing values, among other significant changes in different measure values, provided evidences about the possible origin of the different behaviors presented by these multiclassifier schemes depending on data characteristics.

Keywords: meta-learning, classification, stacking.

1 Introduction

Meta-learning is defined as the process of supervised learning that takes place from the information generated by the initial or base classifiers. This means, a technique to unify the results of multiple classifiers. The idea is to generate a system that performs the base classifiers functionality and increase the precision by means of the improvement of the form in which they correlate themselves [1], which leads to an efficient reduction of the space of incorrect predictions [2].

One technique addressed to study meta-learning is the characterization of datasets. Suitable data characterization is very important for the meta-learning. The complexity of data mining tasks is related to the characteristics of datasets and the inductive bias of learning algorithms [3].

The objective of this paper is the characterization of three datasets. Breast-cancer-wisconsin dataset was the only of 21 datasets from UCI Machine Learning Repository in the schemes (multiclassifiers) used by [4] in what Stacking [5] with meta-decision trees (MDT) was significantly better in all the cases. Inspired by this results, we compared breast-cancer-wisconsin with other two datasets (hepatitis and vote), which have similar composition of attributes, but had different behaviors in the research mentioned before.

We want to search similarities and differences in behaviors of Stacking with MDT and Stacking with Multi-response linear regression (MLR) and to provide some relative contributions related to information-theoretic measures for Stacking meta-learning.

This paper has been organized as following. In section 2 we summarize a group of related works based in dataset characterization techniques. The information-theoretic measures used in our experiments to the dataset characterization are described in section 3. The characterization of breast-cancer-wisconsin, hepatitis and vote datasets from UCI Machine Learning Repository and the analysis of the experimental results obtained are included in section 4. Finally, the conclusions are presented in section 5.

2 Related Works

Dataset characterization techniques are used in order to describe the problem that will be studied, including simple measures as number of attributes and number of classes; statistical measures as variance and average of numerical attributes; and measures based in theory of information as entropy of classes and attributes [6]. Nevertheless, there exists the need to improve the efficiency of the meta-learning developing better meta-attributes and selecting those which provide more information.

The first initiative to characterize datasets to predict the execution of a classification algorithm was made in [7]. Since that, three main strategies have developed for dataset characterization [8], and to suggest in this way what algorithm is most adapted for a specific dataset. One of them is the technique that describes the characteristics of the dataset using statistical and informative measures [9, 10]. STATLOG project [10] allowed the description of datasets by their information and statistical properties. The authors identified three categories of data characteristics: simple, statistical and information theory based measures. METAL [11] project is another example that allows characterizing data for meta-learning by means of different measures. Statistical characteristics are mainly appropriated for continuous attributes, while information-theoretic measures are more appropriated for discrete attributes [3]. On the other hand, in [3, 12] a second strategy is used for dataset characterization. In this case, the characteristics of the induced hypothesis as a way to represent the own dataset are considered. The third strategy consists of characterizing a dataset using the behavior of a system of simplified classifiers named landmarking [13].

Whereas other approaches typically describe a data set with statistical measures and information of attributes, landmarking proposes to enrich such description with fast and easy operation measures from simple learning algorithms. Learning algorithm profiles have been also used in meta-learning. These profiles consist of metalevel feature-value vectors which describe learning algorithms from the point of view of their representation and functionality, efficiency, robustness, and practicality. For certain characteristics related to functionality (attribute types, cost handling), algorithm specifications are given by expert users. Characteristics related to efficiency (learning and classification time and space) and robustness (scalability, resistance to missing values, noise, irrelevant and redundant attributes) can only be extrapolated from multiple executions of these algorithms over a wide variety of datasets [14].

3 Information-Theoretic Measures

Various papers [15, 16] have introduced the use of measures to characterize the data complexity and to relate such descriptions to classifier performance for two classes.

Information-theoretic measures are suitable to characterize discrete attributes. We used in our experiment: the entropy of the class label (ClassEnt), the entropy of all attributes (AttrEnt), the mutual information (entropy) of class and attributes (MutualInf), the joint entropy (JointEnt), the equivalent number of attributes (EquivAttr), proportion of the equivalent number of attributes (PropEquivAttr), the noise signal ratio (NoiseSR), the proportion of missing values (PropMV), proportion of number of examples with missing values (PropExMV) and a statistical measure that is the standard deviation of classes (StdDClass).

Let X be a random variable taking values x in X with distribution $p(x)=Pr[X=x]$. The entropy $H(X)$ of a random variable X (the label of the problem) is defined by:

$$H(X) = - \sum_{x \in X} p(x) \log_b p(x) \tag{1}$$

This is also denoted by $H(p)$ and measures the average uncertainty of a random variable X , b is the base of the logarithm used. Possible values of b are 2, e , and 10. The unit of the information entropy is bit for $b=2$. Then, in this case the entropy of the class label values belong to the interval $[0, \log_2 n]$, being n the number of the different values of the label. It means the maximum value of entropy of the class label for these three datasets is 1 since $n=2$.

The entropy of all attributes and the label (*Joint entropy*) measures the total entropy. It is the sum of the individual entropy of all variables appearing in the dataset.

The entropy of a collection of attributes is an average of the entropy over all the attributes, which is taken as a global measure of entropy of the attributes collectively [10].

Mutual information expresses the mutual dependency of the attributes and the label. It is the amount of information that can be obtained about the label by observing the attributes. The mutual information between X and Y is defined by:

$$MutualInf(X, Y) = \sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log \frac{p(y|x)}{p(y)} \tag{2}$$

The *equivalent number of attributes* estimates the number of attributes that are needed to determine the value of the label variable. If the number of relevant attributes that are provided by the dataset is larger than the value of this measure, there exists a good chance to learn a good classification algorithm [17]. The expression is:

$$EquivAttr = \frac{H(X)}{MutualInf(X, Y)} \tag{3}$$

The *proportion of the equivalent number of attributes* is calculated by the equivalent number of attributes divided by the number of attributes excluding the label.

Noise signal ratio is the amount of irrelevant information; large values of the ratio indicate that the dataset contains a large amount of irrelevant information that may be reduced [18]. It can be calculated by:

$$\text{NoiseSR} = \frac{\overline{H(X)} - \overline{\text{MutualInf}(X, Y)}}{\overline{\text{MutualInf}(X, Y)}} \quad (4)$$

The *proportion of missing values* is the ratio of the total missing values in the dataset between the number of all values in it.

The *proportion of number of examples with missing values* is the division of the total examples with missing values in the dataset between the number of examples.

4 Experimental Results

In this section the characteristics of breast-cancer-wisconsin (BCW) dataset are studied, because it is the only dataset from UCI Machine Learning Repository used in [4] where Stacking by MDT was significantly better than the other schemes used, even better than Stacking by MLR. We want to know the properties of this dataset in order to identify when Stacking by MDT is the best choice rather than Stacking by MLR. The objective is to search and to contribute with new elements in the Stacking meta-learning. We want also to find features that can be employed in another similar technique in order to improve meta-learning.

This dataset has 699 examples and 11 attributes (10 plus the label). There are 16 missing attribute values. The label for classification is composed by diagnosis classes. It has 2 classes (benign and malignant). The other attributes belong to the integer number set but they can be easily discretized because all of them take values in the interval [1,10], then we made the transformation. The names of them are: sample code number (it is not considered), clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. The class distribution is 458 examples (65.5%) of the “benign” class and 241 examples (34.5%) of the “malignant” class. We used Stacking with three base classifiers in our experiments: decision tree, nearest neighbor and Naïve Bayes. Furthermore, MLR was chosen to learn at the meta-level in one case and MJ4.8 in another.

The dimensionality of the BCW input space was reduced due to the removing of marginal adhesion, single epithelial cell size and bare nuclei attributes because there is a multivariate dependency among them. This strategy was also used in [19] to characterize the same dataset for the rule extraction. The missing values disappear when the reduction of the number of attributes is done.

We used the 10 measures described in the previous section and their values were obtained by METAL-MLEE (Machine Learning Experimentation Environment) [20], which is a software package developed for the METAL Project that helps in obtaining the meta-data for new datasets and different algorithms in performing meta-learning.

Two other datasets (hepatitis and vote) from UCI Machine Learning Repository for task classification were selected with only two class label, missing values and similar number of continuous attributes, as BCW dataset, in order to make comparisons.

Hepatitis dataset has continuous and discrete attributes. Vote dataset only has discrete attributes.

Hepatitis dataset did not show the same results than BCW dataset in [4]. Stacking by MDT for hepatitis dataset did not present significant difference in relation to the other seven schemes studied in that research. However, Stacking by MDT for vote dataset was significant better than the other schemes, excluding Stacking by MLR in the same experiment. Significant differences were not found between Stacking by MDT and Stacking by MLR for this dataset.

All datasets achieved high entropy of class label, although vote and BCW (in this order) highlighted because they reached values near to 1. The difference between them was not significant (0.033), see Table 1.

The highest value of the joint entropy was reached by BCW (over 2), while hepatitis and vote datasets obtained similar values.

The entropy of all attributes in BCW dataset is superior to the others. It is more than the double of the values of this metric obtained for hepatitis and vote datasets. It means the attributes of BCW dataset provide as average more information than the others.

In Table 1, we can also observe BCW dataset gets the best value of the useful information that each attribute has about the class label (mutual information).

The proportion of the equivalent number of attributes had a similar behavior in BCW with 9 attributes and vote, being the value for hepatitis dataset almost the double of the other datasets. When we executed the reduction to 6 attributes in BCW, this dataset obtained the highest value (45.3%) and hepatitis dataset got a value very close to 40%.

Table 1. Data characteristics of BCW and other two datasets using METAL-LEE

Measure	BCW with 9 attributes	BCW with 6 attributes	Hepatitis	Vote
ClassEnt	0.929	0.929	0.735	0.962
AttrEnt	2.251	2.299	1.052	0.990
MutualInf	0.512	0.507	0.103	0.304
JointEnt	2.668	2.721	1.683	1.649
EquivAttr	1.813	1.832	7.142	3.165
PropEquivAttr	0.201	0.453	0.376	0.198
NoiseSR	3.392	3.533	9.225	2.257
PropMV	0.002	0	0.059	0.053
PropExMV	0.023	0	0.484	0.467
StdDClass	0.155	0.155	0.293	0.114

Hepatitis dataset attained the greatest value of the noise signal ratio. There is a significant difference with the other two datasets: 6.968 with regard to vote and 5.692 in relation to BCW for 6 attributes.

If we analyze the results of BCW with 9 attributes, it is possible to see in Table 1 that the less percentage of missing values regarding to all values in the dataset was reached by BCW with 0.2%, it is very low, while hepatitis and vote datasets have values close to 6% and 5%, respectively. There is a very significant difference for the proportion of number of examples with missing values between BCW and the other

datasets. BCW only has a 2.3% of examples with missing values and the rest has almost the 50%.

However, BCW with 6 attributes has not missing values, and then all the measures in relation to missing values got values 0.

After the analysis of the measure calculations it is possible to infer there is most probability datasets with few or without missing values achieve better results in classification with Stacking by MDT than Stacking by MLR. Together with this, it is necessary the entropy of the label is very close to the maximum value, the average of the entropy of all attributes, the joint entropy and mutual information obtain high value, and the equivalent number of attributes and noise signal ratio get very low values.

We wanted also to know the ranking of the three algorithms used as base classifiers to predict the label of BCW dataset and therefore, to obtain the meta-attributes.

WekaMetal [21] is a meta-learning extension to the data mining package Weka [22]. It has been used in order to obtain the rankings of IBk (nearest neighbor), decision tree (J48) and Naïve Bayes classifiers for BCW dataset. The ranker selected for this study was Adjusted Ratio of Ratios (ARR), based on expected accuracy and time performance; see the results in Table 2.

Table 2. Ranking of three algorithms for BCW dataset using WekaMetal

Ranking	Algorithm	Measure multi-criteria
1	Naïve Bayes	2.6078
2	J48	0.6659
3	IBk	0.6339

In our experiments, Stacking by MDT and Stacking by MLR had the same accuracy value using BCW with Weka (Fig. 1), and the use of computational resources of the first multiclassifier is almost two times that the second one. In all experiments we carried out ten-fold cross-validation. In Fig. 1, it is possible to show Naïve Bayes is, of course, faster than the two Stacking schemes, and the Naïve Bayes accuracy value

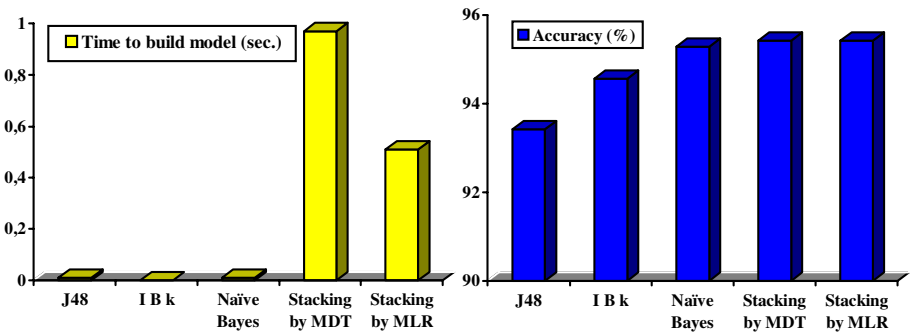


Fig. 1. Times to build models and classification accuracies for BCW dataset

is not significantly lower than the multiclassifiers. Then, the use of Stacking schemes could not be justified because there is a base classifier which is faster and obtains a similar accuracy value than the Stacking schemes used.

5 Conclusions

In our experiment, datasets with discrete attributes exclusively, in classification by Stacking MDT achieved similar behavior when compared to Stacking by MLR.

According the results obtained, it is possible to point out Stacking by MDT and Stacking by MLR have similar classification accuracy in datasets with 2 label values, when the entropy of the label is very close to the maximum value, the average of the entropy of all attributes, the joint entropy and mutual information values are high. Moreover, the equivalent number of attributes must be very low and also the noise signal ratio.

Important information-theoretic measures in this experiment were those related to the missing values. Very low percentages of missing values and a small number of examples with missing values took place only in BCW dataset.

Finally, for future work we are planning a study with other metrics based on information theory and landmarking approach, which can also be applied to data characterization. On the other hand, the incorporation of other meta-attributes to the meta-level could also be useful.

Acknowledgments

This paper has been done in the framework of the research project SAD64A07, supported by the Spanish *Junta de Castilla y León*.

References

1. Fan, D., Chan, P., Stolfo, S.: A comparative evaluation of Combiner and Stacked Generalization. In: Proceedings of AAAI 1996 Workshop on Integrating Multiple Learned Models, pp. 40–46 (1996)
2. Chan, P., Stolfo, S.: On the accuracy of Meta-learning for Scalable Data Mining. *Journal of Intelligent Information Systems* 8(1), 5–28 (1997)
3. Peng, Y., Flach, P., Brazdil, P., Soares, C.: Decision Tree-Based Data Characterization for Meta-Learning. In: Proceedings of the Second International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (IDDM 2002), pp. 111–122. Helsinki University Printing House (2002)
4. Zenko, B., Todorovski, L., Dzeroski, S.: A comparison of stacking with meta decision trees to other combining methods. In: Proceedings of the Fourth International Multi-Conference Information Society, vol. A, pp. 144–147. Jozef Stefan Institute, Ljubljana (2001)
5. Wolpert, D.: Stacked generalization. *Neural Networks* 5, 241–259 (1992)

6. Peng, Y., Flach, P., Soares, C., Brazdil, P.: Improved Dataset Characterisation for Meta-learning. In: Lange, S., Satoh, K., Smith, C.H. (eds.) DS 2002. LNCS, vol. 2534, pp. 141–152. Springer, Heidelberg (2002)
7. Rendell, L., Seshu, R., Tcheng, D.: Layered Concept Learning and Dynamically Variable Bias Management. In: Proceedings of the 10th International Joint Conference on Artificial Intelligence, pp. 308–314 (1987)
8. Vilalta, R., Giraud-Carrier, C., Brazdil, P.: Meta-Learning: Concepts and Techniques. In: Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers. Springer, Heidelberg (2005)
9. Köpf, C., Taylor, C., Keller, J.: Meta-analysis: from data characterisation for meta-learning to meta-regression. In: Proceedings of the PKDD-2000 Workshop on Data Mining, Decision Support, Meta-Learning and ILP (2000)
10. Michie, D., Spiegelhalter, D., Taylor, C. (eds.): Machine Learning, Neural and Statistical Classification, volume: Artificial Intelligence. Ellis Horwood (1994)
11. Kalousis, A., Hilario, M.: Model selection via meta-learning: a comparative study. In: Proceedings of the 12th International IEEE Conference on Tools with AI. IEEE Press, Los Alamitos (2000)
12. Bensusan, H., Giraud-Carrier, C., Kennedy, C.: A higher-order approach to meta-learning. In: Proceedings of the ECMLS 2000 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination, pp. 109–117 (2000)
13. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms. In: Proceedings of the 17th International Conference on Machine Learning, pp. 743–750 (2000)
14. Hilario, M., Kalousis, A.: Building Algorithm Profiles for Prior Model Selection in Knowledge Discovery Systems. In: Proceedings of the IEEE SMC 1999, International Conference on Systems, Man and Cybernetics, Tokyo (October 1999)
15. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 289–300 (2002)
16. Bernardo, E., Ho, T.K.: On classifier domain of competence. In: Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, pp. 136–139 (2004)
17. Studer, R., Staab, H.A., Mädche, A., Jetter, U.: Vorlesung Knowledge Discovery. Data Characterization Tool (DCT). Institute AIFB, Universität Karlsruhe (1999)
18. Köpf, C.: Meta-learning: Strategies, Implementations, and Evaluations for Algorithm Selection. Dissertations in Database and Information Systems-Infix, vol. 91. IOS Press, Amsterdam (2006)
19. Taha, I., Gosh, J.: Characterization of the Wisconsin breast cancer database using a hybrid symbolic-connectionist system. Technical Report UT-CVI-TR-97007. The Computer and Vision Research Center, University of Texas, Austin (1996)
20. Petrak, J.: The METAL Machine Learning Experimentation Environment V3.0 (METAL-MLEE) Manual - Version 3.0. Austrian Research Institute for Artificial Intelligence (October 2002), <http://www.metal-kdd.org/>
21. Farrand, J.: WekaMetal. University of Bristol (February 2002), <http://www.cs.bris.ac.uk/~farrand/wekametal>
22. University of Waikato. Weka (1999), <http://www.cs.waikato.ac.nz/~ml/weka/index.html>

An EM-Based Piecewise Linear Regression Algorithm

Sebastian Nusser^{1,2}, Clemens Otte¹, and Werner Hauptmann¹

¹ Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81730 Munich, Germany
{sebastian.nusser.ext, clemens.otte, werner.hauptmann}@siemens.com

² School of Computer Science, Otto-von-Guericke-University of Magdeburg,
Universitätsplatz 2, 39106 Magdeburg, Germany

Abstract. This contribution describes an EM-like piecewise linear regression algorithm that uses information about the target variable to determine a meaningful partitioning of the input space. The main goal of this approach is to incorporate information about the target variable in the prototype selection process of a piecewise regression approach. Furthermore, the proposed approach is designed to provide an interpretable solution by restricting the dimensionality of the local regression models. We will show that our approach achieves a similar predictive performance on benchmark problems compared to standard regression methods – while the model complexity of our approach is reduced.

1 Introduction

The quality of a piecewise regression algorithm depends on the quality of its partitioning of the input space. Common piecewise regression approaches, for instance the Local Linear Map (LLM) of [1], use only information about the input space for partitioning the data. The target variable is usually ignored while clustering the input space. This strategy becomes inefficient in situations where the data points cannot be distinguished within the input space but where a meaningful partitioning can still be obtained by additionally considering the target variable. Furthermore, regions of high data density in real-world application problems usually correspond to operating points of the system. Such regions are not necessarily appropriate to determine a satisfactory partitioning of the input space, because the underlying function of the system might not change within the operating points but in regions with low data density.

The objective of our work is to incorporate information about the target variable into the process of choosing the best prototypes of a piecewise linear regression model. Furthermore, we are interested in providing a suitable trade-off between an interpretable solution and a solution that provides a high predictive accuracy. Symbolic models like regression trees or rule systems allow a good impression about the “big picture” of a given problem. However, they sacrifice some details and usually will not show such a high accuracy as sub-symbolic solutions (e.g. neural networks). There are many application domains where finding a good trade-off between interpretability and predictive performance of the learned solution is crucial. For instance, in the field of safety-related applications, it is necessary to provide transparent solutions that can be validated by domain experts. “Black box” approaches, like artificial

neural networks, are regarded with suspiciousness – even if they show a very high accuracy on the available data – because it is not feasible to prove that they will show a good performance on all possible input combinations. Another example is the field of bioinformatics, where it is necessary to provide transparent solutions to get an impression how the biological mechanisms are working. The problem that arises in both domains is that the amount of independent samples is often not large enough to apply statistical risk estimation methods and extensive evaluations. This contribution proposes an EM-like algorithm to determine an interpretable solution based on low-dimensional local models that achieves a similar performance compared to high-dimensional regression methods. Different approaches were already proposed to tackle the problem of partitioning the data by incorporating the target function: for instance, in [2, 3] the data is clustered based on the model parameters of local regression models and in [4] a combined distance function is proposed for clustering, where the distance within the input space and the error of the local models is incorporated.

Sect. 2 presents our EM-based piecewise linear regression approach and discusses two extensions of this approach: the restriction of the dimensionality of the local models and the possibility to prune the number of clusters. Experiments performed on artificial and on benchmark data sets are discussed in Sect. 3. Sect. 4 concludes.

2 The LinEM-Algorithm

The algorithm described in the following is an adaptation of the Expectation Maximization (EM) algorithm [5]. The basic idea of our approach is to incorporate information about the target variable while dividing the input space into different regions, each described by a prototype and a corresponding local regression model. The regression models are trained for regions of the input space. Such regions are represented by cluster prototypes. For each cluster region a linear regression model is learned. Comparable to the EM algorithm, this approach consists of two main steps: first, the local regression models are trained according to the cluster prototypes, and, second, the cluster prototypes are updated according to the predictive performance of the linear regression models. The data points are assigned to the linear regression model with the best predictive performance. Then the cluster prototypes are updated to the mean of all samples that are assigned to the corresponding linear regression model.

Learning the Local Models. Given an n -dimensional input space: $V^n = X_1 \times X_2 \times \dots \times X_n$, where $X_l \subseteq \mathfrak{R}$. The target variable $Y \subseteq \mathfrak{R}$ is determined by the (unknown) function: $f : X_1 \times X_2 \times \dots \times X_n \rightarrow Y$. The learning task is to determine a function estimate $f^* : V^n \rightarrow Y$ of f given an observed data set: $D = \{(v_1, y_1), \dots, (v_m, y_m)\} \subset V^n \times Y$.

The LinEM algorithm is given in Algorithm 1. It consists of two main steps: in the first step local regression models are determined according to a given cluster assignment, and in the second step the given data points are assigned to the clusters where the predictive error of the corresponding regression model is minimal. Our algorithm requires a predefined number of clusters $k < m$ that are assumed to represent the data. Each sample data point is assigned to one single cluster. This assignment is defined by the mapping:

$$C(i) = j, \text{ with } 1 \leq i \leq m, 1 \leq j \leq k, \quad (1)$$

Algorithm 1. The LinEM-Algorithm

input: data set D , number of clusters k output: cluster prototypes \mathbf{p}_j , local models g_j 1: Choose random cluster assignment $C(i) = j$

2: repeat

3: Learn k local models according cluster indicies, cf. Eq. 34: Update cluster indicies according to model error: $C(i) = \operatorname{argmin}_{j=1,\dots,k} |y_i - g_j(\mathbf{v}_i)|$

5: Compute cluster prototypes, cf. Eq. 4

6: Update cluster indicies according to prototypes, cf. Eq. 5

7: until termination criterion fulfilled (e.g. maximum of iterations)

which assigns the i -th observation to the j -th cluster. Each local model consists of a tuple of a cluster prototype \mathbf{p}_j and a local regression model:

$$g_j : V^n \rightarrow Y. \quad (2)$$

The local regression model of the j -th cluster is trained on all data points that are assigned by the mapping C to the j -th cluster:

$$g_j(\mathbf{v}_i) = y_i, \text{ where } C(i) = j. \quad (3)$$

The j -th cluster prototype is determined by:

$$\mathbf{p}_j = \frac{1}{m_j} \sum_{C(i)=j} \mathbf{v}_i, \quad (4)$$

where $m_j = \sum_{i=1}^m I(C(i) = j)$ is the number of samples that belong to the j -th cluster

$$\text{and } I(A) = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{if } A \text{ is false} \end{cases}.$$

The LinEM algorithm performs two alternating updates of the mapping C . Firstly, in line 4 the mapping is updated according to the minimal predictive error of the current local regression models g_j . Then the new prototypes are determined according to Eq. 4. Based on the new prototypes and in order to achieve a crisp partitioning of the input space the second update of the mapping C is performed in line 6 of Algorithm 1. The data points are assigned to the cluster with the closest cluster prototype:

$$C(i) = \operatorname{argmin}_{j=1,\dots,k} (\operatorname{dist}(\mathbf{v}_i, \mathbf{p}_j)), \quad (5)$$

where dist is the squared Euclidian distance, $\operatorname{dist}(\mathbf{v}, \mathbf{u}) = \|\mathbf{v} - \mathbf{u}\|^2$.

The corresponding objective function of the LinEM algorithm is:

$$J = \sum_{i=1}^m \sum_{j=1}^k I(C(i) = j) |y_i - g_j(\mathbf{v}_i)|. \quad (6)$$

Applying the Local Models. In the case of a crisp cluster assignment, the prediction is straightforward:

$$f^*(\mathbf{v}) = g_{j^*}(\mathbf{v}), \quad (7)$$

where $j^* = \operatorname{argmin}_{j=1,\dots,k} \{\operatorname{dist}(\mathbf{v}, \mathbf{p}_j)\}$. If there are certain smoothness assumptions about the target function, one can also apply a smoothing function on the predictions of the

local models to determine the global model prediction. As an example, one can use the softmax function:

$$w_{i,j} = \frac{\exp(-dist(\mathbf{v}_i, \mathbf{p}_j))}{\sum_{j=1}^k \exp(-dist(\mathbf{v}_i, \mathbf{p}_j))} \tag{8}$$

to compute weights of the local models. Thus, the final model prediction is computed as the weighted sum:

$$f^*(\mathbf{v}_i) = \sum_{j=1}^k w_{i,j} \cdot g_j(\mathbf{v}_i) . \tag{9}$$

Restricting the Dimensionality of the Local Models. Adopting an idea used in [6], instead of using all input dimensions in each local model, projections of the high-dimensional input space can be used to increase the interpretability of the learned local models. The projection π maps the n -dimensional input space V^n to an arbitrary subspace of V^n . This mapping is determined by a given index set $\beta \subset \{1, \dots, n\}$. The index set defines the dimensions of V^n that will be included in the subspace V_β . The projection π on the input space V^n given the index set β is defined as:

$$\pi_\beta(V^n) = V_\beta = \prod_{l \in \beta} X_l , \tag{10}$$

Thus, the j -th local model can be re-defined as:

$$g_j : \pi_{\beta[j]}(V^n) \rightarrow Y , \tag{11}$$

where β_j denotes the index set of the subspace where the predictive error of the local model g_j is minimal. The best projection can be determined, for instance, by a wrapper model selection method [7]. This ensures that the best possible sub-space is used.

Cluster Pruning. The LinEM algorithm allows a cluster pruning strategy, where clusters can be removed which are ill-posed, that is the number of data points that belong to the current cluster is too small to solve the regression problem. Furthermore clusters with similar regression models can be merged. These heuristics facilitate our algorithm to deal with problems where the optimal number of clusters is unknown.

3 Experimental Results

The LinEM algorithm is compared with the Local Linear Map (LLM) from [1], a standard linear regression method (`robustfit` in Matlab), and a regression tree (`treerefit` in Matlab) on artificial and on common benchmark data sets. We distinguish the crisp and a softmax (c.f. Eq. 8) variant of the LinEM algorithm. The predictive error is estimated by: $err = 1/m \sum_{i=1}^m |y_i - f^*(\mathbf{v}_i)|$. The local models of the LinEM algorithm are estimated by the `robustfit` method and are restricted to two dimensions in all experiments. The best two-dimensional input combination is determined by a wrapper method for feature selection [7] that performs an exhaustive search through all pairwise combinations.

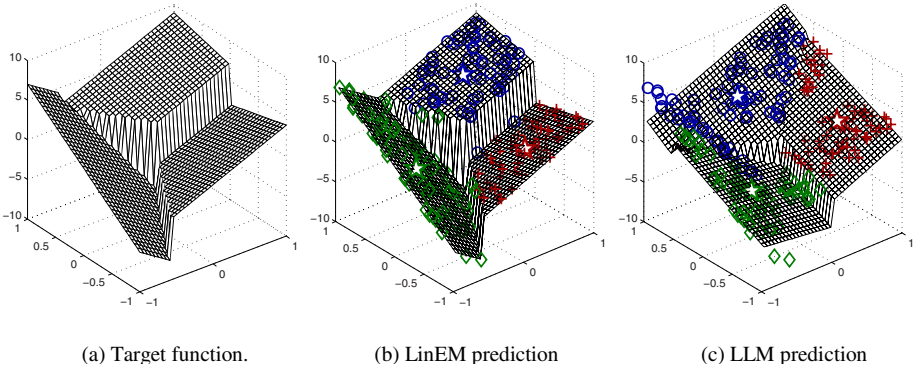


Fig. 1. Artificial1 data set. The original data points are labeled according their cluster assignment. The cluster prototypes are depicted as stars.

Artificial1 Data Set. We used the following function from [3] as target function:

$$f_1(x_1, x_2) = \begin{cases} 3 + 4x_1 + 2x_2 & \text{if } A_1 \wedge A_3 \\ -5 - 6x_1 + 6x_2 & \text{if } \neg A_1 \wedge \neg A_2 \\ -2 + 4x_1 - 2x_2 & \text{if } A_2 \wedge \neg A_3 \end{cases}$$

where $A_1 : 0.5x_1 + 0.29x_2 \geq 0$, $A_2 : 0.5x_1 - 0.29x_2 \geq 0$, and $A_3 : x_2 \geq 0$. This target function is depicted in Fig. 1(a). 300 samples are drawn uniformly from $V^n = [-1, 1] \times [-1, 1]$ and y is determined as $y = f_1(x_1, x_2) + \varepsilon$, where $\varepsilon \sim N(0, 0.1)$. Note that, in this setting, it is impossible to determine appropriate cluster prototypes without regarding the target value.

The LinEM algorithm yields the following function estimate:

$$f_1^*(x_1, x_2) = \begin{cases} 3.020 + 3.965x_1 + 1.956x_2 & \text{if } \mathbf{p}_1 \\ -5.014 - 6.015x_1 + 6.028x_2 & \text{if } \mathbf{p}_2 \\ -1.999 + 3.998x_1 - 2.028x_2 & \text{if } \mathbf{p}_3 \end{cases},$$

where $\mathbf{p}_1 = (0.466, 0.476)^T$, $\mathbf{p}_2 = (-0.583, 0.122)^T$, and $\mathbf{p}_3 = (0.426, -0.564)^T$. This function is almost equivalent to the original target function. The function estimate and the corresponding cluster assignments are illustrated in Fig. 1(b). Fig. 1(c) illustrates the function obtained by applying the LLM algorithm on the same data set. The LLM was unable to determine appropriate cluster prototypes because the prototypes are determined only on the input space. The LLM assumes a wrong partitioning of the input space and, thus, determines inappropriate local regression models.

Fig. 3(a) illustrates the influence of the number of predefined clusters: with one single cluster the LinEM is equivalent to the `robustfit` solution. The only approach that outperforms the regression tree model (`treefit`) is the crisp LinEM approach. Due to the cluster pruning of the LinEM approach, the error estimate remains almost constant for initializations of the number of clusters $k \geq 3$ – due to the fact that most of the LinEM models are pruned to three different clusters.

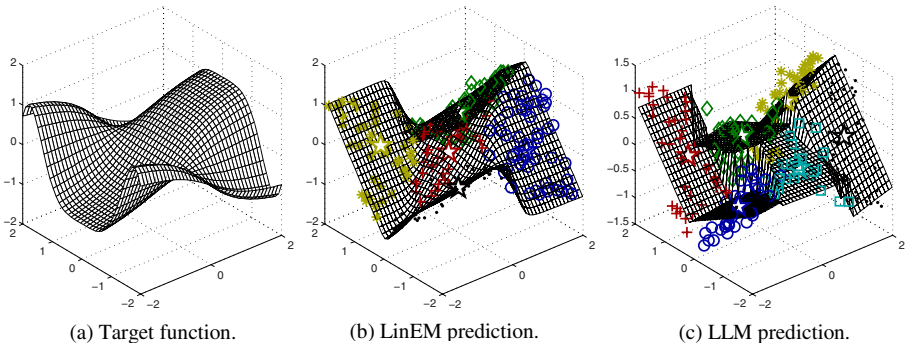


Fig. 2. Artificial2 data set. The initial number of clusters is set to six for both regression methods, LinEM and LLM.

Artificial2 Data Set. This data set is used in [4]. The target function, which is illustrated in Fig. 2(a), is $f_2(x_1, x_2) = \arctan(x_1)\cos(x_2^2)$. 300 data points are drawn uniformly from $V^n = [-2, 2] \times [-2, 2]$ and the value of the target y is determined by $y = f_2(x_1, x_2) + \varepsilon$, where $\varepsilon \sim N(0, 0.1)$. In Fig. 2(b) and Fig. 2(c) the model predictions and the corresponding cluster assignments of the data points of the LinEM and LLM solution are shown. Fig. 3(b) illustrates the influence of the initial number of cluster on the predictive error. The performance of the LinEM approach is superior to the performance of the LLM model due to a more intuitive placement of the prototypes. For cluster initialization larger than six the number of clusters is pruned to six clusters.

Benchmark data sets. All data sets described in the following are randomly divided into a training data set (80% of the data) and a testing data set (20% of the data). 20 different pairs of training and testing data are generated for each data set. All input dimensions are normalized to the interval $[-1, 1]$ in order to rule out scaling effects.

Auto MPG Data Set. This data set can be obtained from <http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html>. The task of this data set is to predict the fuel consumption in miles per gallon (MPG) of different cars. The following five attributes of the original data set are used as input dimensions: 'acceleration', 'displacement', 'horsepower', 'model-year', 'weight'. The data set consists of 398 instances. Six instances with missing values are ignored within the experiments.

Body Fat Data Set. This data set can be obtained from <http://lib.stat.cmu.edu/datasets/>. It includes 13 attributes and 252 instances. The task is to estimate the percentage of body fat based on different body circumference measurements.

Ozone Data Set. The Ozone data set and the Prostate data set can be obtained from <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>. The Ozone data set is used to estimate the daily ozone concentration based on three attributes: wind speed, daily maximum temperature, and solar radiation. This data set includes 111 instances.

Prostate Data Set. The task is to estimate the level of a prostate specific antigen based on a eight clinical measurements. The data set consists of 97 data points.

Discussion. Our experiments are summarized in Fig. 3 and Tab. 1. Especially for a small number of local models ($k \leq 3$) the LinEM approach (that is restricted to two-dimensional local models within the experiments) achieves a better or at least similar performance compared to the LLM approach that always regards the complete input space within its local models. The LinEM approach places the prototypes to regions which are relevant to build local regression models with a small predictive error instead of placing the prototypes to regions with a high data density. The LLM approach can compensate this weakness for a higher number of prototypes – however,

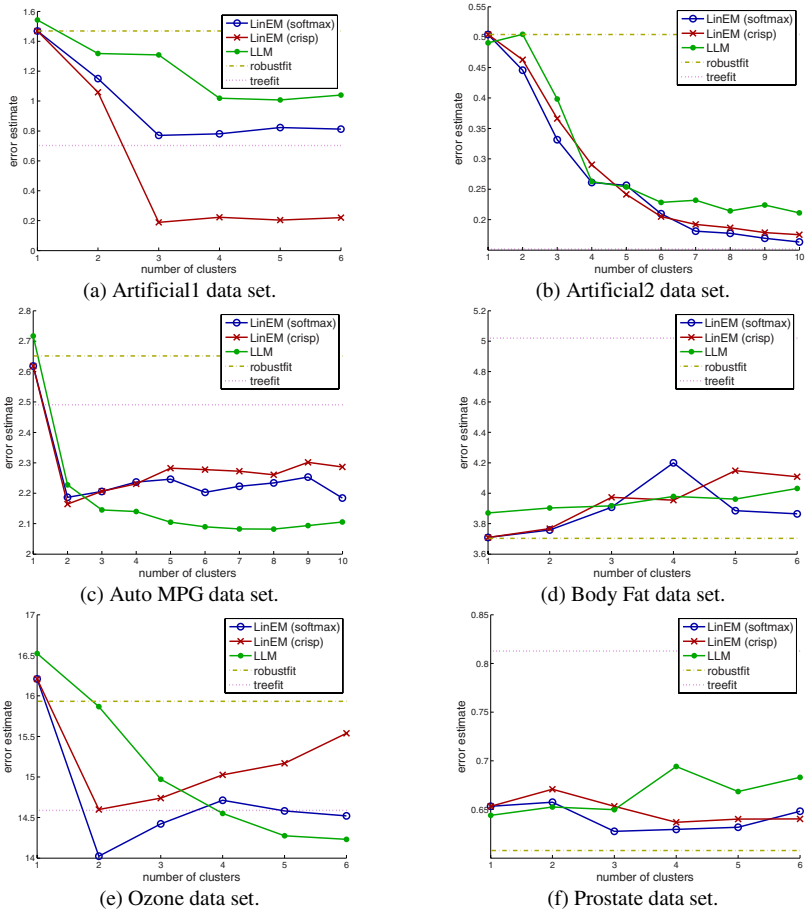


Fig. 3. Artificial and benchmark data sets: Influence of the initialization of the initial number of clusters on the error rate. The error is estimated on 20% of the data where 80% of the data is used for training the model. The results are averaged over 20 different runs for each data set. The dimensionality of the local models of the LinEM approach is set to two in all experiments.

Table 1. Model complexity for each regression method. This table shows the details of the models of Fig. 3 where the predictive error on the testing data is minimal. #M denotes the number of local models or the number of decision nodes in the (unpruned) regression tree. For the LinEM the number of initial clusters (i.e. before cluster pruning) is given in parentheses. #D denotes the dimensionality of each (local) model or decision nodes within the regression tree.

Data set name	Regression method	#M	#D	Predictive error (std)	
				training	testing
Artificial1	LinEM (softmax)	3 (of 3)	2	0.79 (0.02)	0.77 (0.10)
	LinEM (crisp)	3 (of 3)	2	0.20 (0.03)	0.19 (0.09)
	LLM	5	2	0.95 (0.06)	1.01 (0.11)
	robustfit	1	2	1.48 (0.04)	1.47 (0.15)
	treefit	41	1	0.34 (0.02)	0.70 (0.10)
Artificial2	LinEM (softmax)	6 (of 10)	2	0.15 (0.01)	0.16 (0.02)
	LinEM (crisp)	6 (of 9)	2	0.16 (0.03)	0.18 (0.02)
	LLM	8	2	0.20 (0.01)	0.21 (0.03)
	robustfit	1	2	0.50 (0.01)	0.50 (0.05)
	treefit	46	1	0.08 (0.01)	0.15 (0.01)
Auto MPG	LinEM (softmax)	7 (of 10)	2	1.98 (0.07)	2.18 (0.21)
	LinEM (crisp)	2 (of 2)	2	2.07 (0.06)	2.16 (0.23)
	LLM	8	5	1.91 (0.05)	2.08 (0.20)
	robustfit	1	5	2.56 (0.06)	2.65 (0.25)
	treefit	66	1	1.04 (0.08)	2.49 (0.28)
Body Fat	LinEM (softmax)	1 (of 1)	2	3.58 (0.10)	3.71 (0.39)
	LinEM (crisp)	1 (of 1)	2	3.58 (0.10)	3.71 (0.39)
	LLM	1	13	3.68 (0.09)	3.87 (0.40)
	robustfit	1	13	3.39 (0.10)	3.70 (0.41)
	treefit	39	1	1.76 (0.12)	5.02 (0.49)
Ozone	LinEM (softmax)	2 (of 2)	2	12.82 (1.20)	14.02 (2.97)
	LinEM (crisp)	2 (of 2)	2	12.72 (0.81)	14.60 (3.31)
	LLM	6	3	11.83 (0.82)	14.23 (1.87)
	robustfit	1	3	15.15 (0.58)	15.93 (2.43)
	treefit	20	1	6.97 (0.76)	14.59 (2.99)
Prostate	LinEM (softmax)	3 (of 5)	2	0.48 (0.03)	0.63 (0.10)
	LinEM (crisp)	3 (of 6)	2	0.48 (0.03)	0.64 (0.10)
	LLM	1	8	0.58 (0.03)	0.64 (0.12)
	robustfit	1	8	0.50 (0.03)	0.61 (0.11)
	treefit	15	1	0.35 (0.04)	0.81 (0.09)

the larger the number of prototypes the more difficult the interpretation of the solution will be. For the benchmark data sets, the model complexity of the LinEM approach is smaller as for all other regression approaches – the regression trees become quite large, the linear regression model uses the complete input space, and the LLM model uses always k prototypes with n -dimensional models, while the cluster pruning strategy and the restriction to two-dimensional local models of the LinEM approach facilitates the interpretability. If one compares the best results of all regression methods (cf. Tab. 1), the LinEM method provides the best performance on two data sets – on the remaining data sets the performance is only slightly worse compared to the best approaches.

4 Conclusions

Incorporating information about the target variable into the prototype selection process of a piecewise regression can greatly improve the performance on problems where the input space does not provide sufficient information for partitioning the data. The proposed LinEM algorithm provides a good trade-off between the interpretability and the predictive performance of a learned solution. It achieves a similar performance

compared to established regression methods, while the model's complexity can be reduced. It is capable of reconstructing piecewise linear functions from a given data set. Furthermore, the approach facilitates a cluster pruning strategy which has been proven useful in situations where no apriori information about the number of clusters are available.

References

1. Ritter, H.: Learning with the self-organizing map. In: Kohonen, T., Mäkisara, K., Simula, O., Kangas, J. (eds.) *Artificial Neural Networks*, pp. 379–384. Elsevier, Amsterdam (1991)
2. Hathaway, R.J., Bezdek, J.C.: Switching regression models and fuzzy clustering. *IEEE Transactions on Fuzzy Systems* 1(3), 195–204 (1993)
3. Ferrari-Trecate, G., Muselli, M.: A new learning method for piecewise linear regression. In: *Proc. of International Conference on Artificial Neural Networks*, pp. 444–449. Springer, Heidelberg (2002)
4. Höppner, F., Klawonn, F.: Improved fuzzy partitions for fuzzy regression models. *International Journal of Approximate Reasoning* 32(2-3), 85–102 (2003)
5. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B* 39(1), 1–38 (1977)
6. Nusser, S., Otte, C., Hauptmann, W.: Learning binary classifiers for applications in safety-related domains. In: *Proc. of 17th Workshop Computational Intelligence*, Universitätsverlag Karlsruhe, pp. 139–151 (2007)
7. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial Intelligence* 97(1-2), 273–324 (1997)

On the Use of Linear Cellular Automata for the Synthesis of Cryptographic Sequences

A. Fúster-Sabater^{1,*}, P. Caballero-Gil², and O. Delgado¹

¹ Instituto de Física Aplicada, C.S.I.C., Serrano 144, 28006 Madrid, Spain
{amparo, oscar.delgado}@iec.csic.es

² DEIOC, University of La Laguna, 38271 La Laguna, Tenerife, Spain
pcaballe@ull.es

Abstract. This work shows that a class of cryptographic sequences, the so-called interleaved sequences, can be generated by means of linear hybrid cellular automata. More precisely, linear multiplicative polynomial cellular automata generate all the components of this family of interleaved sequences. As an illustrative example, the linearization procedure of the self-shrinking generator is described. In this way, popular nonlinear sequence generators with cryptographic application are linearized in terms of simple cellular automata.

Keywords: interleaved sequence, linear cellular automata, self-shrinking generator, cryptography.

1 Introduction

This work deals with cellular automata and pseudorandom binary sequences. On the one hand, cellular automata are used to simulate complex but stable self-organizing systems and, consequently, are considered one of the main areas of computational intelligence, closely related to hybrid intelligent systems. On the other hand, pseudorandom binary sequences with high linear complexity and low correlation values are widely used in communication systems, error correcting codes and cryptography (stream ciphers).

In the literature we can find a broad family of pseudorandom sequences, the so-called interleaved sequences [1], with the common characteristic that each sequence can be written in terms of a unique shifted PN -sequence. In fact, a PN -sequence is the output sequence of a Linear Feedback Shift Register (LFSR) with primitive characteristic polynomial. For a survey of PN -sequences, shift equivalence and LFSRs, the interested reader is referred to [2].

Interleaved sequences are obtained as output sequences from nonlinear generators based on LFSRs [3]. These sequences are currently generated:

1. By a LFSR controlled by another LFSR (which may be the same one) e.g. multiplexed sequences [4], clock-controlled sequences [5], cascaded sequences [6], shrinking generator sequences [7] etc.

* Corresponding author.

2. By one or more than one LFSR and a feed-forward nonlinear function e.g. Gold-sequence family, Kasami (small and large set) sequence families, GMW sequences, Klapper sequences, No sequences etc. [1].

In brief, a large number of popular cryptographic sequences are included in the class of interleaved sequences. In this work, an easy method of generating interleaved sequences by means of Cellular Automata (CA) is presented. More precisely, the class of linear multiplicative polynomial CA is used in the generation process. In this way, complex nonlinear sequence generators of cryptographic application are expressed in terms of simple linear cellular structures.

2 Interleaved Sequences and Linear Hybrid Cellular Automata

The two basic structures we are dealing with (interleaved sequences and linear multiplicative polynomial CA) are introduced in the following subsections.

2.1 Fundamentals of the Interleaved Sequences

Let $s = \{s(0), s(1), \dots\}$ be a binary sequence where $s(k) \in GF(2)$ and $k \geq 0$. The characteristic polynomial of the sequence s is denoted by:

$$f(x) = x^r + \sum_{j=1}^r c_j x^{r-j} \in GF(2)[x]. \tag{1}$$

It represents the linear recurrence relationship [2] of such a sequence. That is, each element of s can be written as a linear combination of the r previous elements,

$$s(k+r) = \sum_{j=1}^r c_j s(k+r-j) \quad k \geq 0. \tag{2}$$

In this case, s is said to be generated by $f(x)$. The polynomial of the lowest degree in the set of characteristic polynomials of s over $GF(2)$ is called the minimal polynomial of such a sequence.

Definition 1. Let $f(x)$ be a polynomial over $GF(2)$ of degree r and let m be a positive integer. For any sequence $u = \{u(k)\}$ over $GF(2)$, write $k = im + j$ ($i \geq 0, j = 0, \dots, m-1$). If $u_j = \{u(im + j)_{i \geq 0}\}$ is generated by $f(x)$ for all j , then u is called an interleaved sequence of size m associated with $f(x)$.

We can write $u = (u_0, u_1, \dots, u_{m-1})$ where the u_j 's are the sequential components of u . By definition, every u_j is an m -decimation of the sequence u , that is one out of m bits has been taken. As the LFSRs of cryptographic application have primitive characteristic polynomials, in the sequel $f(x)$ will be a primitive polynomial [8] of degree r . In this case, the sequence u is called a primitive interleaved sequence and the u_j 's are the same PN -sequence generated by $f(x)$. Primitive interleaved sequences are characterized by (see [1]):

1. The period of each u_j is $T = 2^r - 1$, thus the period of the interleaved sequence u will be $T_u = m(2^r - 1)$.
2. The minimal polynomial $h(x)$ of u is a divisor of $f(x)^m$ so that the linear complexity of the interleaved sequence (the degree of its minimal polynomial) is upper bound by $LC \leq rm$. In fact, $h(x)$ is the product of $f(x)$ a number of times.

Table 1 shows the interleaved sequence u associated with the 3-degree primitive polynomial $f(x) = x^3 + x + 1$ and size $m = 4$. The period $T_u = m(2^r - 1) = 28$, $h(x) = (x^3 + x + 1)^4$ and $LC(u) = 12$. By rows, the interleaved sequences is $u = \{1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, \dots, 1, 1, 0, 0\}$. By columns, the sequence is made out of four shifted versions of the PN-sequence generated by $f(x)$.

Table 1. Interleaved sequence u with 4 shifted versions of the same PN-sequence

u_0	u_1	u_2	u_3
1	1	1	1
1	0	1	0
0	0	1	1
0	1	0	1
1	0	0	1
0	1	1	0
1	1	0	0

2.2 Linear Multiplicative Polynomial Cellular Automata

CA are particular forms of finite state machines defined as uniform arrays of identical cells in an n -dimensional space [9]. The cells X_i change their states (contents) synchronously at discrete time instants. The next state of each cell depends on the current states of the neighboring cells according to an update state transition rule. CA have been proposed in the bibliography [10] as an alternative to LFSRs because every sequence generated by a LFSR can be also obtained from one-dimensional CA. Moreover, it has been proven [11] that linear one-dimensional CA are isomorphic to LFSRs. In this paper we will show that nonlinear generators of interleaved sequences may also be expressed in terms of certain linear CA.

In particular, linear multiplicative polynomial CA of binary contents are discrete dynamical systems characterized by [12]:

1. Their underlying topology is one-dimensional, that is they can be represented by a succession of L cells X_i ($i = 1, \dots, L$) where L is a positive integer that denotes the length of the automaton. The state of the i -th cell at instant n , notated x_i^n , takes binary values $x_i^n \in GF(2)$.
2. They are linear cellular automata as the local transition rule for each cell is a linear mapping Φ_i defined such as follows:

$$x_i^{n+1} = \Phi_i(x_{i-k}^n, \dots, x_i^n, \dots, x_{i+k}^n) \quad (i = 1, \dots, L)$$

where k is a positive integer that denotes the size of the neighborhood.

3. Each one of these cellular automata is represented by an $L \times L$ transition matrix M over $GF(2)$. The characteristic polynomial of such matrices is of the form:

$$P_M(x) = P(x)^p, \tag{3}$$

where $P(x)$ is a primitive polynomial over $GF(2)$ and p is a positive integer.

3 Realization of Linear Binary Multiplicative Polynomial CA

In the previous sub-section, general characteristics of the binary multiplicative polynomial CA have been described. Now the particular form of such automata is analyzed. In order to realize this class of CA, transition rules with $k=3$ will be considered. In fact, only hybrid CA with rules 90 and 150 (see [9] for notation) are to be used. Such rules are defined as:

Rule 90	Rule 150
$x_i^{n+1} = x_{i-1}^n \oplus x_{i+1}^n$	$x_i^{n+1} = x_{i-1}^n \oplus x_i^n \oplus x_{i+1}^n$

The symbol \oplus denotes addition modulo 2 among cell contents. Remark that such rules are linear and involve just the addition of either two bits (rule 90) or three bits (rule 150). In addition, cells with null contents are supposed to be adjacent to the array extreme cells. For a cellular automaton of length $L=10$ cells, configuration rules (90,150,150,150,90,90,150,150,150,90) and initial state (0,0,0,1,1,1,0,1,1,0), Table 2 illustrates the behavior of this structure: The formation of its output sequences $\{x_i^n\}$ ($i=1, \dots, 10$) (binary sequences in vertical) as well as the succession of states (binary configurations of 10 bits in horizontal). After a succession of 62 states, the automaton goes back into the initial state.

Table 2. A linear 90/150 cellular automata of 10 cells

90	150	150	150	90	90	150	150	150	90
0	0	0	1	1	1	0	1	1	0
0	0	1	0	0	1	0	0	0	1
0	1	1	1	1	0	1	0	1	0
1	0	1	1	1	0	1	0	1	1
0	0	0	1	1	0	1	0	0	1
0	0	1	0	1	0	1	1	1	0
0	1	1	0	0	0	0	1	0	1
1	0	0	1	0	0	1	1	0	1
0	1	1	1	1	1	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

A natural way of specifying 90/150 CA is a L -tuple $\Delta_L = (d_1, d_2, \dots, d_L)$, called rule vector, where $d_i = 0$ if the i -th cell satisfies rule 90 while $d_i = 1$ if the i -th cell satisfies rule 150. In the same way, a sub-automaton of the previous automata consisting of cells 1 through i will be denoted by $\Delta_i = (d_1, d_2, \dots, d_i)$. The characteristic matrices M of these CA are tri-diagonal matrices with the rule vector

on the main diagonal, 1's on the diagonals below and above the main one and all other entries being zero. Since the characteristic polynomial of M is of the form $P_M(x) = P(x)^p$, it seems quite natural to construct a multiplicative polynomial CA by concatenation of the automaton whose characteristic polynomial is $P(x)$. In fact, given $P(x)$ the Cattell and Muzio synthesis algorithm [13] presents a method of computing two reversal 90/150 CA corresponding to the given polynomial. That is, two different CA whose output sequences have $P(x)$ as characteristic polynomial. The concatenation procedure for basic CA is based on the following result.

Theorem 1. Let $\Delta_L = (d_1, d_2, \dots, d_L)$ be the representation of a 90/150 binary linear cellular automaton with L cells and characteristic polynomial $P_L(x)$ of degree L . The cellular automaton whose characteristic polynomial is $P_L(x)^2$ is represented by:

$$\Delta_{2L} = (d_1, d_2, \dots, \overline{d_L}, \overline{d_L}, \dots, d_2, d_1) \tag{4}$$

where the overline symbol represents bit complementation.

Sketch of Proof: The key idea of the proof is the recurrence relationship [13]:

$$P_L(x) = P_L(x) + P_{L-1}(x)$$

where $P_L(x)$ is the polynomial corresponding to $\Delta_L = (d_1, d_2, \dots, \overline{d_L})$ and $P_{L-1}(x)$ the polynomial corresponding to the sub-automaton $\Delta_{L-1} = (d_1, d_2, \dots, d_{L-1})$. By means of successive substitutions, CA corresponding to the polynomial $P_L(x)^2$ are represented by the equation (4).

This can be iterated a number of times for successive polynomials and rule vectors:

with $P_L(x)$ we get $\Delta_L = (d_1, d_2, \dots, d_L)$,

with $P_L(x)^2$ we get $\Delta_{2L} = (d_1, d_2, \dots, \overline{d_L}, \overline{d_L}, \dots, d_2, d_1)$,

with $P_L(x)^{2^2}$ we get $\Delta_{2^2L} = (d_1, d_2, \dots, \overline{d_L}, \overline{d_L}, \dots, d_2, \overline{d_1}, \overline{d_1}, d_2, \dots, \overline{d_L}, \overline{d_L}, \dots, d_2, d_1)$

and so on. Thus, the concatenation of an automaton (with the least significant bit complemented) and its mirror image allows us to synthesize linear multiplicative polynomial CA. Remark that the automaton Δ_{2^qL} includes all the previous

Table 3. A linear 90/150 cellular automata generating a self-shrunken sequence

90	150	150	150	150	150	150	90
0	0	0	1	0	1	1	1
0	0	1	1	0	0	1	1
0	1	0	0	1	1	0	1
1	1	1	1	0	0	0	0
1	1	1	0	1	1	0	0
1	1	0	0	1	0	0	0
1	0	1	1	0	1	1	0
0	0	0	0	1	0	1	1

sub-automata $\Delta_{2^s L}$ with $0 \leq s < q$, that is the automaton $\Delta_{2^q L}$ generates all the sequences $\{x_i^n\}$ ($i = 1, \dots, 2^q L$) of the previous sub-automata whose characteristic polynomials are $P_L(x)^p$ with $1 \leq p \leq 2^q$. The choice of a particular state determines the corresponding characteristic polynomial $P_L(x)^p$ of the generated sequences.

4 Sequence Generators in Terms of CA: Self-Shrinking Generator

Now the generation of cryptographic interleaved sequences by means of linear multiplicative polynomial CA is introduced. As an illustrative example, the case of the self-shrinking generator is described:

4.1 The Self-Shrinking Generator

The Self-Shrinking Generator (SSG) was designed by Meier and Staffelbach [14] for potential use in stream cipher applications. The SSG is attractive by its simplicity as it involves a unique LFSR. This generator consists in a LFSR of L stages and primitive polynomial whose PN -sequence $\{a_n\}$ is self-decimated given rise to the self-shrunk sequence $\{z_n\}$ or output sequence of the SSG. The decimation rule is quite simple. In fact, pairs (a_{2i}, a_{2i+1}) ($i = 0, 1, \dots$) of consecutive bits of $\{a_n\}$ are considered in such a way that:

1. If $a_{2i} = 1$, then $z_j = a_{2i+1}$.
2. If $a_{2i} = 0$, then a_{2i+1} is discarded.

The key of this generator is the initial state of the LFSR although the authors recommend that the characteristic polynomial were also part of the key. Periods, linear complexities and statistical properties [14] make the self-shrunk sequences very adequate for their application in stream ciphers.

4.2 Modelling the SSG as a Linear Cellular Automaton

According to [14], the characteristic polynomial of the self-shrunk sequence generated by a LFSR of length L is:

$$P_M(x) = (x+1)^p \quad 2^{L-2} < p \leq 2^{L-1}. \tag{5}$$

In order to model Self-Shrinking Generators in terms of CA, we proceed as follows. As the characteristic polynomial of a self-shrunk sequence is a unique factor multiplied p times and keeping in mind the previous theorem, we can construct its corresponding automaton by concatenation of the basic automaton corresponding to the factor $(x+1)$. Since the automaton corresponding to $(x+1)$ is a simple rule 150 that is $\Delta_1 = (1)$, the application of the previous result allows us to derive the following relationship polynomials/rule vectors:

$$\begin{aligned}
 (x+1) &\leftrightarrow \Delta_1 = (1) \\
 (x+1)^2 &\leftrightarrow \Delta_2 = (0, 0) \\
 (x+1)^4 &\leftrightarrow \Delta_4 = (0, 1, 1, 0) \\
 \vdots &\leftrightarrow \vdots \\
 (x+1)^{2^{(L-1)}} &\leftrightarrow \Delta_{2^{(L-1)}} = (0, 1, 1, \dots, 1, 1, 0)
 \end{aligned}$$

In this way, rule vectors corresponding to 90/150 CA whose characteristic polynomials are products of $(x+1)$ are easily obtained. The last rule vector corresponds to the required automaton. Let us see an illustrative example.

Example 1. Let $\{z_n\} = \{0, 0, 0, 1, 1, 1, 0\}$ be the shrunken sequence generated by a LFSR of length $L = 4$, characteristic polynomial $x^4 + x^3 + 1$ and initial state $(1, 0, 0, 0)$. The sequence $\{z_n\}$ has period $T = 8$, linear complexity $LC = 5$ and characteristic polynomial $P_M = (x+1)^5$. According to the previous results, the 90/150 linear cellular automaton that generates such a sequence is $\Delta_8 = (0, 1, 1, 1, 1, 1, 0)$. Table 3 depicts the automaton's state succession, in this example starting at the state $(0, 0, 0, 1, 0, 1, 1, 1)$, for which the sequence $\{z_n\}$ is generated at the most left cell (in bold) under law 90. Remark that the adjacent cell (law 150) generates exactly the same sequence but shifted upwards one position. It must be also noticed the symmetry of these CA, since sequences produced at symmetric cells are the same shifted $T/2$ positions. Now, different considerations regarding the realization of these CA can be stated:

1. Notice that the form of the computed CA is standard: rule 90 at both extremes and rule 150 at all intermediate positions.
2. The automaton generates all the self-shrunken sequences produced by all LFSRs of length L and primitive polynomial. In this case, the knowledge of the LFSR characteristic polynomial is not necessary.
3. The automaton generates all the self-shrunken sequences corresponding to LFSRs of lengths $\leq L$. That is the longest automaton always includes all the sequences corresponding to shorter automata by starting at symmetric initial states.
4. The implementation of these 90/150 linear models is easy and very adequate for FPGA logic. This characteristic makes it suitable for developments where time execution is relevant as in stream ciphers and in communication systems with high transmission rates.
5. The linearity of the CA-based model as well as the encountered symmetries (see Table 3) allows us to mount a cryptanalytic attack based on the reconstruction of the keystream sequence from portions of intercepted sequence.

5 Conclusions

Interleaved sequences are particular types of pseudorandom sequences that can be generated by linear multiplicative polynomial CA. As the characteristic polynomial of such sequences is a unique factor multiplied by itself a number of times, it seems

quite natural to construct the pertinent automaton by concatenation of the basic automaton associated to such a factor. The linearization process is simple and can be applied to cryptographic examples in a range of practical application. In this way, very popular cryptographic sequence generators conceived and designed as nonlinear generators can be linearized in terms of cellular automata.

Acknowledgments. Work supported by Centro para el Desarrollo Tecnológico Industrial (CDTI) in the frame of project HESPERIA under programme CENIT.

References

1. Gong, G.: Theory and Applications of q -ary Interleaved Sequences. *IEEE Trans. Information Theory* 147, 400–411 (1995)
2. Golomb, S.W.: *Shift Register-Sequences*. Aegean Park Press, Laguna Hill (1982)
3. Fúster-Sabater, A.: Run Distribution in Nonlinear Binary Generators. *Applied Mathematics Letters* 17, 1427–1432 (2004)
4. Jennings, S.M.: Multiplexed Sequences: Some Properties. In: *Proc. Eurocrypt 1983*. LNCS, vol. 149. Springer, Heidelberg (1983)
5. Beth, T., Piper, F.: The Stop-and-Go Generator. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) *EUROCRYPT 1984*. LNCS, vol. 209, pp. 124–132. Springer, Heidelberg (1985)
6. Gollmann, D., Chambers, W.G.: Clock-Controlled Shift Register. *IEEE J. Selected Areas Commun.* 7, 525–533 (1989)
7. Coppersmith, D., Krawczyk, H., Mansour, A.: The Shrinking Generator. In: Stinson, D.R. (ed.) *CRYPTO 1993*. LNCS, vol. 773, pp. 22–39. Springer, Heidelberg (1994)
8. Lidl, R., Niederreiter, H.: *Introduction to Finite Fields and Their Applications*. Cambridge University Press, Cambridge (1986)
9. Kari, J.: A Survey of Cellular Automata. *Theoretical Computer Science* 334, 3–39 (2005)
10. Chaudhuri, P.P., Chowdhury, D.R., Nandi, S., Chatterjee, S.: *Additive Cellular Automata—Theory and Applications*, vol. 1. IEEE Computer Society Press, Los Alamitos (1997)
11. Serra, M., Slater, T., Muzio, J.C., Miller, D.M.: The Analysis of One-dimensional Linear Cellular Automata and Their Aliasing Properties. *IEEE Trans. on Computer-Aided Design* 9(7), 767–778 (1990)
12. Fúster-Sabater, A., Caballero-Gil, P., Delgado, O.: Solving Linear Difference Equations by means of Cellular Automata. In: Corchado, E., Corchado, J.M., Abraham, A. (eds.) *Innovations in Hybrid Intelligent Systems*. *Advances in Soft Computing Series*, vol. 44, pp. 183–190. Springer, Heidelberg (2007)
13. Cattell, K., Muzzio, J.C.: Synthesis of One-Dimensional Linear Cellular Automata. *IEEE Trans. Computers-Aided Design* 15, 325–335 (1996)
14. Meier, W., Staffelbach, O.: The Shelf-Shrinking Generator. In: De Santis, A. (ed.) *EUROCRYPT 1994*. LNCS, vol. 950, pp. 205–214. Springer, Heidelberg (1995)

Ontology-Based Deep Web Data Sources Selection

Wei Fang, Pengyu Hu, Pengpeng Zhao, and Zhiming Cui

Jiangsu Key Laboratory of Computer Information Processing Technology,
Soochow University, Jiangsu Suzhou, China
{064027065001, szzmcui}@suda.edu.cn

Abstract. Many Web applications are all required high quality information from the Deep Web data sources at present. Such as the Deep Web data source discovering, the extract from query results, and the information integration of query results. They are the key technology in the process of Deep Web information integration. This paper proposes a data sources selection method based on domain ontology. It is used for finding the most relevant sets of query interface related to domain ontology, which is a premise for Deep Web information integration. This approach is based on the interface for its own features and the back-end database semantic. Therefore, it makes better performance than existing approaches.

1 Introduction

World Wide Web (WWW) is divided into Surface Web and Deep Web (also named Invisible Web, Hidden Web) according to distributed situation. Surface Web indicates the Web can be indexed by traditional search engine, and the Web is mainly made up of static pages whose hyperlink can reach. Deep Web shows those data resources hidden in Web databases can not be accessed by Web search engines. With the development of Web database application, the WWW is accelerating “deepening” [1], more and more information are hidden behind Web query interface. The search and integration of Internet information both will utilize Web query interface namely Form which can obtain structural data from Web database. Then, the challenge for us is how to identify the query Form, query elements and semantic character. In order to submit correct query to database, and get Deep Web data resources to satisfy the users’ query requirements. Therefore, how to find the most relevant data sources to satisfy the user’s requirements has become more and more important.

In this paper, we utilize ontology to map the attributes of query form, then semantic select the Deep Web data sources. As a global schema Web database should be followed, domain ontology is introduced to the process of data sources selection. Moreover, it can directly map the users’ query to a knowledge ontology, which is also easy for Deep Web information integration and search service.

2 Deep Web Data Source Selection Based on Ontology

2.1 Formal Definition

Given an interface set of Deep Web data sources and a user query which consists of a number of attributes, the goal is to find the data sources subset which mostly satisfy user's requirement. Formal term definitions are defined as following:

Definition 1 (Query Interface). $QI = \langle N, \{A_1, A_2, \dots, A_n\}$, N is the various elements of the Form, A_i indicates logic attribute including in query interface, $A_i = \{L_i, F_j, \dots, k\}$, where L_i is attribute label, F_j is Form component.

Definition 2 (Data Resource Interface Set). Assuming that a Deep Web data source interface set belong to a field is $DWS = \{S_{i1}, S_{i2}, \dots, S_{im}\}$, each data source interface S_{ii} has a corresponding data source ontology template which consists of instance R_i appeared in query interface. The combination of all the instance in ontology template is A , and $A = \cup R_i$. Instance is the corresponding label name, internal attribute name, one or more attributes and domain values are designate to query interface. It is the smallest semantic unit of query interface.

Definition 3 (Query Model). Deep Web is described as the relationship table DB: $Aq = \{a_{q1}, a_{q2}, \dots, a_{qn}\}$ (interface mode) which consist of a series of interface attributes and a series of query attribute results: $Ar = \{a_{r1}, a_{r2}, \dots, a_{rm}\}$ (results mode). Among them, each attribute $a_{qi} \in A$ indicates query attribute received by query interface. While the result attribute $a_{rj} \in A$ indicates attribute of query result. Each query operation can be represent look like the SQL: "Select $a_{r1}, a_{r2}, \dots, a_{rm}$ from DB Where $a_{q1} = val_{q1}, a_{q2} = val_{q2}, \dots, a_{qn} = val_{qn}$, where val_{qi} indicates the attribute values filled in the query form.

Definition 4 (Data Source Rank). Given a Deep Web data source interface set and user query $Q = \{a_{v1}, \dots, a_{vk}\}$, the objective is to find data source sequence $\eta = \{dw_{i1}, dw_{i2}, \dots, dw_{im}\}$ which is ranked according to the semantic similarity.

Definition 5 (Data Source Selection). Given a Deep Web data source query interface set and a user query $Q = \{a_{v1}, \dots, a_{vk}\}$, and the objective is to find data source query interface set $\delta = \{dw_{i1}, dw_{i2}, \dots, dw_{it}\}$ whose similarity to user's query is above a certain threshold.

Definition 6 (Query Cost). Query cost of Deep Web is represented as: $cost(q_i, DB) = \lceil \text{num}(q_i, DB) / k \rceil$, where $\text{num}(q_i, DB)$ indicates all the matching records with query q_i and the maximum value k of result pages displayed in each target Web site[6].

2.2 Semantic Deep Web

Nowadays, the research on Semantic Web and Deep Web is becoming a very hot topic. But integration of the two aspects is relatively few [8]. Here, a novel approach is proposed on adding an ontology level in the Deep Web to construct Semantic Deep Web to get higher accuracy data source of Deep Web related to the query. And given a Deep Web data sources interface set and a user query which consisted of

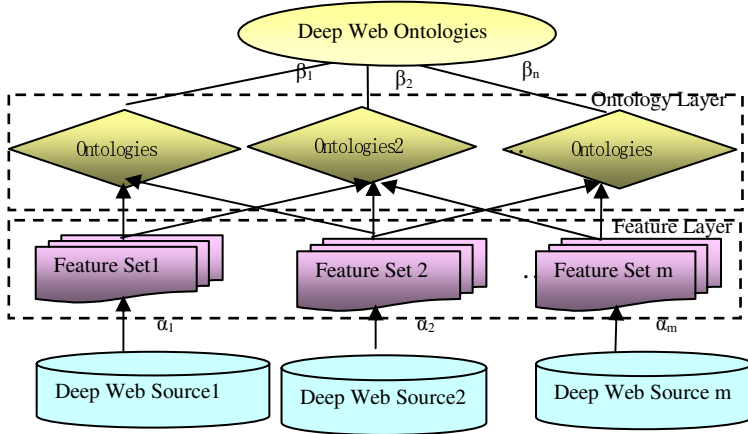


Fig. 1. Architecture of Semantic Deep Web

multi-attributes, by using the ontology ‘WordNet’ and creating the domain ontology to extend query interface sets. Finally we will obtain the most relevant data source sets.

Deep Web sources features are domain-oriented. Deep Web from the same domain is similar in terms of their attribute values respectively [7]. Matching query Form to ontology is a generalization of the schema matching process. We construct semantic feature sets according to generating specific domain ontology, creating a hierarchy tree structure based on the concept, whose lowest node is the instance set belonging to the parent’s concept node. The proportion of classification information possessed by Deep Web can be estimated through instance query. Then we can sum up the features on this attribute of Deep Web better. In addition, retrieving all the data from Deep Web through query can get a best extraction of features. But this is insignificance because when consider the Deep Web access costs and repeated Deep Web content updates, and the frequent queries on Web sites and networks which is unnamable action. Therefore, we select Deep Web data source based on semantic characteristic of query interface to obtain a higher degree of semantic relation and lower query costs. The specific process is shown in Figure 1. Here, α , β represent the ratio of weight values of classified information in corresponding fields.

2.3 Ontology Generation and Management

Ontology is hierarchical structures which describes semantic relationships between concepts in a specific domain. We can automatically build a large, yet domain-specific, ontology by interweaving sub-taxonomies of WordNet with domain specific information extracted from Deep Web service pages [8]. Ontology management is to ascertain the conception of the shared conception and the relation of these conceptions in this domain which handled by users with the help of field expert. The detailed process is to construct a domain ontology and update it according to high frequency concept appeared in the process of query. Deep Web information integration is based

on domain ontology and background knowledge. We can apply graphic method and some certification technology to generate ontology automatically [8,9].

2.4 Similarity Ranking between the Queries

A key issue of semantic selection for Deep Web is how to compute semantic similarity between query and the data sources. Related terms can have various relationships. For example, one might be used in more documents than the others. Consider the relationship between “Car” and “BMW”(a kind of Car). The former has broader usage, and documents besides “BMW” will likely also contain “Car”, while documents containing “Car” will less likely contain “BMW”. So we should consider the relationships between query terms. Term-Matching Based Similarity Ranking is usually an effective, multi-faceted summary of a web page and provides a novel metadata for similarity ranking [7]. A direct and simple use of the domain Ontology building by expert is to calculate the similarity based on the count of shared terms between the query and ontology. Let $q=\{q_1, q_2, \dots, q_n\}$ be a query which consists of n query terms and $O(p)=\{o_1, o_2, \dots, o_m\}$ be the building ontology set of Deep web source s , Formula 1 shows the similarity computation approach based on the shared term count.

Here, that $sim(q, s)$ is defined as o when $O(s)$ is empty.

$$sim(q, s) = \frac{|q \cap O(s)|}{|O(s)|} \quad (1)$$

2.5 Deep Web Selection Strategy

In the process of information integration of Deep Web, when you want to query a certain domain in a combined query interface, if you only transform the query in the combined q interface into each Deep Web in this domain. Apparently, it is not a feasible means. There are three primary reasons [4]. Firstly, there is substantive invisible Deep Web source belonging to a certain domain. Abstractly, we can obtain abundant queries, but the cost is visiting a large number of Deep Web, which is a time-consuming and tedious process; Secondly, every Deep Web cannot satisfy a specific query. Clearly, any domain of Deep Web could not contain all the information in their field. Therefore, it can not satisfy the any queries; the last one is that redundant information exists in most Deep Web in this field. In addition, for query the more Deep Web are visited, the more redundant information are returned, which making it difficult to process redundant information because of continuing increasing. Therefore, the goal of Deep Web data source selection is how to get adaptive data source from large scale Deep Web, which can minimize the number of visited Deep Web, reduce the cost of selection $cost(q_i, DB)$ and take from the redundancy of query result on the premise of satisfying the specific query.

The domain Ontology consist of a collection of entries in the form of the domain probability β that a candidate ontology instance query Qc occurs in Deep Web. From calculating the correspond probability β of Deep Web, we can obtain the harvest rates of some candidate set of Deep Web, then select the most relevant data source.

General speaking, query interface data source selection algorithm is based on the extractive data source interface object DWI . And then calculate the correlativity

between it and the user query. As for the form interface object to the target $DWI = (S, P, M)$ and a query $Q = (W, C \text{ and } O)$, we can calculate the similarity of Q and DWI counterparts of the three components to measure the overall similarity of user query and data sources. The calculation formula of data source correlation as follow:

Definition 7 (Topic Similarity Based on Ontology). Assume that $S=\{s_1, s_2, \dots, s_n\}$ is the set of Deep Web data sources, $C=\{c_1, c_2, \dots, c_i\}$ is a concept collection of topic. Let c_i be a topic concept chosen from the domain ontology. Let i be the size of topic concept. So the topic vector can be described as $R=(w_{1,r}, w_{2,r}, \dots, w_{i,r})$, here $w_{i,r}$ represents the weight of concept c_i in the topic vector. The formula of semantic weight in topic concept is defined as follows:

$$W_{i,j}=f_{ij} * w_{i,r} \tag{2}$$

Here $f_{i,j}=tf_{i,j}/max_i tf_{i,j}$ denotes the frequency of topic concept c_i in the Deep Web data source s_j . Quotient is the frequency of $tf_{i,j}$ in the Deep Web data source s_j in the theme concept c divided by the largest concept c_i in the Deep Web data sources s_j . Ontology-based correlation similarity formula between tuple S and concept query R can be defined as follows [5]:

$$Sim(S_j, R) = \frac{S_j \bullet R}{|S_j| \times |R|} = \sum_{i=1}^i w_{i,j} \times w_{i,r} / (\sqrt{\sum_{i=1}^i w_{i,j}^2} \times \sqrt{\sum_{i=1}^i w_{i,r}^2}) \tag{3}$$

Data Source selection algorithm utilizes the similarity of user query and query interface via keyword matching to realize that. However, the query results are usually inaccurate and high redundant. It is easy to find some irrelevant data sources. Therefore, we can apply an extended model based on the semantic query interface for relevant instance, which makes full use of the query interface semantic features. In a way, it will use domain ontology to extend user query. Then the selection sets of query interface can satisfy user’s requirement better.

Deep Web data sources generally orient to a specific field. The same Deep Web is similar in their attribute values and their attributes distribution. For example, in a product database, the same product brands in different domains of the database are consistent. And in reality, a pre-visited product database may also be visited again in the same domain database. Therefore, not only we need to balance the sample database in a certain domain, which have the feature summaries and construct the domain ontology, but also need to map these features to domain ontology. Finally, we can gain a whole most harvest information from the Deep Web.

Existing semantic relationship model is based on the following experiences: 1) relevant attributes appear in the relevant data sources, that is to say attributes appear in query interface is always related with the relevant subject; 2) relevant data source contains the relevant attributes, the relationship of data source and a related subject is embodied through its attributes in the query interface.

In addition, we also can construct a semantic link map graph based on the common attributes, which is similar to the connective map reflecting the importance weight in the Deep Web data sources. From the map, we can easy to find the other attributes related to the user query and their topics. Meanwhile, it is a simple method to calculate the data source weight. And, we can calculate the relation value between the

attributes appeared in the query interface of the specific domain. Finally, we can determine which interfaces (data sources) are more related to the domain.

3 Experiment Evaluation

In order to perform efficient semantic selection, we have made some pre-computing work, including the depth of every concept c of ontology O , some domain ontology generator, and annotation some Deep Web data sources. The semantic selection is carried out by the following algorithm:

Deep Web Selection Algorithm

```

Input: Query  $q$ , Data Sources
Output: Relevance set of Data Sources (query interface)
Procedure:
  int cost=0;
  Set  $S_i$  =empty;
   $S$  = the size of the Deep Web data source of some
  topic
  converter a query  $q$  from  $Q_I$  to Query concept in-
  stance set  $Q_c$ ;
  Repeat {
    Calculate Relevance( $Q_c, O_i$ ) //between  $Q_c$  and domain
    On tology  $O_i, W_{i,j}=f_{i,j} * w_{i,r}$ 
     $S_i$  = set of Deep Web data source returned; // according
    to the of domain  $O_i$ 
    Quality=  $S_i$  /total( $S$ );
    cost= Cost( $Q_c, O_i$ );
  } until ( min(cost) and Quality( $S_i$ ) > )
  Return a set of DW( $S$ );
End

```

The semantic selection algorithm based on ontology should work well than existing approaches. Query interface is ambiguous and contain few data instances, so it is important to exploit as many features of them as possible, from the Form structure of Deep Web to the data instances, and the semantics features. In addition, there are a few uncertainties in the algorithm. For example, what Ontology we should use, how to select the queries from the Ontology, how many query sets should be returned in each step, and when to stop the selection process.

In order to validate the effectiveness of the semantic-based data source selection method presented in this paper, we made a prototype experiment. First, we collected 15,000 computer books' items from the Web by crawler [5], which was distributed in 10 different domains, such as Jobs, Books, Mobiles, Autos, Music, etc. We collected the number of data sources k from 1 to 10, then calculated the optimal sets of data sources under corresponding different k . Meanwhile, we also built some domain Ontologies and utilized the JWNL API to call the WordNet. Figure 5 is shown the result of comparing the precision of returning the data sources among the probability-based method of data sources selection [2], the traditional random data sources selection

method [3] and the semantic-based data sources selection method. The horizontal axis shows the number of data sources, the vertical axis denotes the precision of the result.

$Precision = \frac{|S_{selected} \cap S_{optimal}|}{|S_{selected}|}$, $S_{optimal}$ denotes the corresponding optimal data source collection under K , $S_{selected}$ presents the data sources set choosing by the corresponding adoption method, $|S_{optimal} \cap S_{selected}|$ presents the intersection between corresponding set of data sources and optimal data sources set. From figure 2 we can see that the semantic-based method has the higher precise and recall than the two methods.

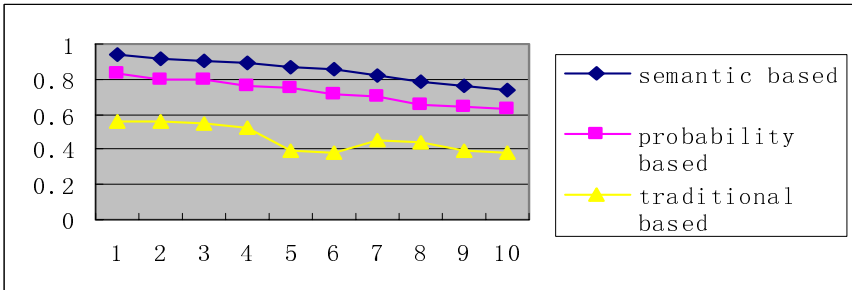


Fig. 2. Comparison of data sources selection by several kinds of methods

Table 1. Performance of various data sources selection methods

Data sources selection method	Kendall's τ
the traditional random method	0.351
the probability-based method	0.589
the semantic-based method	0.786

At the same time, we also use Kendall's method to evaluate the effectiveness of method proposed in this paper. P_1 denotes the order sequence of the data source selection created by the user. p_{2i} shows the data sources order sequences of the probability-based data source selection method [2], the traditional random data source selection method [3], and the semantic-based approach. In P_1 the two data sources have very strict sequence, the same to p_{2i} . If the sequence P_1 and the corresponding p_{2i} have the same order couple, we can say it coordination, or say it incompatible. P is the record number of coordination, and Q records the number of uncoordinated couple. The Kendall's distance between p_1 and p_{2i} is: $(PQ) / (P + Q)$. The results are shown in Table 1, we can see that the semantic-based data sources selection method has made a good performance relatively. Kendall's value has reached 0.786.

4 Conclusion

In this paper we have proposed a novel data sources selection approach based on ontology. As a global schema Web database should be followed, domain ontology is

introduced to the process of data sources selection, which mainly depended on the semantic characters of query interface. And, we can obtain the Deep Web data sources through computing the similarity between query and domain ontology. The experiments also demonstrate that domain-knowledge-based data sources selection shows the greatest promise among existing methods. We believe that the novel domain-knowledge selection method is a practical solution for real world application.

Acknowledgment

This research is partially funded by the Natural Science Foundation of China (60673092); the Key Project of Ministry of Education of China(205059); Jiangsu Software and Integrated Circuit Special Outlay Program([2006]221-41); the 2006 Jiangsu Sixth Talented-Personnel Research Program(06-E-037); and the 2008 Higher Education Graduate Research Innovation Program of Jiangsu Province.

References

1. Ghanem, T.M., Aref, W.G.: Databases Deepen the Web. *IEEE Computer* 73(1), 116–117 (2004)
2. Liu, V.Z., Richard, J.C., Luo, C., et al.: A probabilistic approach for hidden Web database selection using dynamic probing. In: *ICDE 2004* (2004)
3. Ipeirotis, P., Gravano, L.: When one sample is not enough: improving text database selection using shrinkage. In: *Proceeding of the 2004 ACM International Conference on Management of Data* (2004)
4. Liu, W., Meng, X.: Web Database Integration. In: *Proceedings of the Ph.D Workshop in conjunction with VLDB 2006 (VLDB-PhD2006)*, Seoul, Korea, September 11 (2006)
5. Fang, W., Cui, Z., Zhao, P.: Ontology-Based Focused Crawling of Deep Web Source. In: Zhang, Z., Siekmann, J.H. (eds.) *KSEM 2007. LNCS (LNAI)*, vol. 4798, pp. 514–519. Springer, Heidelberg (2007)
6. Wu, P., Wen, J.-R., et al.: Query Selection Techniques for Efficient Crawling of Structured Web Sources. In: *Proceedings of the 22nd International Conference on Data Engineering (ICDE 2006)* (2006)
7. Bao, S., Wu, X., et al.: Optimizing Web Search Using Social Annotation the International World Wide Web Conference Committee (IW3C2). In: *WWW 2007, Banff, Alberta, Canada* (2007)
8. An, Y.J., Geller, J., et al.: Automatic Generation of Ontology from the Deep Web. In: *18th International Workshop on Database and Expert Systems Applications. IEEE, Los Alamitos* (2007)
9. Walny, J.: SemaForm: Semantic Wrapper Generation for Querying Deep Web Data Sources (2006), <http://www.ucalgary.ca/~jkwalny/502/finalreport.pdf>

A Type-2 Fuzzy Set Recognition Algorithm for Artificial Immune Systems

Andrea Visconti¹ and Hooman Tahayori²

¹ Università degli Studi di Milano, Dipartimento di Informatica e Comunicazione, Via Comelico 39/41 Milano, 20135, Italy

² Università degli Studi di Milano, Dipartimento di Scienze dell'Informazione, Via Comelico 39/41 Milano, 20135, Italy

Abstract. In this paper, we suggest a flexible type-2 fuzzy set algorithm for analysing anomalous behavior trends of some system parameters. This algorithm can be implemented in a performance-based Artificial Immune System (AIS) and used as anomalous behavior recognition engine for a biological-inspired Intrusion Detection System (IDS). The suggested algorithm is based on the idea that real-world applications have the necessity of providing a strong, reliable discrimination between normal and abnormal behaviors but such discrimination is not always well-defined. This fact introduces many degrees of uncertainties in rule-based systems and convinced us to implement a type-2 fuzzy set algorithm that can easily manipulate and minimize the effect of uncertainties in our system.

1 Introduction

Developed in the 1990s, Artificial Immune Systems (AISs) are inspired by the workings of the biological immune systems, focusing on their capability to recognize elementary *self* and *non-self* components of the body. It is known that biological immune systems draw up several lines of defense to protect living organisms from all kinds of unwanted invaders. These lines of defense are grouped in chemical and physical barriers, innate immune system and adaptive immune system. Skin, mucus secretions, stomach pH, white blood cells could be enumerated as examples of different defense lines from various groups. Similarly, computer networks rely on several lines of defense e.g. firewall, login policies, antivirus, antispam, etc., to protect servers against unwanted invaders or, more generally, unwanted behaviors originated from within or out of the system.

Real-world applications, based on the artificial immune systems, have the necessity of providing a strong, reliable discrimination between normal and abnormal behavior. This is a crucial point for an AIS because the discrimination between self and non-self behaviors is not always well-defined and such discrimination can introduce many degrees of uncertainties in our model.

A number of ground-breaking solutions to this problem was proposed [9, 22, 8, 6], in particular from the area of computer security and fault detection [5]. Many analogies between computer security systems and biological immune systems could be find in [4, 6, 7]. In [9, 23], interesting approaches introduce the

possibility of using a sequences of system calls as discriminator between normal and abnormal behavior, while references [10,11,3] discuss the design and testing of a LAN traffic anomaly detector that monitors TCP SYN packets for detecting abnormal requests. In [5], authors suggested collecting host and network-based systems data on normal and abnormal behaviors in order to automatically detect, analyse and control future anomaly behaviors. Tarakanov et al. in [22] described a formal model, based on the singular value decomposition to the matrix of connection logs, that can be used as search engine in a biological intrusion detection system. Aickelin and Cayzer [1] and Aickelin and Bentley [2] have suggested interesting approaches based on the Danger Theory. This new theory has shifted control of immunity to the tissues that need protection, hypothesizing that such tissues send signals to the immune system, which in turn will choose the appropriate immune response. Inspired by the behavior of innate immune system, Pagnoni and Visconti in [21] described an artificial immune system based on the working of the innate immune system, while authors in [8,9] suggested applying the negative selection algorithm for detecting network intrusions. Unfortunately, the AISs proposed are not able to completely solve the problem of providing a strong and reliable discrimination between normal and abnormal behavior and no single solution has achieved one hundred percent precision.

In the following, contributions and motivations are discussed in section 2. In section 3, we shortly introduced interval type-2 fuzzy set, while the interval type-2 fuzzy set algorithm for granulating requests is presented in the Section 4. Finally, conclusions are provided in Section 5.

2 Contributions and Motivations

For identifying anomalous behaviors and modeling the possible sources of uncertainties that can be introduces in our system, we suggest an AIS for identifying anomalous behaviors through the analysis of system parameters, based on a type-2 fuzzy set algorithm. Instead of using type-1 fuzzy sets — type-1 are not able to model the sources of uncertainties because their membership functions are well-defined — type-2 fuzzy sets can help us to manipulate and minimize the effect of uncertainties in our system [16].

Looking the problem more closely, we can see a number of sources of uncertainties that make using type-2 fuzzy set suitable. First, a subset of system parameters may be sufficient for describing a specific anomalous behavior, but which and how many parameters belong to such subset is not well defined. Second, once chosen a specific subset of system parameters, the alarm thresholds for some such system parameters may have different values to different experts — i.e. the same values means different things to different people. Third, measurements of system parameters may be imprecise because a high workload situation can introduce an excessive level of background noise. Fourth, also the real data that can be used for the training phase of our AIS may be noisy.

These motivations convinced us to suggest an algorithm based on interval type-2 fuzzy set paradigm for analysing system parameters when one or more of

them show an anomalous trend. The suggested algorithm can be implemented in a performance-based Artificial Immune System (AIS) and used as anomalous behavior recognition engine for a biological-inspired Intrusion Detection System (IDS).

3 Interval Type-2 Fuzzy Set

Zadeh in [25] proposed type-2 fuzzy sets as an extension to ordinary fuzzy sets, that is a fuzzy set with a fuzzy membership function. “A fuzzy set is of type n , $n = 2, 3, \dots, n$ if its membership function ranges over fuzzy sets of type $n - 1$. The membership function of a fuzzy set of type 1 ranges over the interval $[0, 1]$ ” [25]. Putting in more formal form a fuzzy set of type-2 \tilde{A} in a set X is the fuzzy set which is characterized by a fuzzy membership function $\mu_{\tilde{A}} : X \rightarrow [0, 1]^J$ with $\mu_{\tilde{A}}(x)$ being a fuzzy set in $[0,1]$ (or in the subset J of $[0,1]$) denoting the fuzzy grade of membership of x in \tilde{A} [19]. We adopt the notions used in [16], that is,

$$\tilde{A} = \sum_{x \in X} \sum_{u \in J_x} \mu_{\tilde{A}}(x, u)/(x, u) = \sum_{x \in X} \left[\sum_{u \in J_x} f_x(u)/u \right] / x \tag{1}$$

where, $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$, and equivalently $f_x(u) \in [0, 1]$ and, $u \in J_x \subseteq [0, 1]$ denotes a type-2 fuzzy set over X . Here, x is *primary variable* and J_x represents the *primary membership* of x . In this regard $\left[\sum_{u \in J_x} f_x(u)/u \right]$ which is a type-1 fuzzy set, is referred to as *secondary membership function* or *secondary set* and $f_x(u)$ is called the *secondary grade*. Regarding the three-dimensional nature of type-2 fuzzy sets, it is difficult to understand and use them, henceforth, *Footprint Of Uncertainty* (FOU) is defined to be a remedy to partially overcome the problem; partially, since the distribution sitting on top of the FOU is overlooked [15]:

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \tag{2}$$

If $\forall x \in X, u \in J_x$, then $f_x(u) = 1$ type-2 fuzzy set reduces to interval type-2 fuzzy set. Formally a discrete interval type-2 fuzzy set \tilde{A} is characterized by

$$\tilde{A} = \sum_{x \in X} \left[\sum_{u \in J_x} 1/u \right] / x \tag{3}$$

Be noticed that an interval type-2 fuzzy set can be fully described by its FOU, regarding to the uniform distribution sitting on top of the FOU. Given $\bar{u}(x) = \text{Sup}(u)$ and $\underline{u}(x) = \text{Inf}(u)$, *Upper Membership Function* (UMF) and *Lower Membership Function* (LMF) of interval type-2 fuzzy set \tilde{A} would be defined as type-1 fuzzy sets, respectively associated with upper and lower bounds of FOU and shown as

$$UMF(\tilde{A}) = \overline{FOU(\tilde{A})} = \sum_{x \in X} \bar{u}(x)/x \quad LMF(\tilde{A}) = \underline{FOU(\tilde{A})} = \sum_{x \in X} \underline{u}(x)/x \tag{4}$$

Note that for an interval type-2 fuzzy set \tilde{A} then $J_x = [\underline{u}(x), \bar{u}(x)]$ hence

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x = \bigcup_{x \in X} [\underline{u}(x), \bar{u}(x)] \tag{5}$$

Centroid of a type-1 fuzzy set $A = \sum_{x \in X} \mu_A(x)/x$ is defined as

$$C_A = \frac{\sum_{x \in X} x \mu_A(x)}{\sum_{x \in X} \mu_A(x)} \tag{6}$$

Using the Zadeh’s Extension Principle [25], the centroid of a discrete interval type-2 fuzzy set $\tilde{A} = \sum_{x \in X} [\sum_{u \in J_x} 1/u]/x$ that is defined on a discretized domain of $X = \{x_1, \dots, x_N\}$, $u_i \in J_{x_i}$ is given as follows

$$C_{\tilde{A}} = \sum_{u_1 \in J_{x_1}} \dots \sum_{u_N \in J_{x_N}} 1 / \frac{\sum_{i=1}^N x_i u_i}{\sum_{i=1}^N u_i} \tag{7}$$

which is an interval that can be simply represented as $C_{\tilde{A}} = [c_l, c_r]$; \star denotes a t-norm. Although yet no closed form formula for calculating c_l and c_r exists, however, Karnik and Mendel have proposed an iterative algorithm — called *KM Algorithm* — for calculating the bounds exactly [12]. Moreover, in [24, 18], Mendel and Wu have proved that the end points of the centroid of an interval type-2 fuzzy set are bounded and have driven formulas for calculating these bounds, that is

$$\underline{c_l} \leq c_l \leq \bar{c_l} \quad , \quad \underline{c_r} \leq c_r \leq \bar{c_r} \tag{8}$$

Toward the aim, using (6), the centroid of $LMF(\tilde{A})$ and $UMF(\tilde{A})$ are:

$$C_{LMF(\tilde{A})} = \frac{\sum_{i=1}^N x_i \underline{u}(x_i)}{\sum_{i=1}^N \underline{u}(x_i)} \quad , \quad C_{UMF(\tilde{A})} = \frac{\sum_{i=1}^N x_i \bar{u}(x_i)}{\sum_{i=1}^N \bar{u}(x_i)} \tag{9}$$

So the bounds would be given as

$$\bar{c_l} = \min\{C_{LMF(\tilde{A})}, C_{UMF(\tilde{A})}\} \quad , \quad \underline{c_r} = \max\{C_{LMF(\tilde{A})}, C_{UMF(\tilde{A})}\} \tag{10}$$

$$\begin{aligned} \underline{c_l} &= \bar{c_l} - \frac{\sum_{i=1}^N (\bar{u}(x_i) - \underline{u}(x_i))}{\sum_{i=1}^N \bar{u}(x_i) \sum_{i=1}^N \underline{u}(x_i)} \\ &\quad \times \frac{\sum_{i=1}^N \underline{u}(x_i)(x_i - x_1) \sum_{i=1}^N \bar{u}(x_i)(x_N - x_i)}{\sum_{i=1}^N \underline{u}(x_i)(x_i - x_1) + \sum_{i=1}^N \bar{u}(x_i)(x_N - x_i)} \end{aligned} \tag{11}$$

$$\begin{aligned} \bar{c_r} &= \underline{c_r} + \frac{\sum_{i=1}^N (\bar{u}(x_i) - \underline{u}(x_i))}{\sum_{i=1}^N \bar{u}(x_i) \sum_{i=1}^N \underline{u}(x_i)} \\ &\quad \times \frac{\sum_{i=1}^N \bar{u}(x_i)(x_i - x_1) \sum_{i=1}^N \underline{u}(x_i)(x_N - x_i)}{\sum_{i=1}^N \bar{u}(x_i)(x_i - x_1) + \sum_{i=1}^N \underline{u}(x_i)(x_N - x_i)} \end{aligned} \tag{12}$$

For father details on type-2 and interval type-2 fuzzy set refer to [16, 17, 19, 25].

4 Interval Type-2 Fuzzy Set Based Algorithm

There are some factors — or system parameters — that their changes may have indication of an attack. In fact, if the system is under attack some parameters would change rapidly. An example of an abrupt change in the number of interrupts per second during a Distributed Denial of Services (DDOS) is shown in Figure 1. In the following, for simplicity, we suppose to have M system parameters and M artificial lymphocytes, each of which checks a system parameter; in reality, for security policies, we engineered artificial lymphocytes that redundantly check all the parameters. The designed independent-artificial lymphocytes are to monitor the factors or system parameters $F = \{f_1, \dots, f_M\}$ and report their changes. We show the changes of the system parameters as a vector $V = [v_1, \dots, v_M]$, where $v_i, i \in \{1, \dots, M\}$ indicates the percentage of the changes in the parameter f_i with respect to the latest measurement.

We have surveyed a number of experts on their intuitions on how far the changes in different system parameters may be an indication of an attack. Using the *person MF approach* discussed in [14] that enable modeling intra and inter uncertainties about the effects of changes in the system parameters, primarily we asked each expert to provide a broad-brush FOU for each parameter in the domain of $X = [0, 100]$; it is argued that it is more natural and feasible than asking for a crisp number [14]. Aggregating all equally weighted experts' FOUs on each parameter through mathematical operation of union and in order to get to a parsimonious one, using the advice of [14] and filling the results of the union, we reached to an *interval type-2 fuzzy map* (IT2FM) of each parameter. Using (5) the IT2FM of the system parameter f_i would be extensively shown as

$$IT2FM(f_i) = \sum_x [\underline{u}_{f_i}(x), \bar{u}_{f_i}(x)] / x = \sum_x \tilde{\mu}_{f_i}(x) / x \tag{13}$$

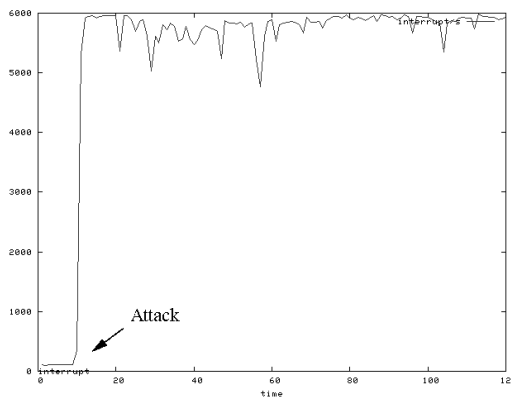


Fig. 1. Interrupts/sec during a DDOS attack with a small number of users that surf the website

where $\tilde{\mu}_{f_i}(x) = [\underline{u}_{f_i}(x), \overline{u}_{f_i}(x)] \subseteq [0, 1]$ represent the uncertain attack-indication of x percent change in the system parameter f_i . We also assumed that $\tilde{\mu}_{f_i}(x) = \tilde{\mu}_{f_i}(0)$, $x < 0$ and $\tilde{\mu}_{f_i}(x) = \tilde{\mu}_{f_i}(100)$, $x > 100$.

To decide if a change of a system parameter needs a special attention, based on the experts recommendations, we devised three non-necessarily pair-wise disjoint regions in $[0,1]$, namely *White* $= [\underline{r}_1, \overline{r}_1]$, *Yellow* $= [\underline{r}_2, \overline{r}_2]$ and *Red* $= [\underline{r}_3, \overline{r}_3]$. The new measurement of the system parameter f_i cause v_i to be renewed, then the region to which $\tilde{\mu}_{f_i}(v_i)$ is more close by calculating their distance i.e., $\tilde{\mu}_{f_i}(v_i) \approx [\underline{r}_i, \overline{r}_i]$ will be found where

$$d(\tilde{\mu}_{f_i}(v_i), [\underline{r}_i, \overline{r}_i]) = Inf(d(\tilde{\mu}_{f_i}(v_i), White), d(\tilde{\mu}_{f_i}(v_i), Yellow), d(\tilde{\mu}_{f_i}(v_i), Red)) \tag{14}$$

The reasonable distance function used [20] is, $d([\underline{a}, \overline{a}], [\underline{b}, \overline{b}]) = Max(|\underline{a}-\underline{b}|, |\overline{a}-\overline{b}|)$.

$\tilde{\mu}_{f_i}(v_i)$ being *White*, indicates that changes of the system parameter f_i has not been abnormal, whereas $\tilde{\mu}_{f_i}(v_i) = Yellow$ issues a warning and the corresponding agent will measure the f_i instantly and will *add* the new registered changes to v_i instead of replacing it. If it was backed to the *White* position, then the monitoring of the parameter will be set to normal. However if it is positioned in the *Red* region, a command will be issued to all other agents whose system parameters had not been in *Red* region, to measure their corresponding parameters instantly and refresh their related cell in V , based on which the *Interval Type-2 Fuzzy Map* of the System — $IT2FM(sys)$ — will be built. Initially, $IT2FM(sys)$ is set to null, i.e., $IT2FM(sys) = \sum_x 0/x$. When the system is put in the *Red* position and after V is refreshed the map will be built iteratively as

$$\tilde{\mu}_{sys}(v_i) \leftarrow Hull(\tilde{\mu}_{sys}(v_i) \cup \tilde{\mu}_{f_i}(v_i)) \tag{15}$$

where $i \in \{1, \dots, M\}$ and $Hull([\underline{a}, \overline{a}] \cup [\underline{b}, \overline{b}]) = [Min(\underline{a}, \underline{b}), Max(\overline{a}, \overline{b})]$.

To make the final decision, i.e. whether there has been an attack or not, using (9)–(12) artificial lymphocytes calculate the boundaries of the centroid of $IT2FM(sys)$ and compare the result with the predefined granules $G = \{Non-Attack, Suspicious-Non-Attack, Non-Decidable, Suspicious-Attack, Attack\}$. The granules are defined through first surveying the experts using the same method of *person MF approach*, aggregating all the equally weighted individual *person MFs* and then filling them and finally drawing their centroid using the KM-Algorithm. More precisely, we can show G as

$$G = \{g_i \mid i = 1, \dots, 5\} \quad where \quad g_i \triangleq [\underline{cg}_i, \overline{cg}_i] \tag{16}$$

To draw the decision, lymphocytes calculate the distance of the centroid bounds of $IT2FM(sys)$ i.e. $C'_{IT2FM(sys)} = [[\underline{c}_l, \overline{c}_l], [\underline{c}_r, \overline{c}_r]]$ with each and every granules in G to find the minimum distance; the granule with the minimum distance to the centroid bound indicate the status of the system, more formally,

$$System_Status \approx g_l, \quad g_l \in G \tag{17}$$

given

$$\begin{aligned} d(C'_{IT2FM(sys)}, g_i) &= \mathop{\text{Inf}}_i \left(d(C'_{IT2FM(sys)}, g_i) \right) = \\ &= \mathop{\text{Inf}}_i \left(\text{Max} \left(d([\underline{c}_l, \bar{c}_r], g_i), d([\bar{c}_l, \underline{c}_r], g_i) \right) \right) \end{aligned} \quad (18)$$

5 Conclusions

Getting use of the granular computing concepts administrators recognize intrusion through the usage of proper information granules — e.g. Attack, Suspicious-Attack, Non-Decidable, etc., — that play a central rule in the process of abstraction of the problem. The proposed scheme revealed that granular computing would be appropriate for intrusion detection and can be implemented in an Artificial Immune System for monitoring, analysing system parameters, and identifying anomalous behaviors.

Although still at a preliminary stage, first results revealed that AIS is able to (a) quickly detect anomalous behaviors previously encountered, (b) recognize proliferation of foreign processes, and (c) quickly identify attacks with a strong fingerprint such as denial of service, buffer overflow, and password guessing attacks. On the other hand, preliminary testing has shown that in presence of many net-surfing users recognizing attacks with a feeble fingerprint can be increasingly difficult, because a high workload situation may introduce an excessive level of background noise, enhancing the risk of false positive.

It should be stressed that yet, no single method, biological or artificial, can achieve one hundred percent precision. For these reasons, the suggested system is not meant to replace firewalls, login policies, or antivirus because it cannot blocks every kind of attack. The proposed artificial immune system should be used in conjunction with other complementing technologies either biologically inspired or not.

Acknowledgements

This research was funded by the State of Italy via FIRST.

References

1. Aickelin, U., Cayzer, S.: The Danger Theory and Its Application to Artificial Immune Systems. In: Proc. of 1st Int. Conf. on Artificial Immune Systems (2002)
2. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger Theory: The Link between AIS and IDS? In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)
3. Balthrop, J., Forrest, S., Glickman, M.: Revisiting lisy: Parameters and normal behavior. In: Proc. Congress on Evolutionary Computation, pp. 1045–1050 (2002)
4. Dasgupta, D.: Immune-based intrusion detection system: A general framework. In: Proc. of the 22nd National Information Systems Security Conference (1999)

5. Dasgupta, D.: Advances in Artificial Immune Systems. *IEEE Computational Intelligence Magazine* (November 2006)
6. Forrest, S., Hofmeyr, S., Somayaji, A., Longstaff, T.: A sense of self for UNIX processes. In: *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy* (1996)
7. Forrest, S., Hofmeyr, S., Somayaji, A.: Computer immunology. *Communication of ACM* 40(10), 88–96 (1997)
8. Forrest, S., Glickman, M.R.: Revisiting LISYS: Parameters and Normal behavior. In: *Proceedings of the Special Track on Artificial Immune Systems, Proceedings of the 2002 Congress on Evolutionary Computation* (2002)
9. Hofmeyr, S., Somayaji, A., Forrest, S.: Intrusion Detection using Sequences of System Calls. *Journal of Computer Security* 6(3), 151–180 (1998)
10. Hofmeyr, S., Forrest, S.: Architecture for an artificial immune system. *Evolutionary Computation* 8(4), 443–473 (2000)
11. Hofmeyr, S.: An immunological model of distributed detection and its application to computer security. PhD thesis, University of New Mexico (1999)
12. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. *Inf. Sci.* 132, 195–220 (2001)
13. Kim, J., Bentley, P.: The human Immune system and Network Intrusion Detection. In: *Proceedings of 7th European Congress on Intelligent techniques Soft Computing*, Aachen, Germany (1999)
14. Mendel, J.M.: Computing with Words and Its Relationships with Fuzzistics. *Information Sciences* 177, 988–1006 (2007)
15. Mendel, J.M., Bob John, R.I.: Footprint of uncertainty and its importance to type-2 fuzzy sets. In: *Proc. 6th IASTED Int. Conf. Artificial Intelligence and Soft Computing*, Banff, Canada, pp. 587–592 (July 2002)
16. Mendel, J.M., Bob John, R.I.: Type-2 Fuzzy Sets Made Simple. *IEEE Transactions on Fuzzy Systems* 10(2) (April 2002)
17. Mendel, J.M., Bob John, R.I., Liu, F.: Interval Type-2 Fuzzy Logic Systems Made Simple. *IEEE Trans. Fuzzy Systems* 14(6), 808–821 (2006)
18. Mendel, J.M., Wu, H.: Centroid uncertainty bounds for interval type-2 fuzzy sets: forward and inverse problems. In: *Proc. of IEEE Int'l. Conf. on Fuzzy Systems*, Budapest, Hungary (July 2004)
19. Mizamoto, M., Tanaka, K.: Some properties of fuzzy set of type-2. *Inform. Control* 31, 312–340 (1976)
20. Moore, E.R.: *Interval Analysis*. Prentice-Hall, Englewood Cliffs (1966)
21. Pagnoni, A., Visconti, A.: An Innate Immune System for the Protection of Computer Networks. In: Baltes, B.R., et al. (eds.) *Proceedings of the 4th International Symposium on Information and Communication Technologies* (2005)
22. Tarakanov, A.O., Skormin, V.A., Sokolova, S.P.: *Immunocomputing: Principles and Applications*. Springer, New York (2003)
23. Warrender, C., Forrest, S., Pearlmutter, B.: Detecting intrusions using system calls: Alternative data models 1999. In: *IEEE Symposium on security and Privacy* (1999)
24. Wu, H., Mendel, J.M.: Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* 10, 622–639 (2002)
25. Zadeh, L.A.: The Concept of a Linguistic Variable and Its Application to Approximate Reasoning-I. *Information Science* 8, 199–249 (1975)

Symbolic Hybrid Programming Tool for Software Understanding

Erkki Laitila

Jyväskylä University, Box 40014, Jyväskylä Finland
SwMaster Oy, Sääksmäentie 14, 40520 Jyväskylä Finland
erkki.laitila@swmaster.fi

Abstract. Research on source code understanding has attracted computer scientists for decades. It is known [12] that the code is the functional standard for each computer system. One can assume that most functional errors can automatically be captured from code. In this paper we describe a metatheory for software understanding, with pieces of domino as a metaphor. With the help of the tiles of the game, each of the stages of the data flow in software reverse engineering can be modeled. The last element in this hybrid construction is the maintainer, who plans modifications by using symbolic code information based on previous game movements. This process was evaluated for Java.

Keywords: Program comprehension, reverse engineering, hybrid architecture.

1 Introduction

In this paper we specify a metatheory for program comprehension (PC)¹, which aims helping the maintainer to find the relevant places in the code, i.e., the potential candidates for program changes in future versions. The traditional methods for code analysis are static analysis and dynamic analysis. Both have their limitations and drawbacks: it is known that static analysis, in spite of its several methods [10], is not sufficient for modern object-oriented programs [17], and that dynamic analysis requires often too much work from the maintainer to create a new test case and an understanding of the situation. Dynamic analysis is thus too expensive for many practical situations [14]. There is a need for a more flexible, higher-abstraction method for PC.

Symbolic analysis is a novel methodology [8], abstracting code elements as symbols. The connections between the elements are expressed as predicates, typical for logic programming. That helps the user to deduce, from the output of a simulation of the critical objects, whether the behaviour meets the assumed requirements, or not.

In this paper we depict the steps from parsing the code to knowledge capture as automata, which mirror the movement of the pieces in a game of Domino. In that “game” there are some symbolic rules on how to play. The indicator for parsing is a domino tile with one dot; it describes the source code such as Java or C++. We then assume that moving to the next step adds a new dot to the tile, until the last step

¹ Program comprehension (PC) is a synonym for software understanding.

(six dots) is reached. The last step offers a chance for the user to play iterative game. The process from the first to the last tile specifies a single source code simulation session. We argue that for symbolic analysis this domino tile analogy is relevant: the objective here is to minimize the number of tiles and movements illustrating the corresponding sub-symbolic code area of whatever size, from the whole application to just one line of Java. This division to both symbolic and sub-symbolic concepts can be regarded as an extremely flexible approach when compared with the traditional methods.

2 Using Symbolic Analysis for Source Code Interpretation

Symbolic analysis (SA) [8] can be divided into four (4) technology spaces, which coherently build a methodology for reverse engineering. They specify the code information data flow for program comprehension as follows:

1. **GrammarWare** describes those technological solutions of SA that are grammar related. One of its essential constructions is the symbolic language, *Symbolic* (see Table 2), which is able to express a semantic correspondence for each programming language term (Java and others) as a Prolog predicate [18].
2. **ModelWare** tells how source code can efficiently be modelled as a sub-symbolic layer by using very small elements, software atoms (see Fig. 1), as its components.
3. **SimulationWare** describes how to simulate computer programs using automata theories (see Fig. 2) [7]. Modeling Java semantics can be very challenging, due to dynamic bindings and many other special features. We have implemented this technology space with the help of a symbolic abstract machine (SAM).
4. **KnowledgeWare** is a set of capabilities for the user with which a maintenance task can be conquered and divided into smaller parts, i.e., program concepts. These can then be mapped into the code for opportunistic investigation (Fig. 3).

The most central part of the SA construction is the atomistic model. The code is transformed into software atoms while maintaining the original program semantics. The only information carrier in SA is the *Symbolic* language, which is able to connect all relevant artifacts, both nested and sequential as deduction chains to be used for PC.








2.1 Automata in Symbolic Analysis

All the functions of SA are automata, and are abstracted as Domino tiles in this paper. They have been programmed in Visual Prolog [18] in order to implement the corresponding tool, JavaMaster [8].

Theories for automata have been described in the Chomsky hierarchy [3,7]. These theories can be simplified into state transition tables. A general notation for an automaton is a table whose columns consist of the current state, a condition, the next state, the necessary IO-functions and an optional return value for each line.

In Table 1 we list the data formats for the seven automata of SA with the proposed icons illustrating the Domino tiles with their role.

Table 1. Data for the automata of symbolic analysis illustrating reverse engineering phases

Phase	Data for the automaton	Formulation for the data	Icon
1	Grammar	Grammar definition (like EBNF [4])	
2	Original language	Parse tree (Java, C++ etc).	
3	Symbolic language	Abstracted semantics of the language	
4	Atomistic model	Atoms, elements to form the model	
5	Input for simulation	List of atoms to be simulated	
6	Output from simulation	An ordered list of executed atoms	
7	Human (the user)	Knowledge created by the person (information ladder).	

2.2 Programming Language as Input

The first phase of Table 1 is the grammar. A grammar for a programming language describes the program by using a syntax, which consists of tokens. Grammar terms are collected as production rules to create the definitions for the production names. In the grammar of Java (3rd ed. [6]) there are 247 production names, 368 production rules and 56 lists to specify sequences for the terms.

It is typical that in a grammar only the syntax is presented for the language. However, syntax is not enough to specify the semantics of the tool. Therefore, in SA we have developed a convention to add a semantic id to each grammar term. For example, we have the following presentation for the production name *Statement* [6]:

```

Statement =
  Block                                → blk,
  "assert" Expression Expression ";"   → assert_,
  "if" ParExpression Statement OptElse → iff,
  "for" "(" ForControl ")" Statement   → for,
  "while" ParExpression Statement     → while,
  "do" Statement "while" ParExpr ";"  → do,
  "try" Block CatchBlock              → try,
  "switch" ParExpression "{" Block "}" → switch,
  "synchronized" Expression Block     → sync,
  "return" Opt_Expression ";"         → return_,
  "throw" Expression ";"              → throw_,
  "break" Opt_Identifier              → break_,
  "continue" Opt_Identifier           → continue_,
  StatementExpression ";"             → stmtExpr,
  Identifier ":" Statement            → labelStmnt.

```

The above grammar definition for *Statement* creates 15 production rules, e.g., grammar terms. There is a semantic id to the right of each arrow. The references are collected from the left side. In the atomistic model there is an atom for each grammar term having the semantic id and with its references as the *command* of the atom.

2.3 Transformation 1-2: Automaton from Grammar to Parse Tree

Grammar is a flat definition, but parser generator, the definition in 1-2, transforms the corresponding token list to a strongly hierarchical parse tree (Phase 2 in Table 1). Traditionally this is done with the help of the *Abstract Syntax Tree* (AST) method [9]. AST has a serious drawback, however: typically AST implementation consists of individual cells and variables that do not have a consistent and coherent semantics corresponding to the original grammar terms. So the responsibility to program the semantics is left to the developers. In programming AST structures, the grammar is in the head of the programmer, not in the development tool. This runs counter to the main idea of GrammarWare, i.e., to automatize parsing and to minimize the programming efforts.

2.4 Transformation 2-3: Automaton for Java-to-Symbolic - Translation

Symbolic is an artificial language to abstract Java and other formal languages. Its structures are compatible programming languages. One should note that the only base term in *Symbolic* is *clause*. Hence, all the references in that language can be identified as *clauses*. Clause is our selection for the foundation of symbolic processing, because all its tokens can be parsed and written as clauses. Each clause type can contain some exact subtypes.

In the following, some subtypes of *clause* and their meanings are listed:

Table 2. Examples of Symbolic clause types; Phase 3 in Table 1

<i>Nr</i>	<i>Symbolic clause type</i>	<i>Corresponding predicate <T></i>
1	Definitions	def(defClause)
2	Creators	creator(createClause)
3	References	ref(refClause)
4	Method calls	get(getClause)
5	Change clauses	set(setClause)
6..12	<i>Other clauses</i>	(See closer [8]).

2.5 Transformation 3-4: Automaton for Creating an Atomistic Model

A symbolic model weaver creates, for each leaf *X* of the parse tree, an atom *Y*. *X* has always the format $f(X_1..X_N)$ having *f* as a semantic id and with arguments $X_1..X_N$. The weaved notation in the atom *Y* is then $f(Y_1..Y_N)$. It is an explicit definition, where each Y_i points to one or more lower atoms, which are created by using a lower level weaving function for each input term $X_1..X_N$.

The atom (See Fig.1) is a hybrid object, which seamlessly combines the benefits of the object-oriented and logic programming paradigms. As the semantics for each atom there is the *Command* variable the type of which is clause. The base class for the atoms is *Symbolic*, which defines the universal language for the model. Class *SymbolicElement* is the super class for all the atoms whose template name is *Symbolic<T>Element* (See Table 2).

It is essential in the architecture of Fig.1 that simulation be activated from outside only by calling the *run*-method, which only needs its *Command*-field for execution.

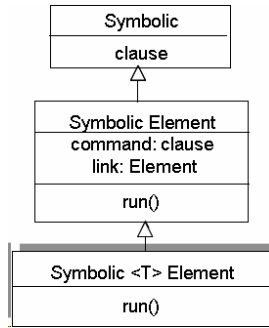


Fig. 1. The architecture of a software atom as a class hierarchy; Phase 4 in Table 1

Simulation for an atom consists of a finite domain automaton [7], which internally calls other atoms to execution. After execution, each atom returns its result to the caller. This simple call/return sequence creates the foundation for atomisticity [8].

2.6 Transformation 4-5: Input Sequence for Simulation

The definition for a simulation (Phase 5 in Table 1) is an ordered list of independent atoms. Creating the list is simple, because each atom automatically invokes its callees. The user creates the sequence by copying the relevant *caller* atoms to the queue. For example, the atom for the *main* method calls the entire logic of an application.

2.7 Transformation 5-6: Automaton to Simulate the Code

The computer simulation is based on the Turing machine model [7], where there is a tape to be executed. Our novel implementation uses atoms and their Command-fields as inputs (See Fig.2). During execution the simulator adds intermediate results (p) and (q), such as assignments and invocations, to the tape, to be used for later analyses. After execution the tape contains a complete run-time history from the simulation.

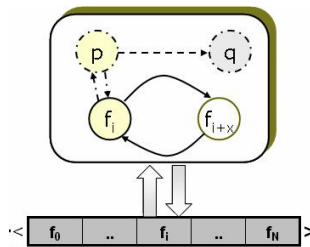


Fig. 2. Symbolic Turing Machine creates an output tape; Phase 6 in Table 1

2.8 Transformation 7: Human as an Observer of the Output Sequence

Observations of the user (See Fig.3) can be directed either forwards for planning changes or backwards for troubleshooting purposes. Maintainers use conjectures and

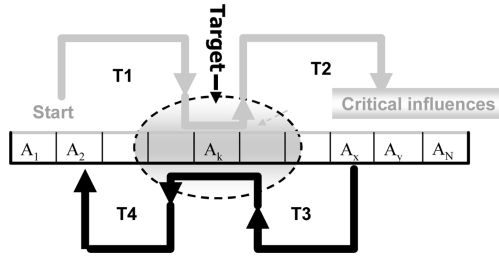


Fig. 3. Checking program correctness based on an output sequence; Phase 7 in Table 1

hypotheses as drivers for understanding [16]. We formulate these hypotheses via atoms as formal, provable theorems (See T1 to T4 in Fig.3) for argumentation. The theorems are constraints to tell what the target atoms should activate, and what they should not do. Temporal logic is a useful formulation for that purpose [11].

3 Evaluating the Theory with JavaMaster

We have created a tool, JavaMaster, to support symbolic analysis for Java (See Fig.4).

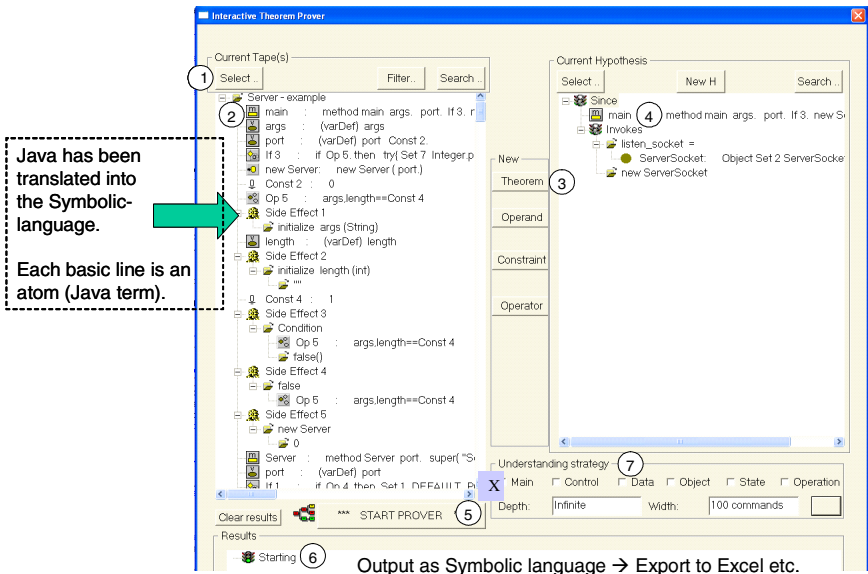


Fig. 4. Exploring program correctness with the JavaMaster tool [8]; Phases 5 to 7

3.1 Domino Tiles Analogue for the Tool

The proving process is initiated by loading the relevant Java code according to Phases 1 to 4 of Table 1. For each simulation a test sequence is created (Phase 5). The simulated







sequence (Phase 6) is displayed in the window by using the Select button (1). It is printed downwards, one atom on each line (2). The user creates theorems (3) according to Phase 7 for ensuring the program correctness.

3.2 Evaluating the Domino Play and Results of Implementing the Metatheory

We evaluated the understanding process (Domino tile buttons 1 to 6) by using a test program *Server.java* [5] as an input. The result from simulating its *main* goal produced an output sequence (See the left side of Fig.4) containing 234 atoms. We checked that it contained all relevant object instantiations, activations, and functions.

In Fig.4 the *main*-method has been selected as the starting point (4) of Fig. 3. Below it, an invocation which creates a *ServerSocket* object, has been specified as a target. When the theorem (4) is ready, the proving process is started with the **Start Prover** button. The results are shown (6). They are in the symbolic notation, enabling iterative playing with Phases 5 to 7. Therefore, it is possible to continue the analysis by using them as new input. The output can be filtered by selecting *Understanding strategy* (7). In Table 3 there is a summary of the results of the automata of Table 1.

Table 3. Results from transformation tiles of the automata of symbolic analysis

<i>Section</i>	<i>Transformation</i>	<i>Icon</i>	<i>Results</i>
See 2.3	Parser generation		Parsing Java complete, 100%
See 2.4	Abstracting programs		Java-to-Symbolic translation 100%
See 2.5	Model weaving		Atomistic model complete, 100%
See 2.6	Specifying simulations		Complete - sometimes laborious. ²
See 2.7	Simulating		Sequential simulation is supported. ³
See 2.8	Knowledge capture		Systematic process enables splitting a problem into actions of Fig.3.

4 Related Work and Conclusions

Modeling source code has been studied for many decades. The most popular method for that purpose is UML [13], even though it doesn't include code simulation. *Architecture Driven Modernization* [1], which approaches modeling from an opposing direction to one of the basic UML's supporting standard, Model Driven Architecture, is too complex to be used as a basis for a program understanding strategy itself. Instead, this paper describes symbolic analysis as a novel methodology for software understanding. The purpose of it is to minimize the number of relevant elements, e.g., atoms, in the play. Loading the relevant code forms Phases 1 to 4 in the "Domino play". With the help of that information the user can select the simulations of Phases 5 and 6. From their output the user can then extract hierarchical maintenance knowledge (Phase 7) in order to decide which elements should be changed. The biggest

² The tool makes a list of all ambiguous cases for each atom to be used for user selections. The user can then select the best case for each situation in interactive simulation.

³ Some Java-commands are omitted (error handling and real time functions).

limitation for the work is that parallel computing cannot be analyzed stepwise. The proposed play is, however, adequate for simulating sequential computations in order to capture the corresponding behavior model and the corresponding influences.

Acknowledgments. This work has partially been funded by Jyväskylä University. The tool was developed by SwMaster Ltd.

References

1. ADM, Architecture Driven Modernization. Object Management Group, <http://adm.omg.org>
2. Binkley, D.: Source Code Analysis: A Road Map. In: Future of Software Engineering, ICSE 2007 (2007)
3. Chomsky, N.: Three models for the description of language. *IEEE Transactions on Information Theory* 2(3), 113–124 (1956)
4. EBNF. Information technology - syntactic metalanguage - extended BNF, ISO/IEC 14977, International Organization for Standardization (2001)
5. Flanagan, D.: Java Examples in a Nutshell: Example 7-5 (Server.java), (referred August 10, 2007). O'Reilly, Sebastopol (2000), <ftp://ftp.ora.com/examples/nutshell/java/examples>
6. Gosling, J., Joy, B., Steele, G., Bracha, G.: The Java Language Specification, 3rd edn. Addison-Wesley, Boston (2005)
7. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, Reading (1979)
8. Laitila, E.: Symbolic Analysis and Atomistic Model as a Basis for a Program Comprehension Methodology. PhD-thesis, Jyväskylä University (2008), <http://dissertations.jyu.fi/studcomp/9789513932527.pdf>
9. Mak, R.: Writing Compilers and Interpreters, 2nd edn. John Wiley & Sons, Chichester (1996)
10. Müller, H.A., Jahnke, J.H., Smith, D.B., Storey, M.-A.D., Tilley, S.R., Wong, K.: Reverse engineering: a roadmap. In: ICSE 2000 (2000)
11. Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th Annual Symposium on the Foundations of Computer Science, New York, pp. 46–57 (1977)
12. Reeves, J. W.: Code as design: Three essays, Developer.* What Is Software Design? http://www.developerdotstar.com/mag/articles/reeves_design_main.html
13. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Reading (1999)
14. Sneed, H.M.: Reverse engineering of test cases for selective regression testing. In: CSMR 2004, pp. 69–74. IEEE, Los Alamitos (2004)
15. Turing, A.: On Computable Numbers with an Application to the Entscheidungsproblem. *Proc. London Math. Soc.* 2(42), 230–265 (1936)
16. Vans, A.M., von Mayrhauser, A., Somlo, G.: Program understanding behavior during corrective maintenance of large-scale software. *Human-Computer Studies* 51, 31 (1999)
17. Wilde, N., Huit, R.: Maintenance Support for Object-Oriented Programs. *IEEE Transactions on Software Engineering* 1 SE-18, 1038–1044 (1992)
18. VisualProlog. The Visual Prolog Development-tool, <http://www.visual-prolog.org>

Characterizing Massively Multiplayer Online Games as Multi-Agent Systems

G. Aranda, C. Carrascosa, and V. Botti

GTI-IA, DSIC, Universidad Politécnica de Valencia.

Camino de Vera s/n 46022,

Valencia, Spain

garanda@dsic.upv.es, carrasco@dsic.upv.es, vbotti@dsic.upv.es

Abstract. The number of players of Massively Multiplayer Online Games is increasing quite fast, and the mechanics of these virtual worlds are getting more and more complex. Due to the eminently distributed nature of these games and their growing necessity of modern AI techniques, it is time to introduce design methods that take advantage of the power of Multi-Agent Systems in order to face the challenges of designing a modern Multiplayer Online Game. In this work, an introductory step into the characterization of Multiplayer Online Games as Multi-Agent Systems is presented. It is based on using mechanisms and concepts derived from the agent theory, like Electronic Institutions and Agent Organizations, as foundations for building the game system and the virtual environments.

1 Introduction

Massively Multiplayer Online Games (MMOGs) videogames played online, allowing players, through their self-created digital personas called 'avatars', to interact not only with the gaming software (the designed environment of the game and the AI-controlled characters within it) but with other players' avatars as well. These virtual worlds can be persistent, social and somehow material worlds, where players are largely free to do what they want. The challenge of building and deploying an online game has become more complex than ever, rivalling the complexity and cost of other non-games related network and AI projects, such as distributed simulation networks or online auction houses[7]. More information about MMOGs can be found at [8, 2]. In the past, there have been some researches about agents and online games[3, 11, 12, 13] but there have been no online games that have been designed following the pattern and ways of a Multi-Agent System using neither Agent Organizations (AOs) [4, 14] nor Electronic Institutions (EIs)[6, 9]. On this article, a proposal for an initial characterization of a MMOG using the Multi-Agent Systems paradigm is presented. In section 2, there is a brief outline of the topology of a MMOG. In section 3, the basic model of the proposal is presented. In section 4, the different kinds of software agents that would take part in the proposal are introduced. Finally, in section 5, some conclusions and hints at future work lines are presented.

2 Topology of an Online Game

Most MMOGs share characteristics that make them different from other types of online games. MMOGs host a large number of players in a single game world, and all of those players usually can interact with each other at any given time. As stated earlier, popular MMOGs have hundreds of thousands of players (even millions of them) online at any given time, usually on a company owned server. Also, MMOGs usually do not have any significant user-made modifications to the game since the game must work on company servers. Despite technology and content constraints, most MMOGs can fit up to a few thousand players on a single game server at a time and they need large-scale game worlds and servers to connect players to those worlds.

2.1 Organizing Players and Zones

Players of MMOGs tend to organize themselves spontaneously in groups normally called “*clans*” or “*guilds*”, which normally maintain a hierarchical structure based on the *seniority* of the members and/or worthiness within the semantics of the game. The bonds that unite the members of a clan can be of most diverse natures: previous relations, same country or language, or simply common goals. Clan members usually play the game together (at the same time and in the same virtual space). Clans, as a unit, usually have short and long term agendas that are expressed in goals that have a meaning within the semantic boundaries of the game (e.g. a clan wants to invade a small town. First, they recruit new members and train them. Second, they strike a raid towards the town). Clans are *weak* organizations: they do not have a persistent leadership (their leaders may change at any time) and do not hold a strong layered hierarchy. There are usually two kinds of memberships: regular members and clan leaders. Regular members belong to the clan, participate in clan activities and, normally, have a share in the benefits that the clan (as a whole) earns. Clan leaders dictate the strategies of the clan; they are at the head of clan activities and are usually the ones that share out the clan benefits.

Whatever the structure of a game world is, in the end it normally ends up divided into segmented virtual *game zones*. Whether these zones appear visually segmented to the player or not is irrelevant, as the game interface can present them visually continuous if desired. The differentiation of game zones usually carries the differentiation of the visual style and graphical representation of each zone and, more importantly, differences in how these zones *work* or are played.

2.2 The Game Software Elements

MMOGs are software architectures split into several game elements. In order to *agentify* these architectures, the game elements must be identified and classified first. On one hand, there is the interface the human player uses to play the game, what is normally called the *client software*. This application connects to the virtual game world and allows the player to become the avatar and actually play the game. This element is out of the scope of this article and will be covered in future works. The other game element would be the *server software*, which is where the actual virtual world is. But, since this is a huge element, it will be split into several ones.

- First, there are the *avatars*, the representations of the players in the virtual game world, both human and AI. Avatars respond to a certain look and feel based on the general visual design of a particular game. For instance, in a racing game, avatars could be cars and motorcycles, but in a medieval game, avatars may be knights or princesses. The most important feature of avatars is that there is a collection of quantifiable factors that define the capabilities of each avatar within the context of the game. These are called the *stats* of the avatar, and they usually mark how *skilled* is the avatar in performing several types of virtual tasks. For instance, in a racing game, a car avatar may have some stats called '*Maximum Speed*', '*Grip*' or '*Gas Consumption*' which may hold quantifiable values.
- Another element of the game is the *virtual environment* itself, which has been already described in section 2.1. All in all, the environment tries to be a world of its own, a world that is simulated and controlled by a computer program.
- There are also the *game rules*, the internal logics of the game and the context where the stats of the avatars make sense. It is where the genre of the game is usually defined. These rules may be as simple or as complex as the designers of the game desire, and all the avatars within the game *do* follow them.

3 Agents, Organizations and Institutions: The MAS Approach

The technology of software agents and Multi-Agent Systems (MAS) is a well-known computational paradigm based in the interaction of autonomous and independent functional units (the agents). However, the same distributed and somehow *free* nature of MAS makes them prone to *chaos*. How can agents be organized in a structured way? This concern is especially true in open environments, where agents external to the core system can join the platform at any time. There is an approach to solve this issue: AOs, as well as more strict, norm-driven offspring of it called EIs.

In essence, the characterization of an online game as a MAS can structurally be seen as a layer in the design of a whole game (in addition to developing game assets, art design, music composition, programming, etc...), which goes *on top* of the basic layer of access to the medium (the virtual environment), as can be seen in the works of Barella et al. in the realm of intelligent virtual environments[5]. For the rest of the work, it is assumed that the game system includes a FIPA-compliant[10] *agent platform* which supports AOs and EIs to their full extent. This platform has a standard Agent Management System, which identifies and keeps track of the agents within the platform, and a Directory Facilitator, which does the same with services.

3.1 Game Zones as Electronic Institutions

As section 4 will explain, there are some agents whose task is to manage and represent the environment of the game. However, in most situations the environment may be highly dynamic and large and it can be a problem to instantiate many of these type of agents to manage everything. In the end one could end having game servers *overcrowded* with these kind of agents with problems to coordinate themselves. This is where the EIs come in. The basic idea is to model the environment of the game with a series of federated EIs. As stated earlier, EIs are controlled environments where it is

mandatory that participant agents follow a strict set of rules and norms, and where each agent may play a different role in different times and virtual places depending on the semantics of the institution. So, in the end, the game environment can be seen as a series of interconnected EIs that the game agents *dwell*, the exit points of some tied to the entry points of others and vice-versa.

One of the main advantages of using EIs to represent game zones is that they can be set to allow entrance only to agents that fulfil some requirements (which are established by the game designers). Every time an agent is going to enter into an EI, it has to pass a test that decides whether it meets the criteria in order to enter the institution or not. EIs within the game may apply restrictions not only to individual agents that wish to enter them, but to whole clans. When a player clan wishes to enter an EI as a clan activity, there may be requirements that may be fulfilled not by individual members of the clan, but by the clan *itself*. For instance, certain institutions may allow entrance only to player clans with more than certain amount of members, or only to player clans which have a clan goal that can only be fulfilled inside the institution.

3.2 Player Clans as Organizations

As explained in section 2.1, players of online games usually organize themselves in clans. It is desirable that the game software itself helps players form and manage clans, specially the part where clan leaders specify the norm of the clan, the clan goals (as a group) and which is the policy of sharing the benefits obtained through clan activities. AOs are quite suited for the task of player clans. They already have a possible hierarchical structure (such as the one in the ‘*Hierarchy*’ class), enforcement of internal norms and establishment of goals, policies and even penalties and ban options for malicious members. And so the proposal is to implement player clans as AOs which are formed with agents representing the player members.

Let’s think of a group of players which wish to form a clan in order to accomplish a game task (e.g. in a fantasy game, defeat a powerful AI-controlled enemy and collect the treasures it guards). While in the game, the interface should show the players some kind of option to form a clan. Once in it, the players (now the clan *founders* or *lords*) can customize the clan features: the clan name, the clan internal structure (is it anarchic? is there some kind of hierarchy or leadership? etc...), which are the starting members, clan policies for membership (is it an invite-only clan? is it open to any player? etc...), some kind of visual marks for the clan, the clan short-term and long-term goals (e.g. to defeat such enemy), the policy for sharing the benefits (e.g. 1% of the loot goes for every member), and some other factors. Once the clan is set up, it becomes active within the game and clan members can start acting as such. Internally, what happens is that a new AO is spawned and configured with the options established by the clan lords, which in turn become the *supervisors* of that AO. Then, all the AvatarAgents of the members gain membership of the AO. Whenever an organizational event happens (i.e. an internal norm is broken or a goal is achieved), the system automatically applies the policies of the AO for that case, and the supervisors need not to manually apply or approve these actions if they wish. In the end, the versatility and customization of AOs allows to build such player clans on top of them.

4 Game Agents

And so, once the game elements have been identified, the process of characterization continues by defining and characterizing the type of software agents that take part in the system. Each one of these types roughly identifies with at least one of the game elements presented previously, and can be seen as “*templates*” for other, more specialized agents, to appear in further iterations of this work. Let’s see them.

4.1 InterfaceAgent

InterfaceAgent stands for “*Interface Agent*” and it is the agent rough equivalent to the client software. It is an agent that is executed in the *client* machine, that is, the player’s computer. As such, it is out of the scope of the present article and will be covered in future works. However, assume that one of the purposes of this agent is to generate action commands to the player’s avatar that will be forwarded to the game system, and to receive responses and other notifications from the game system. The InterfaceAgent interacts solely with another kind of agent: the ProfileAgent.

4.2 ProfileAgent

ProfileAgent stands for “*Profile Agent*” and it is the agent that manages a player’s status and profile within the game community. A ProfileAgent is executed server-side, that is, inside the agent platform (which makes it an *internal* agent). It manages the player’s preferences in the game world, which avatars the player uses, the *role* that the player plays in the system (i.e. the permissions that the player has towards the game system), and the player’s experience in the game community (not the avatars’). It also holds the attention profile for the player. These profile information can be used internally, within the context of the game system and community, or externally. It can be *exported* as attention information for other games or services to use, much like an APML[1] profile on modern web services. A ProfileAgent basically controls the limits of what a user can do within the context of the game, and it does so by means of the *role* it plays. A ProfileAgent can play three different roles, according to the role its user plays:

- **Spectator:** the user is someone which can only observe how other people play it. It cannot control nor register any avatar. However, it may be able to communicate with other users.
- **Player:** the user is a normal player of the game with no special permissions. The player may control as many AvatarAgents as the game allows (normally just one) and play as normal.
- **GameMaster:** it means that the user is a special supervisor or administrator of the game that has special privileges over the game system (e.g. it is not bound to the same game rules as normal users).

ProfileAgents communicate mainly with AvatarAgents, but can also communicate with other ProfileAgents (for instance, whenever a user wishes to send a human-readable message to another user, it goes directly from one ProfileAgent to the other).

When the `InterfaceAgent` generates game actions, they are received by the `ProfileAgent`, which in turn forwards them to the appropriate `AvatarAgent`.

4.3 AvatarAgent

AvatarAgent stands for “*Avatar Agent*” and is another *internal* agent. It is a persistent kind of agent (is not deleted and re-spawned often, it bears a long life cycle) and it represents a Player Character (or PC) avatar within the game. It is the agent that holds the PC *stats* (server-side), and so, as they are not stored in the client machine, a malicious user cannot modify them locally (this prevents *cheating* or *hacking* the game).

`AvatarAgents` hold the visual representation of the avatar according with the style and design of the game. This representation, usually, highly customizable by the user controlling the avatar, is the one that the other users get to visualize in their interfaces. The representation is not unique: there can be multiple representations for multiple kinds of interfaces (e.g. one in 2D and other in 3D; or one with high visual detail and other with low visual detail). The `AvatarAgent` is the kind of agent that actually performs the actions for the player in the virtual world. It is a persona of the player inside the virtual world and it is a very active type of agent (i.e. it generates a lot of communication). Since it is an internal agent of the game and it is designed by the game developers, it will strictly follow the game rules. It does not matter whether the player *orders* it to do something that goes against the rules of the game (e.g. in a racing game, drive over water; or in a game like chess, move the knight only forward), the `AvatarAgent` knows the rules and it will not break them, much like a `Governor` agent does not break the rules of an EI[9]. This quality of the `AvatarAgents` removes the need of making constant *sanity checks* on the integrity of the game system and also removes the need of having some kind of *guardian agents* that enforce the rules of the game. The `AvatarAgents` do not break the rules at all, so there is no need for this kind of enforcers. Another good outcome of the faithful and truthfulness of the `AvatarAgents` is the solving of in-game conflicts, which in the majority of MMOGs is some kind of virtual combat among avatars. Let’s imagine a situation where two `AvatarAgents` engage in a combat. The game has its combat rules set for these occasions and they are based on the stats of the avatars. Since both agents know the rules and are truthful to each other, there is no need for an external *referee agent* that mediates in conflicts. The `AvatarAgents` alone can play and solve the situation rules-wise.

`AvatarAgents` interact with almost all the other kind of agents (except for the `InterfaceAgent`).

4.4 NPCAvatarAgent

NPCAvatarAgent stands for *Non-Player Character Avatar Agent* and it is very similar to the `AvatarAgent`. The main difference between both kinds of agent is that the `NPCAvatarAgent` is *not* controlled by the actions of a player at all. It is an autonomous kind of agent that plays a role within the context and semantics of the game.

Some `NPCAvatarAgents` can also offer services to other agents. For instance, let’s say the developers want to model a virtual shop in a game. They could set a `NPCAvatarAgent` to represent the shopkeeper that advertises the service of “Selling” in the directory facilitator of the platform. Then, the agents that would like to know how to

buy items (AvatarAgents mostly) would know that this particular agent offers that service and would engage in a trading protocol. As AvatarAgents, NPCAvatarAgents are also truthful and faithful to the game rules, so everything stated about the enforcing of game rules and solving conflicts stays the same for them. NPCAvatarAgents interact mostly with AvatarAgents and other NPCAvatarAgents.

4.5 ZoneMasterAgent

ZoneMasterAgent stands for “*Zone Master Agent*” is the kind of agent that implements the internal logics of the environment and the game in a virtual game zone. There can be many of its kind and each one manages a particular virtual game zone. A *ZoneMasterAgent* manages all the game events and services (usually offered to the AvatarAgents and NPCAvatarAgents) of its virtual zone. It also manages the visual representations of its zone that players end up visualizing in their interfaces.

It could be said that *ZoneMasterAgents* manage all the active elements of the virtual game world that are not already covered with the AvatarAgents and NPCAvatarAgents. For instance, in a racing game where players control some cars and NPCAvatarAgents control other cars and pedestrians, the developers want to include a gas station where both NPCAvatarAgents and AvatarAgents can access services like “Buy Gas”, “Wash Car” and the like. This station could be modelled using one or more *ZoneMasterAgents* inside an EI that manages the virtual zone of the station.

4.6 BankAgent

BankAgent stands for *Bank Agent* and it is the kind of agent that stores permanently (i.e. backs up) the information of the game using a database back end wrapped into a service for all the other agents in the system. This agent falls out of the scope of this article and will be explained in future works.

5 Conclusions and Future Work

In this work it has been presented the first step towards the full characterization of Multiplayer Online Games as Multi-Agent Systems. Using modern mechanisms from agent theory, like EIs and AOs, the level of complexity of designing and developing a modern Multiplayer Online Game can be overcome. There is work to do in the future in this line of research. As hinted in some of the previous sections, there is a lot of potential in using services offered by the platform and by some agents in order to implement some game elements, like in-game commerce. These elements need to be identified and represented in agent service terms, along with a common ontology for game agents to use and access these services.

Also, colourful and instructive examples of this characterization must be provided in order to ease its comprehension and expand its adoption, along with an initial prototype using modern agent platforms that support all the features needed by these systems (specially AOs and EIs), which is already underway.

Acknowledgements

This work is partially supported by the PAID-06-07/3191, TIN2006-14630-C03-01 projects and CONSOLIDERINGENIO 2010 under grant CSD2007-00022.

References

1. APLM: Attention Profiling Mark-Up Language, <http://www.apml.org/developers/spec/>
2. An Analysis of MMOG Subscription Growth (2008), <http://www.mmogchart.com>
3. Aranda, G., Carrascosa, C., Botti, V.: Intelligent agents in serious games. In: Fifth European Workshop On Multi-Agent Systems (EUMAS 2007). Association Tunisienne d'Intelligence Artificielle (2007)
4. Argente, E., Palanca, J., Aranda, G., Julian, V., Botti, V., Garcia-Fornes, A., Espinosa, A.: Supporting agent organizations. In: Burkhard, H.-D., Lindemann, G., Verbrugge, R., Varga, L.Z. (eds.) CEEMAS 2007. LNCS (LNAI), vol. 4696, pp. 236–245. Springer, Heidelberg (2007)
5. 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)
6. Bogdanovych, A., Berger, H., Simoff, S., Sierra, C.: Narrowing the Gap between Humans and Agents in E-commerce: 3D Electronic Institutions. In: Bauknecht, K., Pröll, B., Werthner, H. (eds.) E-Commerce and Web Technologies, Proceedings of the 6th International Conference, EC-Web, pp. 128–137 (2005)
7. Cuni, G., Esteva, M., Garcia, P., Puertas, E., Sierra, C., Solchaga, T.: MASFIT: Multi-Agent System for Fish Trading. In: ECAI, vol. 16, p. 710 (2004)
8. Ducheneaut, N., Yee, N., Nickell, E., Moore, R.J.: Alone together?: exploring the social dynamics of massively multiplayer online games. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 407–416 (2006)
9. Esteva, M., Rodriguez, J.A., Sierra, C., Garcia, P., Arcos, J.L.: On the formal specification of electronic institutions. In: Agent Mediated Electronic Commerce, 1991, pp. 126–147 (2001)
10. FIPA, Abstract architecture specification, Technical Report SC00001L (2002)
11. Laird, J., van Lent, M.: Human-Level AIs Killer Application. AI Magazine, 15–25 (summer, 2001)
12. Nareyek, A.: Intelligent Agents for Computer Games. In: Marsland, T., Frank, I. (eds.) CG 2001. LNCS, vol. 2063, pp. 414–422. Springer, Heidelberg (2002)
13. Niederberger, C., Gross, M.H.: Towards a Game Agent, Institute of Visual Computing, ETH Zürich, Technical Report 377 (2002)
14. Zambonelli, F., Jennings, N., Wooldridge, M.: Developing multiagent systems: The Gaia methodology. ACM Transactions on Software Engineering and Methodology 12, 317–370 (2003)

A Dialogue Game Protocol for Recommendation in Social Networks*

Stella Heras, Miguel Rebollo, and Vicente Julián

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n. 46022 Valencia, Spain
Tel. (+34) 96 387 73 50
{sheras,mrebollo,vinglada}@dsic.upv.es

Abstract. Social networks and recommender systems that advise items between their members have spread across the Internet. These recommenders must take into account the distributed and dynamic nature of the knowledge shared in social networks. Opinions and preferences can change and thus, allowing agents to argue about recommendations could improve their quality. This paper proposes the use of dialogue games as mechanism for modelling the dialogue between users that share their recommendations in a social network. The paper specifies the interaction protocol, defines some metrics to evaluate recommendations and finally, offers a brief discussion about decision policies.

1 Motivation

The evolution of the Internet has given rise to the growing of social networks as social structures that relate nodes (i.e. individuals or organizations) via ties representing different types of interdependence (e.g. friendship, conflict or trade). By using these networks, people exchange huge amounts of information with their *contacts*. Therefore, there is a need for some filtering mechanism that helps people to decide which information fits their preferences the most. As social networks spread across the Web, *recommender systems* that advise items between their members do it as well [5]. Multi-Agent Systems (MAS) have also echoed this trend and over the last years, many MAS featuring social networks of personal agents that interact with different purposes have been developed [4]. However, these and most approaches are focused on developing models of trust and reputation in social networks, but our research has been addressed to control the interaction between their members by taking advantage of social network analysis and the dialogical context of these networks.

Commonly, recommender systems set a central entity that *knows* the preferences of the network users and advises them items that are similar to others that interested the user in the past or that similar users tend to prefer. However, in most cases such knowledge is distributed across the network and the existence of a central recommender

* This work was partially supported by CONSOLIDER-INGENIO 2010 under grant CSD2007-00022 and by the Spanish government and FEDER funds under TIN2005-03395 and TIN2006-14630-C0301 projects.

that has unlimited access to information is not realistic. Since each user keeps some relation with their contacts in the social network, he/she could better know their preferences and act as a more appropriate recommender for them.

In addition, the high dynamicity of social networks should also be considered when recommending items to their members. Opinions and preferences can change over time. A mechanism that allows users to argue about recommendations would help recommenders to justify their recommendations and to receive some feedback to provide more suitable recommendations in the future. Moreover, such mechanism would also allow users to reason about the recommendations that they receive and take decisions over them (e.g. select the most useful or the one that comes from the agent that has more friends on the network or that better argues).

Our approach uses a well-known concept of argumentation theory, *dialogue games*, as a suitable mechanism for modelling the dialogue between users that share their recommendations in a social network. Dialogue games are interactions between several players where each player moves by advancing locutions in accordance to a set of rules. Recently, they have been successfully applied in MAS as a tool for specifying communication protocols between agents [2] and for evaluating reputation [1]. The dialogical nature of recommender systems, where people argue about their preferences and interests, makes dialogue games highly suitable for modelling them.

In this paper, we consider a distributed recommendation system that makes use of a social network to reach to potential recommenders. Furthermore, social network analysis allows agents to extract relevant information that helps them to make a decision over the set of available recommendations. The rest of the paper is structured as follows: section 2 introduces the main concepts of argumentation theory applied in this paper; section 3 shows the dialogue game-based model for recommendation in social networks; and finally, section 4 summarises the paper.

2 Argumentation Schemes and Dialogue Games

As pointed out before, the purpose of this paper is to define a dialogue game protocol that controls the recommendation process among a set of personal agents related via a social network. Concretely, we have built our work on a specific instantiation of a general dialogue game, the *Argumentation Scheme Dialogue (ASD)* [3]. This game extends traditional dialogue games to take into account the dialogical nature of certain stereotyped patterns of common reasoning called *argumentation schemes*. This feature allows us to model the interaction between different users by using a well-known argumentation scheme that captures the way in which people reasons about experts' opinion, the *Argument from Expert Opinion* [6]. That pattern perfectly represents the reasoning process that people follow when are faced with the problem to evaluate recommendations (i.e. opinions) of other individuals (with some degree of expertise) in any domain. Its general structure is the following:

- *Major Premise*: Source E is an expert in field F containing proposition A .
- *Minor Premise*: E asserts that proposition A (in field F) is true (false).
- *Conclusion*: A may plausibly be taken to be true (false).

In addition, the scheme features a set of *critical questions* that specify the way in which the conclusion of the scheme can be rebutted. The instantiation of any critical question by an individual represents a potential attack to the main claim supported by this pattern of reasoning. The six basic critical questions are the following:

1. *Expertise Question*: How credible is E as an expert source?
2. *Field Question*: Is E an expert in the field F that A is in?
3. *Opinion Question*: What did E assert that implies A ?
4. *Trustworthiness Question*: Is E personally reliable as a source?
5. *Consistency Question*: Is A consistent with what other experts assert?
6. *Backup Evidence Question*: Is E 's assertion based on evidence?

The argumentation scheme *Argument from Expert Opinion* shifts the burden of proof from a proponent to a respondent in a dialogue. By using this scheme as the reasoning pattern that defines the possible dialogical movements in our dialogue game protocol, the interaction between the members of a social network for item recommendation can be controlled.

3 A Dialogue Game Model for Recommendation in Social Networks

We consider a scenario in which a set A of n personal agents $\{a_1, a_2, \dots, a_n\}$ represents people that are connected in a *social network* for item recommendation. Let us consider, for example, the specific case of a social network of *friends* that share their suggestions about restaurants that they know. The *friendship relation* is represented by an edge between an agent and a friend (i.e. a neighbour agent) that have had a previous interaction. Each agent has a preference profile $PP = \{p_1, p_2, \dots, p_k\}$ that shows its preferences over the k features that characterise the items of the domain (e.g. types of restaurants or food). We assume that agents specify its preferences at the beginning of each recommendation process and they do not change during it. In addition, agents have a knowledge database where they store reviews about the items that they know. Therefore, the knowledge is distributed across the network and agents only have a partial view of it (which fits the assumptions of the algorithms used for social network analysis). In that context, agents can contact to their neighbours and ask for recommendations about a specific item (e.g. a restaurant) that fits their preferences.

Furthermore, at the end of each recommendation process, the initiator agent gives feedback to the participants. Therefore, the agent informs the neighbour whose recommendation was accepted about the utility degree $u \in \{-1, 0, 1\}$ that was finally assigned to its recommendation (a discrete value -1: inappropriate, 0: useless or 1: useful). Similarly, the rest of participants are given a neutral utility degree (i.e. 0: useless), since their recommendations were not finally accepted.

To evaluate recommendations, we have defined two metrics based on social network analysis. On one hand, the edges of the social network are directed and weighted with a *local utility measure* $U_{ij} \in [-1, 1]$ that measures the success of the recommendations that an agent receives. The weights are calculated by using the formula (1), where $u_{j(k)}$ is the utility degree of the recommendation k that an agent a_i accepted from the neighbour a_j and K is the total number of recommendations received

from a_j . Values close to 1 show more appropriate recommendations. On the other hand, the *expertise degree* $e_j \in [-1, 1]$ of an agent as recommender is calculated by the formula (2), where U_{ij} is the local utility for each agent a_i of the recommendations received from a_j and $\text{deg}^+(a_j)$ is the *centrality indegree* of the agent a_j . Values close to 1 represent expert recommenders.

$$U_{ij} = \frac{\sum_{k=1}^K u_{j(k)}}{K} \quad (1)$$

$$e_j = \frac{\sum_{i=1}^I U_{ij}}{\text{deg}^+(a_j)} \quad (2)$$

In the current specification of the model, we assume *cautious agents* that only consider the recommendations of their neighbours (i.e. level 1 recommendations). The initiator agent is in charge of controlling the course of the dialogue, asking neighbours for recommendations, evaluating them and taking a final decision. Neighbours only have to provide the agent with recommendations, to justify them when required and to rebut, if possible, different recommendations offered by other agents.

Finally, each agent is assumed to have its individual reasoning mechanism to propose recommendations in view of their knowledge and the similarity of preference profiles. Moreover, agents must also have a set of rules of inference of propositional logic and the *Argument from Expert Opinion* scheme to create arguments. The definition of the reasoning mechanism that agents use for proposing recommendations and the way by which agents create arguments are out of the scope of this paper. Here we focus on the evaluation of arguments on the basis of the metrics proposed and on the definition of the interaction protocol that controls the recommendation dialogue. In what follows, the dialogue game of the model is described.

3.1 Dialogue Game Protocol

We have made some assumptions and simplifications over the dialogue game ASD and the *Argument from Expert Opinion* scheme that are appropriate in our domain. On one hand, our model assumes the existence of:

- A finite set of players, denoted *Agents*, consisting of the agent that initiates the recommendation dialogue (i.e. *Initiator*) and its neighbours, with elements $a_1 \dots a_n$.
- A finite set of items to recommend (e.g. restaurants), denoted *Objects*, with elements $\{o_1 \dots o_m\}$. Agents can only recommend items that they personally know.
- A finite set of features that characterise the objects in the domain where the system operates (e.g. type of cuisine or price), denoted *Features*, with elements $\{f_1 \dots f_l\}$.
- A finite set of values for each feature (e.g. type of restaurant = {Italian, Japanese, Spanish, etc.}), denoted *Values*, with elements $\{v_1 \dots v_p\}$.
- A finite set of discrete values that represent the feedback that an agent offers about the utility of the recommendation received, denoted *utility*, with elements $\{-1: \text{inappropriate}, 0: \text{useless or } 1: \text{useful}\}$.
- A finite set of variables that represent agent's preferences when it asks for a recommendation, denoted *Preferences*, with elements $\{p_1 \dots p_q\}$.
- A function *preference*: $P \rightarrow F \times V$, which maps each *Preference* of an agent to a pair $\langle f, v \rangle$, where $f \in \text{Features}$ and $v \in \text{Values}$.

- A function *review*: $Object \rightarrow 2^{F \times V}$, which maps each *Object* that an agent personally knows to a vector of pairs [$\langle f_1, v_1 \rangle \dots \langle f_r, v_r \rangle$], where $f \in Features$ and $v \in Values$.
- A function *feedback*: $Object \rightarrow utility$, which maps an *Object* that an agent has recommended to one of the possible values of *utility*.

On the other hand, the model includes some variations in the original ASD dialogue game:

- Walton’s *Strategic Rules* have been reformulated as *Commencement Rules* and *Termination Rules* of the McBurney and Parsons’ style.
- For the sake of simplicity, we assume that the fact that an agent asserts a recommendation or an argument for justifying or rebutting a recommendation commits it to having a reason (inferred from its knowledge database or its recommendation record) for doing so. Therefore, agents only have propositional commitments either to the last recommendation that they have proposed or accepted or to the assumptions that it implies. Those commitments are stored in a record D that represents the current state of the argumentation dialogue and emulates the *commitment stores* of the agents:

$$D^i = (\langle o_1, \alpha_1 \rangle, \langle o_2, \alpha_2 \rangle, \dots, \langle o_n, \alpha_n \rangle, d^i)$$

where D^i stores the objects o_j recommended by each neighbour a_j , the arguments α_j that support each recommendation and the partial decision of the initiator to accept $d^i \in \{o_1, \dots, o_n\}$ as the best recommendation at the moment i . Arguments in our model can be information that justifies the matching between the features of an object and the preferences of the initiator, or values of utility or expertise that try to stand out the recommendation of a neighbour from the rest.

Furthermore, we have specified the interaction protocol as a formal dialogue game with the components identified by McBurney and Parsons [2]:

- *Commencement Rules*

The dialogue starts when an agent is looking for a recommendation. At that moment, the initiator of the dialogue uses the social network to contact their neighbours and send them its preference profile. The *preference* function is used here to specify the values for, possibly some, features that characterise the object to recommend. Then, each neighbour uses its own reasoning mechanism to match the preferences with the reviews about the objects that it knows and proposes its recommendation. Note that neighbours can decide to engage or not in the dialogue.

- *Locutions*

- Statements: are the permissible locutions and their compounds: *propose(object)*, *accept*, *reject*, and *assert(data)*.
- Withdrawals: *noCommitment(object)* is the locution for the retraction of a specific recommendation when the proponent agent changes its mind.
- Questions: *propose(object)?* is the locution for asking neighbours for recommendation proposals. In addition, *assert(data)?* is used for getting more information from an agent (e.g. asking it for its opinion about an unspecified preference). Note that agents are not committed to answering questions.

- Critical Attacks: the locution *pose(CQ)* poses a critical question associated with the *Argument from Expert Opinion*. In our model, we assume: (a) that agents have some degree of expertise as recommenders (*field question*); (b) that recommenders always propose the recommendation that, from its point of view, fits the profile of the initiation the most (*opinion question*) and (c) that agents are honest and their assertions are based on evidence (*backup evidence question*). Therefore, possible attacks to arguments are: (i) to question the degree of expertise of the proponent (*expertise question*); (ii) to demonstrate that the proponent is not personally reliable as source (*trustworthiness question*) or (iii) to demonstrate that the proponent's recommendation is not consistent with the one of other expert with at least equal or greater degree of expertise (*consistency question*). Note that the burden of proof in the case of the trustworthiness and consistency critical questions falls on the agent who asks them and therefore, the initiator must provide the proponent with some data that justifies those criticisms.
 - Challenges: *why(locution)?*, where *locution* can be a recommendation proposal or a critical attack, is the locution to request arguments that support them.
- *Commitment Rules*
 - The assertion of the locution *propose(object)* includes the recommended object in the dialogue record and commits the proponent to the assumptions of the *Argument from Expert Opinion* scheme.
 - The assertion of the locution *noCommitment(object)*, deletes the recommendation made by the proponent from the dialogue record.
 - The assertion of the locution *why(propose(object))?* commits the proponent of the recommendation either to providing an argument for supporting it or withdrawing the recommendation.
 - The assertion of the locution *why(pose(CQ))?* commits the initiator to providing data that justifies the critical attack.
 - *Dialogue Rules*
 1. The recommendation process consists of a set of parallel dialogues between the initiator agent and its neighbours. At each step of the dialogue, the initiator agent advances (or receives) a locution to (from) one of its neighbours. The initiator has the whole control of the dialogue and neighbours do not speak directly between them. However, the information that a neighbour provided can be used by the initiator when speaking with other neighbours.
 2. The locution *propose(object)?* opens the recommendation process and must be followed by a request for more information about preferences (i.e. *assert(data)?*), a recommendation proposal (i.e. *propose(object)*) or a rejection statement (i.e. *reject*).
 3. After the opening, neighbour agents can ask for more data (i.e. *assert(data)?*) while the initiator accedes to provide them with it (i.e. *assert(data)*). The process ends when the neighbour makes a recommendation (i.e. *propose(object)*) or when it rejects the proposal (i.e. *reject*). In addition, note that the initiator can also reject providing neighbours with more information.
 4. The locution *propose(object)* starts the recommendation dialogue and can be challenged by a request for an argument that supports the recommendation

(i.e. *why(propose(object))?*), or followed by the acceptance (i.e. *accept*) or the rejection of the recommendation (i.e. *reject*).

5. In case a recommendation is challenged (i.e. *why(propose(object))?*), the challenge must be followed by the locution *assert(data)*, where data includes an argument that supports the recommendation (e.g. the proponent's review of the object) or by the locution *noCommitment(object)*.
6. The locution *assert(data)* in response to a challenge can be followed by the acceptance of the recommendation (i.e. *accept*), its rejection (i.e. *reject*) or a critical attack associated with the *Argument from Expert Opinion*. Possible moves are: (i) *pose(expertise)*, (ii) *pose(trustworthiness)* or *pose(consistency)*.
7. Critical attacks can be answered by the locution *assert(data)*, where data tries to rebut the attack, or by the locution *noCommitment(object)*.
8. *Trustworthiness* or *consistency* critical attacks can also be challenged (i.e. *why(pose(CQ))?*). In that case, they must be followed by the locution *assert(data)*, where the initiator provides the neighbour with data that identifies an agent with higher utility value of its recommendations (more trustworthy) or with higher degree of expertise (more consistent) that has proposed a different recommendation, or by the locution *noCommitment(object)*.

- *Termination Rules*

The initiator agent has the entire control of the recommendation protocol. Neighbour agents cannot change their recommendations during the dialogue, but only propose objects, answer challenges and critical attacks or withdrawn their recommendations. Therefore, the game ends when the initiator has taken a decision over the set of offered recommendations and their proponents are informed by using the *accept/reject* locutions. However, after a maximum time is exceeded, the last recommendation accepted is taken as the final decision.

3.2 Decision-Making Policies

The dialogue rules defined in the previous section specify the legal movements that the initiator agent can make for gathering information from its neighbours and taking a decision about their recommendations. However, it can follow different decision policies to make a choice between recommendations. These policies determine the decision-making algorithms for each locution that it sends. Similarly, each neighbour can follow its own decision policies to answer the initiator with specific locutions.

Basic decision-making policies could be *expertise-based policies*, where the initiator would send the *accept* locution to the recommendation of that neighbour with higher degree of expertise or *utility-based policies*, where the initiator would send the *accept* locution to the recommendation of the neighbour that tends to offer more useful recommendations. In both, for instance, the initiator *i* could make a random choice in case of draw. However, more elaborated policies could take advantage of the dialogical possibilities of the model proposed. Let us suppose that there are 3 neighbours engaged in the recommendation process. Following a *utility-based policy*, the dialogue record *D* at the step *t* of the recommendation process could look as follows: $D^t = (\langle o_1, U_{i1}=1 \rangle, \langle o_2, U_{i2}=0.5 \rangle, \langle o_3, U_{i3}=1 \rangle, d^t = o_1)$.

However, let us suppose that the neighbour 3 would have a negative utility measure of its interactions with neighbour 1 (i.e. having received inappropriate

recommendations). Following a more elaborated policy that allows agents to argue about recommendations, the neighbour 3 could try to persuade the initiator to accept its recommendation. Then, after n steps where the initiator interchanges with the neighbour 3 the necessary locutions to pose a *trustworthy* critical attack, the dialogue record of the process could change to: $D^{t+n} = (\langle o_1, U_{i1}=1 \rangle, \langle o_2, U_{i2}=0.5 \rangle, \langle o_3, U_{3j}=-1 \rangle, d^{t+n} = o_3)$. Here, the neighbour 3 rebuts the attack by showing the negative utility of the recommendations that it has received from the neighbour 1. At this point of the dialogue, the neighbour 1 could try to provide other argument to justify its recommendation and rebut the other's, and so forth.

4 Conclusions

In this paper we have presented a dialogue game protocol based on the ASD game for controlling the recommendation process in a social network. In addition, two metrics of social network analysis have been proposed to evaluate recommendations. Moreover, a brief discussion about decision making policies is also presented.

At the moment, neighbour agents cannot propose different recommendations during the dialogue. Furthermore, the preferences of the initiator cannot be modified. Future work will extend the model to consider more dynamicity in the dialogue.

For the sake of simplicity, we assume that the agents' profile shows their preferences about the item under discussion (i.e. restaurants in the example), but it can be extended to consider a set of profiles representing the preferences of the agents about the various items that can be advised in the network. Other assumptions, as the pre-determined classification of items under a set of specific categories that the system previously knows will be also relaxed, allowing agents to argue about such decisions. In addition, further research must be done to extend the model to deal with level 2 (or higher) recommendations (i.e. those of the neighbours of an agent's neighbours).

References

1. Bentahar, J., Meyer, J.-J., Moulin, B.: Securing Agent-Oriented Systems: An Argumentation and Reputation-based Approach. In: Proc. of INTG 2007, pp. 507–515 (2007)
2. McBurney, P., Parsons, S.: Games That Agents Play: A Formal Framework for dialogues between Autonomous Agents. *The Journal of Logic, Language and Information* 11(2), 315–334 (2002)
3. Reed, C., Walton, D.: Argumentation Schemes in Dialogue. In: *Dissensus and the Search for Common Ground*, CD-ROM, pp. 1–11. OSSA, Windsor (2007)
4. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: Proc. of IJCAI 2002, vol. 1, pp. 475–482 (2002)
5. Walter, F.E., Battiston, S., Schweitzer, F.: A model of a trust-based recommendation system on a social network. *The Journal of Autonomous Agents and Multi-Agent Systems* 16(1), 57–74 (2008)
6. Walton, D.: *Appeal to Expert Opinion*. Penn State University Press (1997)

Friends Forever: Social Relationships with a Fuzzy Agent-Based Model

Samer Hassan^{1,2}, Mauricio Salgado², and Juan Pavon¹

¹ GRASIA: Grupo de Agentes Software, Ingeniería y Aplicaciones, Departamento de Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid, Madrid, 28040, Spain

{samer, jpavon}@fdi.ucm.es

² CRESS: Centre for Research in Social Simulation, Department of Sociology, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom
m.salgado@surrey.ac.uk

Abstract. Sociological research shows that friendship and partner choice tend to reveal a bias toward social similarity. These relations are ruled by the so called “proximity principle” which states that the more similar two individuals are, the more likely they will become friends. However, proximity, similarity or friendship are concepts with blurred edges and grades of membership (acquaintances, friends, couples). Therefore, in order to model the friendship dynamics we work on an Agent-Based Model that already manages the social relationships, together with demographics and evolutionary crossover. To introduce these theoretical concepts we decided to fuzzify the system, explaining the process in detail. Thus, we end up with fuzzy sets and operations, a fuzzy friendship relationship, and a logistic function for its evolution.

Keywords: agent-based modelling, friendship, fuzzy agent, fuzzy logic, social simulation.

1 Introduction

The dynamics of social relationships is a highly complex field to study. Even though it can be found many literature regarding friendship networks, weak links / acquaintances, relationship evolution and so on, we are still far from understanding all the processes involved. Social research has shown that people use numerous criteria when they consider the possibility of turning an acquaintance into a friend. But this paper considers only one: socio-demographical characteristics of people (i.e. ideology, age) that determine the emergence and evolution of friendship. After studying the theory available, we have decided to use the “proximity principle” in order to model the friendship dynamics. This principle assesses that the more similar two individuals are, the stronger their chances of becoming friends. Thus, we attempt to model the processes in which strangers turn to be acquaintances, those turn into friends, and some friends into couples.

In order to do that, we will work with an Agent-Based Model (ABM) which already handles social relationships: Mentat [1], which is deeply described. However, the application of the friendship modelling in the ABM has been accomplished using

fuzzy logic. Therefore, we expose how the theory has been guiding the fuzzification process step by step, resulting in a new ABM called Fezztat. The whole process consisted in the fuzzification of the agent characteristics, the similarity process, the fuzzification of the friendship relationship together with the introduction of an evolution function, and a new couples matchmaking calculation. Comparing the results of the different ABM we might assess that the fuzzy version deals with the problem in a more accurate way.

The section 2 explains some theoretical concepts of friendship dynamics. Section 3 resumes the needed basics of fuzzy logic, while the next one describes the Mentat ABM. Section 5 analyzes the fuzzification process, and the last two parts finish with a discussion about the results obtained and conclusions.

2 Friendship Dynamics

2.1 Understanding Friendship

Selecting a friend is among the most personal of human choices, and thus it is not surprising that friendship groups tend toward social homogeneity. Members of the working class usually associate with other workers, and middle-class individuals generally choose friends who are middle class.

A preliminary step to constructing a friendship modelling is an examination of the way that the social context structures friendship choice. Contextual explanations for individual behaviour argue that (i) individual preferences and actions are influenced through social interaction, and (ii) social interaction is structured by the individual's social characteristics [2]. This is consistent with the important homophily principle in social networks of [3]. Principles of meeting and “mating” by which strangers are converted to acquaintances, acquaintances to friends, and even maybe friends into partner, follow the same rules. Meeting depends on opportunities alone (that is, to be in the same place at the same time); instead, mating depends on both opportunities and attraction. How readily an acquaintance is converted to close friendship depends on how attractive two people find each other and how easily they can get together.

The “proximity principle” indicates that the more similar people are, the more likely they will meet and become friends [4]. Therefore, features like social status, attitudes, beliefs and demographic characteristics (that is, degree of “mutual similarity”) channel individual preferences and they tend to show more bias toward homogeneous friendship choices.

2.2 A Friendship Evolution Function

Similarity, proximity or friendship are vague or blurry categories, because they do not have clear edges. For this reason, we have developed a formal model of friendship dyads using the general framework presented above, but considering similarity and friendship as continuous variables. Besides, because friendship occurs through time, we have considered our model in dynamic terms.

We conceive the friendship process as a search for compatible associates, in terms of the proximity principle, and where strangers are transformed to acquaintances and acquaintances to friends as a continuous process over time.

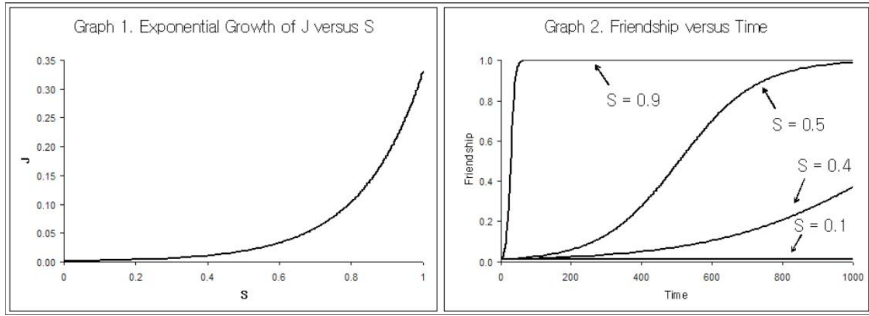


Fig. 1. Graphs showing exponential growth of J and the logistic friendship function

We propose the hypothesis that a logistic function [5] can describe formally the “friendship relation” or degree of friendship for every couple of individuals. The logistic function is one of the most useful (and heavily exploited) modelling strategies used in the social sciences. In order to model the evolution of friendship, we have specified it as in Equation 1:

$$W(t) = \frac{dF}{dt} = F(t) \times K(t) \times r \tag{1}$$

The equation expresses the hypothesis that friendship increases over time; thus, at each point of time, $F(t)$ defines the minimum degree of friendship that is given as an initial condition ($0 < F(t) < K$); K is the maximum degree of friendship that agents can reach (K can be understood as the level of “close friends”), and finally r value defines the growth rate of friendship. However, this equation does not include the “proximity principle” described above. We can include this principle in equation (1) by modifying the growth rate r and stating it as follows: the more similar in social characteristics two individuals are, the higher the growth rate of their friendship is (we need to make r sensitive to the similarity value). Thus, we can express the following equation:

$$r = S \times J \tag{2}$$

Where S is a measure of similarity and J defines a multiplicative factor that increases the magnitude of S within r . The objective of J is turning r more sensitive to S values, and specially sensitive to high S values. For this reason, J describes an exponential growth depending on S values. We can formalize J as follows:

$$J(s) = J_0 \times e^{ps} \tag{3}$$

Where J_0 is the initial value of J , P defines the constant of proportionality and S is the similarity value between the individuals. In the following graphs we can see how the friendship will develop over time given different initial conditions¹.

¹ In the Graph 1 of Figure 1 it is assumed that P is equal to 5.8 and J_0 is equal to 0.001. In Graph 2 it is assumed that K is equal to 1 and F_0 is equal to 0.01; r value is equal to $S \times J(s)$. The constants were generated by experimental procedures.

3 Be Fuzzy, My Friend

3.1 Why, What and When Fuzzy Logic

Individuals are often vague about their beliefs, desires and intentions. They use linguistic categories with blurred edges and gradations of membership, for instance: “acquainted or friend”. Fuzzy logic is oriented at modelling the imprecise modes of reasoning in environment of uncertainty and vagueness [6]. Thus, because vagueness is such a common thing in the social realm, fuzzy logic provides us with a useful way to handle this vagueness systematically and constructively [7].

Fuzzy logic shows that if A is a fuzzy set in a universe of discourse U , then every member of U has a grade of membership in A between 0 and 1 (instead of the classical two-valued-logic). This membership function defines A as a fuzzy subset of U [8]. According to this framework, the mapping of the function is denoted by m_a : $A \rightarrow [0,1]$.

3.2 The Importance of Fuzzy Logic in ABM

There is an increasing interest among social scientists for adding fuzzy logic to the social science toolbox [9]. Likewise, even though it is still incipient, there are numerous examples of researches linking fuzzy logic with social simulation. For instance, in some ABM, agents decide according to fuzzy logic rules; “fuzzy controls” or “fuzzy agents” are expert systems based in “If \rightarrow Then” rules where the premises and conclusions are unclear. Unlike traditional multi-agent models, where these completely determined agents are an over-simplification of real individuals, fuzzy agents take into account the stochastic component of the human behaviour.

Some authors have proposed to improve the agents' strategy choices within the iterated prisoner's dilemma using fuzzy logic decision-rules [10]. Others researchers have claimed that simulation based on two-player games can use fuzzy strategies when analytic solutions do not exist or are computationally very difficult to obtain (because agents use fuzzy strategies; i.e., “If I think my opponent will choose action x , I will choose action y ”) [11]. Examples of multi-agent based models are the fate of spatial dilemmas [12], an extension of sugar space model with fuzzy agents [13] and computational modelling of “fuzzy love and romance” [14].

4 The Case Study: Mentat, the Original ABM

Therefore, to study the friendship dynamics commented, an ABM has been chosen for its fuzzification. The aim of the Mentat model [15, 1] is to understand the evolution of the moral values in Spain from 1980 to 2000. It carries out an analysis of the evolution of multiple factors in the period, trying to determine to which extent the demographic dynamics explains the magnitude of mentality change in Spain. Due to its broad spectrum, it needs to cope with: gender, age, education, economy, political ideology, religiosity, family, friend relationships, matchmaking and reproduction patterns, demographic dynamics, life cycles and others. The aggregated statistics of these variables evolve over time, together with the agent network.

The agents in Mentat are initialised using data from the Spanish census, research studies and sample surveys [16]. Thus, each agent has different values of their attributes, with a behaviour deeply influenced by distributions representing demographic rules (life expectancy, fertility rate, etc).

The simulation has been configured with a population of 3000 static agents, randomly distributed in a space 100x100 (thus, around one agent each 3.3 cells), and simulated for a period of 20 years (1000 agent steps). The agents are able to communicate, establishing friendship and couple relationships, and reproduce. The communication is always local, with a Moore neighbourhood of distance 6 (168 cells). This means that an agent will be able to communicate with around 50 other “possible friends” along its life. From these, each agent will be able to choose, which ones will be its friends. This election will be determined by a probability directly proportional to a similarity measure between the two agents.

It should be mentioned that agents are randomly distributed in the space because the context of an individual can be initially considered random: in which neighbourhood you grew, in which university did you study. Only with the common context already chosen, the individuals can relate with each other taking into account non-random factors: once the student is in the university, they will decide whom will be their friends.

An agent will choose a couple among their friends, if certain conditions are given: reproduction probability taken into account the age and the demographic distributions; election of the “candidates” based on boolean conditions as “not child”, “not married”, “same sex”, etc; candidate choice determined by the similarity rate. These rules are consistent with the friendship dynamics explained in the section 2. Again according to the demography implemented, the couples will give birth a certain number of children that will inherit their characteristics and values.

Thus, the agents form a network where the nodes are the individuals and the links can be of type “friend” or “family” (couple, parents, children). The more friendships exist, the more couples and families will be formed. However, it is not only the quantity which is important: the matchmaking process should return similar couples, minimizing the exceptional cases where two very different people are married.

5 Fuzzification of Mentat

5.1 The Baby's First Steps: Attributes and Similarity

Applying the concepts of section 3 into Mentat, it has been modified, step by step, fuzzifying a collection of aspects: agent characteristics, similarity measure, friendship relationship, and the matchmaking process. Besides, we will introduce the friendship evolution function previously mentioned in section 2. The result ABM has been called Fezztat.

Mentat uses a similarity function for several purposes, as it has already been explained. It is built with a gratification method based on the comparison of the agent attributes. However, the technique is not very sophisticated and could be improved. The use of fuzzy logic would significantly increase its accuracy. But if we want to use

fuzzy operators, first we have to fuzzify the variables where they are applied or, formally, define fuzzy sets over these variables.

Thus, the agent attributes, very different from each other, were normalized in the real interval $[0, 1]$. For example, we would have the fuzzy set $\mu_{\text{economy}}: U \rightarrow [0, 1]$, and an individual with a $\mu_{\text{economy}}(\text{ind}) = 0.7$ would be a person quite wealthy.

Afterwards, the fuzzy similarity can be defined using a T-indistinguishability, which generalizes the classical equivalence relations. The mathematical explanation beneath it can be found in [17], but roughly the distance between the attributes of the two agents compared is “how far are they”, so its negation will point out “how similar are they”. The aggregation of each couple of attribute similarities will return the total similarity rate. The negation used is a fuzzy strong negation N [18] and the aggregation an OWA operator [19]. And so, the relation is:

$$R_{\text{similarity}}(\text{ind}, \text{ind}_2) = \text{OWA}(\forall \mu_i \in \text{ind}, N(d(\mu_i(\text{ind}), \mu_i(\text{ind}_2)))) \quad (4)$$

An OWA is a family of multicriteria combination (aggregation) procedures. By specifying suitable order weights (which sum will result always 1) it is possible to change the form of aggregation. The arithmetic average in the example OWA would need a value of $1/n$ to each weight (where “n” would be the number of attributes). Through these weights, it is possible to control the importance of each attribute in the global similarity.

5.2 Growing-Up: Friendship and Couples

Although the agents comparisons have been improved with the previous method, the potential of this new subtle similarity function would not be used if left like that. Supporting what it was pointed out in section 2, we will link the concepts of similarity and friendship in a fuzzy way. For doing so, the friendship is turned into a fuzzy relationship, completely different from its boolean nature in Mentat. This relationship is naturally “fuzzy”, and the previous use (to be or not to be a friend) was an oversimplification not found in reality: in real world, there is a continuous range of degrees of friendship. With the new $R_{\text{friend}}: U \times U \rightarrow [0, 1]$, each agent has a range from close friends to acquaintances.

Now that it is formally defined, it is needed to specify an evolution for it. Therefore, the friendship logistic function of (1) is used here. Every “step” of time, each agent will update its friendship with its linked agents. Depending on how similar they are, and how old is their friendship, they will end up being very close friends or just stay as acquaintances. Formally (where W is the already defined logistic function (1) and t_{friend} is the time of friendship):

$$R_{\text{friend}}(\text{ind}, \text{ind}_2) = W(t_{\text{friend}}(\text{ind}, \text{ind}_2), R_{\text{similarity}}(\text{ind}, \text{ind}_2)) \quad (5)$$

The final improvement through fuzzy logic is related to couples and the matchmaking process. The “has couple” relation is clearly boolean and cannot be fuzzified. But the process of choosing the couple, as it was described in the section 4, can take into account new information obtained with the modifications already made. Mentat has several boolean conditions for filtering the “candidates”, and these can not be changed (“same sex” or “not married” are impossible to fuzzify). The similarity function that

uses has already been improved. And now we propose to introduce the friendship degree in the selection.

Therefore, it has been defined a new relationship “compatible” that will measure the possibilities of a candidate to be selected as a couple:

$$R_{compatible}(ind, ind_2) = OWA(R_{friend}(ind, ind_2), R_{similarity}(ind, ind_2)) \quad (6)$$

The weights of the OWA can be redefined depending on the importance given to each term. After some experimentation, Fezztat uses equal weights. It could be argued that, as the friendship evolution already takes into account the similarity, this one would not be necessary, as its information is already included. However, the friendship relationship can achieve its maximum with several candidates of the same agent, and it is specially in those cases where the similarity is very useful.

6 Results and Discussion

Here we present a comparison between several versions of the ABM. Four different implementations, each one with two configurations have been analyzed, focusing in three measures. The fuzzy modifications have been grouped in two main ones: “*Fuzz-Sim*”, when the attributes are normalized and the similarity operator fuzzified, as stands the subsection 5.1; and “*Fuzz-Fri*”, when the friendship turns to be fuzzy, evolving over time and affecting the partner choice (subsection 5.2). The four ABM represent all the possible combinations between these, represented in the table 1 in the pair (*Fuzz-Sim*, *Fuzz-Fri*). Thus, we have the classic version of *Mentat* explained in section 4, with no fuzzy properties; the *Mentat_{FuzzSim}*, simply the same ABM but with the “*Fuzz-Sim*”; the *Mentat_{FuzzFri}* with “*Fuzz-Fri*” but not “*Fuzz-Sim*”; and last *Fezztat*, with all the fuzzy modifications.

The two configurations deal with two possible ways of friendship emerging: one promoting random friends (and therefore an agent can be linked to a non-similar neighbor) and the other promoting similarity-based friends (and therefore an agent will rarely be linked to a non-similar neighbor, as it will give priority to the most similar ones). This is not a trivial decision, because the friendship evolution function already deals with similarity, and if a neighbor is not similar at all, it will never be more than an acquaintance. It is not evident if the closer way to real-world is giving double strength to similarity (in the second option) or letting randomness to decide who will be the friend (and thus maybe ignoring similar people). It has to be mentioned that none of the two configurations is so deterministic and both are based on probabilities.

The parameters analyze the couples and how they are affected by the changes in the configuration and fuzzification. The $R_{Similarity}$ shows the proximity taking into account all the characteristics of each partner in a couple. The R_{Friend} focuses in the friendship link between them, which in a way (according to the logistic function) depends in their similarity too, but also in the time spent together. The $R_{Compatibility}$ is taken as an average of the other two. The values have been obtained after averaging the output of several executions of each version. And in every execution, it is the mean of the property in every couple.

As the first two ABM has a boolean friendship, their compatibility is always the same as the similarity. In the first configuration, when the friendship is rarely involved in the neighbors linked, the similarity rates are very similar in all the versions. However, in the second one it's clear that the ones with fuzzy similarity slightly increase their success. But the bigger changes can be observed in the friendship: *Fezztat* beats the other versions with a greater R_{Friend} and $R_{Compatibility}$, specially in the second configuration. These results approach the theoretical qualitative assessments.

Finally, we compared the four different orders of fuzzification described above for both $R_{Similarity}$ and R_{Friend} of couples, by using the statistical test "One-Way Analysis of Variance", in order to detect evidence of difference among the population means. The Fisher's statistical significance test equals to 7.281, with a P -value $P < .0001$. This small P -value provides strong evidence against null hypothesis, namely, that the difference in the means among the four orders of fuzzification are by chance, both for $R_{Similarity}$ and R_{Friend} of couples. Therefore, the differences among the means analysed can be attributed to model's fuzzification.

Table 1. Comparison among the different ABM, in increasing order of fuzzification

	<i>Mentat</i> (0,0)	<i>Mentat</i> _{FuzzSim} (1,0)	<i>Mentat</i> _{FuzzFri} (0,1)	<i>Fezztat</i> (1,1)
Config. Random-friendship				
Mean $R_{Similarity}$ of couples	0.76*	0.77	0.76*	0.77
Mean R_{Friend} of couples	(**)	(**)	0.72*	0.80
Mean $R_{Compatibility}$ of couples	0.76*	0.77	0.54*	0.62
Config. Similar-friendship				
Mean $R_{Similarity}$ of couples	0.73*	0.77	0.73*	0.78
Mean R_{Friend} of couples	(**)	(**)	0.54*	0.76
Mean $R_{Compatibility}$ of couples	0.73*	0.77	0.39*	0.59

*: The original *Mentat*'s similarity has other range, but here they have been normalized in the interval [0,1] in order to be compared.

** : When the friendship is not fuzzified, all the couples are friends (as this is a boolean property).

7 Concluding Remarks

In this paper we have explained some concepts of social relationship dynamics, including an evolution function that was applied for the changing of friendship over time. After justifying the suitability of fuzzy logic in this context, we proceeded to apply it in an existing ABM. Therefore, we defined fuzzy sets over each agent attribute, and a new fuzzy similarity operator that would influence friendship emergence and partner choice. The friendship relationship nature and importance in the model was significantly modified, fuzzifying it, making it evolve using the defined logistic function, and letting it influence in the partner choice as much as the similarity rate. The results of these changes are clearly positive, as long as they improve the proximity to the qualitative assessments of the theory.

To sum up, we have exposed a theory, formalized it, searched where it can be applied, found the useful tools to do that, implemented the application and extracted a collection of results that are used to validate the model against the theory. This validated formalization of the theory could be useful for further study in the field.

Future research lines that could be followed could take into account other interesting friendship theories. There are deep studies in homophily in social networks [3] that could be implemented. An aspect that our model ignores but it is important enough to be considered is the stability of friendship [20]. Besides, Fezztat could be extended to analyze the importance of weak links along one's life, a new possibility that Mentat did not allow.

Acknowledgments. We acknowledge support from the project *Methods and tools for modelling multi-agent systems*, supported by Spanish Council for Science and Technology, with grant TIN2005-08501-C03-01, and the Ford Foundation International Fellowships Program together with the Chilean National Scholarship Program for Graduate Studies.

References

1. Hassan, S., Antunes, L., Arroyo, M.: Deepening the Demographic Mechanisms in a Data-Driven Social Simulation of Moral Values Evolution. In: MABS 2008: Multi-Agent-Based Simulation. LNCS (LNAI), vol. 5269. Springer, Heidelberg (to appear, 2008)
2. Huckfeldt, R.R.: Social Contexts, Social Networks, and Urban Neighborhoods: Environmental Constraints on Friendship Choice. *The American Journal of Sociology* 89, 651–669 (1983)
3. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in Social Networks. *Annual Review of Sociology* 27, 415–444 (2003)
4. Verbrugge, L.M.: The Structure of Adult Friendship Choices. *Social Forces* 56, 576–597 (1977)
5. Blanchard, P., Devaney, R.L., Hall, G.R.: *Differential Equations*, 2nd edn. Brooks Cole (2002)
6. Zadeh, L.A.: Fuzzy Sets. *Information and Control* 8, 338–353 (1965)
7. Smithson, M.J., Verkuilen, J.: *Fuzzy Set Theory: Applications in the Social Sciences*, vol. 147. Sage Publications, Thousand Oaks (April 2006)
8. Zadeh, L.A.: The Birth and Evolution of Fuzzy Logic. *International Journal of General Systems* 17, 95–105 (1990)
9. Katz, A., Hau, M.V., Mahoney, J.: Explaining the Great Reversal in Spanish America: Fuzzy-Set Analysis Versus Regression Analysis. *Sociological Methods & Research* 33, 539–573 (2005)
10. Lomborg, B.: Nucleus and Shield: The Evolution of Social Structure in the Iterated Prisoner's Dilemma. *American Sociological Review* 61, 278–307 (1996)
11. West, J.E., Linster, B.: The Evolution of Fuzzy Rules as Strategies in Two-Player Games. *Southern Economic Journal* 69, 705–717 (2003)
12. Fort, H., Perez, N.: The Fate of Spatial Dilemmas with Different Fuzzy Measures of Success. *Journal of Artificial Societies and Social Simulation* 8 (June 2005), <http://jasss.soc.surrey.ac.uk/8/3/1.html>
13. Epstein, J.G., Ohring, M.M., Troitzsch, K.G.: Fuzzy-Logical Rules in a Multi-Agent System. In: *SimSocVI Workshop*, Groningen, p. 25 (September 2003)
14. Situngkir, H.: *The Ribbon of Love: Fuzzy-Ruled Agents in Artificial Society*. BFI Working Paper Series (2007)
15. Hassan, S., Pavón, J., Arroyo, M., León, S.: Agent Based Simulation Framework for Quantitative and Qualitative Social Research: Statistics and Natural Language Generation.

- In: Amblard, F. (ed.) ESSA 2007: Fourth Conference of the European Social Simulation Association, Toulouse, France, pp. 697–707 (2007)
16. Pavón, J., Arroyo, M., Hassan, S., Sansores, S.: Agent-based modelling and simulation for the analysis of social patterns. *Pattern Recogn. Lett.* 29, 1039–1048 (2008)
 17. Valverde, L.: On the Structure of F-Indistinguishability Operators, University of California at Berkeley, Computer Science Division, Berkeley (September 1984)
 18. Schweizer, B., Sklar, A.: *Probabilistic Metric Spaces*. Dover Publications (November 2005)
 19. Yager, R.R.: Families of OWA operators. *Fuzzy Sets Syst.* 59, 125–148 (1993)
 20. Wellman, B., Wong, L., Tindall, D., Nazer, N.: A decade of network change: Turnover, persistence and stability in personal communities. *Social Networks* 19, 27–50 (1997)

R²-IBN: Argumentation Based Negotiation Framework for the Extended Enterprise

Lobna Hsairi¹, Khaled Ghédira¹, Adel M. Alimi², and Abdellatif BenAbdelhafid³

¹ LI3: Ecole Nationale des Sciences Informatique, 2010 Campus Universitaire la manouba, Tunisie

² REGIM: Ecole Nationale d'Ingénieurs de Sfax, BPW 3038 Sfax, Université de Sfax, Tunisie

³ CERENE-SILI: Université du Havre, Place Robert Schuman BP 4006, 76610 Le Havre, France

Lobna.Hsairi@isimsf.rnu.tn, khaled.ghedira@isg.rnu.tn,
adel.alimi@ieee.org, benabdelhafid@univ-lehavre.fr

Abstract. Nowadays, in order to face new challenges, extended enterprises must be up to date to the new strategic, economic and organizational structures. Consequently, intelligent software based on agent technology emerges to improve system design, and to increase enterprise competitive position as well. The competitiveness is based on the cooperation. Thus, within this cooperation, conflicts may arise. The automated negotiation plays a key role to look for a common agreement. Argumentation theory has become an important topic in the field of Multi-Agent Systems and especially in the negotiation problem. In this paper, first, an overview of an already proposed model MAIS-E² (Multi-Agent Information System for an Extended Enterprise) is presented. Then an argumentation based negotiation framework: Relationship-Role and Interest Based Negotiation (R²-IBN) framework is presented, and within this framework, we focused mainly on, argument generation module via inference rules and argument selection module via fuzzy logic.

Keywords: Extended Enterprise, Multi-Agent System (MAS), Argument Based Negotiation (ABN), MAIS-E², R²-IBN, fuzzy logic.

1 Introduction

Nowadays, a number of new concepts have been proposed, e.g., Virtual Organization, Supply Chain Management, Virtual and Extended Enterprise, etc [1][2]. An extended enterprise is a cooperation of legally independent enterprises, institutions, or individuals. The extended enterprise will be characterized by intensively concurrent engineering based on information technologies such as digitalization, computer network, and artificial intelligence [1]. The intelligent software agent technology provides a natural way to overcome such problems [2]. Agents help to capture individual interests, local decision making using incomplete information, autonomy, responsiveness, robustness, modular and distributed. A Multi-Agent System (MAS), as a society of autonomous agents, is an inherently open and distributed system. It is made up of a

group of agents combined with each other to solve a common problem cooperatively. In addition, negotiation is a key form of interaction in systems composed of multiple autonomous agents [3]. The automated negotiation plays a key role in sharing information and resources to look for a common agreement. The research literature proves that ABN is an effective means of resolving conflicts in MAS [3]. Besides, the fuzzy logic of Zadeh [4] opens new horizons in the vast world of information analysis and treatment. One of the present tendencies in the fuzzy modeling is generating models that take into consideration two fundamental conditions at the same time: interpretability (which is the description capacity of the modeled systems behavior) precision and fidelity of model towards the original system [5].

In this paper, in the first place, we present an overview of our research efforts in developing a MAS architecture named Multi-Agent Information Systems for an Extended Enterprise (MAIS-E²). Then, we define the Relationship-Role and Interest Based Negotiation (R²-IBN) framework. R²-IBN framework is an extension of an existing one namely IBN [6]. In this paper, we present mainly the extensions made in two modules: the argument generation module via inference rules and argument selection module via fuzzy rules based system as an intelligent method in order to better estimate the desirability degree of the argument to send.

The remainder of this paper is structured as follows: section 2 reviews negotiation approaches, section 3 reviews related works, section 4 presents MAIS-E² the Multi-Agent model for an Extended Enterprise, section 5 describes the R²-IBN negotiation framework, and in section 6 concluding remarks and perspectives are given.

2 Negotiation Approaches

A variety of automated negotiation approaches have been studied in the literature. The major families are:

Game-theoretic approaches: Game theory offers a very powerful tool for studying and engineering strategic interaction among self-interested computational agents in general [7], and to automated negotiation in particular. But, there are many drawbacks associated with its use in such negotiation:

- Game theory assumes that it is possible to characterize agent's preferences with respect to possible outcomes.
- Game theory often assumes perfect computational rationality.

Heuristic-based approaches: In heuristic-based approaches, agent designers relax some of the assumptions of game theory, particularly regarding unbounded rationality. But, they still have a number of drawbacks [8].

- The models often lead to the outcomes which are sub-optimal because they adopt an approximate notion of rationality.
- It is very difficult to predict precisely how the system and the constituent agents will behave.

These two approaches are centered around the trading of proposals. However, they have the following main limitations:

- The only feedback that can be made to a proposal is a counter-proposal or an acceptance or withdrawal;
 - It is hard to change the set of issues under negotiation in the course of a negotiation.
- In such case alternative techniques would be needed:

Argument-based approaches: The aim of ABN is to remove the above limitations. The basic idea behind the ABN approach is to allow additional information, called argument¹, to be exchanged, over and above proposals [8]. This information provides a means of changing the negotiation space itself.

3 Related Works

A number of researchers attempted to play agent technology to industrial enterprise integration. Thus, we can distinguish, an Agent-Based Intelligent Manufacturing for the 21st Century [9] and a supply web co-ordination by an agent based trading network with integrated logistic services [10]. With these different efforts, researchers take into consideration the integration problem according to different levels. Our research efforts are different from the others. We take into consideration the communication level and the exchange of information and knowledge. In fact, communication takes place between multiple entities but in this paper we focus only on the reasoning mechanism of just one peer.

Moreover, other researchers proposed many formal frameworks based on ABN for automated negotiation. These efforts take into consideration different aspects. We can distinguish works that deal with protocols [11], abductive reasoning [12], devising and formalizing rebutting and undercutting arguments [13][14], and catering epistemic and motivational reasoning [15]. Our approach is different from these models since they tend to focus on the product of argumentation. Our approach aims at catering in the same framework the main modules of argumentation: Evaluation, Generation and Selection. The whole framework is considered to be appropriate to industrial domain and especially to the extended enterprise case.

4 MAIS-E²: Cooperation Model for an Extended Enterprise

In our proposed model MAIS-E², we distinguish four types of agents: "Enterprise" agents, "Mediator" agents, "Specialist" agents and Personal assistant as shown in figure 1.a. and detailed in [16].

In this work, we consider the extended enterprise like a society of intelligent agents that cooperate to satisfy a common objective.

In this stage of work, we are interested in the functioning of cooperation between "Enterprise" agents by the intermediate of "Mediator" agent agency as depicted in figure 1.b. The MAIS-E² model rely on the existence of the distributed "Mediator" agent agency that assume binding between all partners in the extended enterprise.

¹ An argument is a piece of information presented by the pair (Premises, Conclusion) and have a degree of desirability. It is attached to the exchanged proposals during negotiation. It is also used to improve the negotiation by introducing flexibility via the introduction of new information, which are taken into consideration in the negotiation space.

In MAIS-E² model, “Mediator” agent agency communicates and collaborates between them in order to coordinate their tasks. Within this cooperation, as defined by Ferber [17], the partners are also getting to: Collaborate; Coordinate their tasks and roles; and Resolve conflicts. To describe MAIS-E² functioning, and more precisely “Mediator” agent agency cooperation, we will focus on the conflict resolution part of cooperation. Furthermore, the extended enterprise environment is characterized by: openness, heterogeneity, dynamism etc. These characteristics make the conflict resolution process more and more difficult. As a solution we use automated negotiation. In fact, designing negotiation mechanisms aim at satisfying a number of features: simplicity, efficiency, distribution, symmetry, stability and flexibility [15]. Thus, in the extended enterprise case study the "Flexibility" feature play a crucial role by taking into consideration the heterogeneity of different actors. For this reason, we adopt the Argumentation Based Negotiation approach, and we propose the R²-IBN framework detailed in the following section.

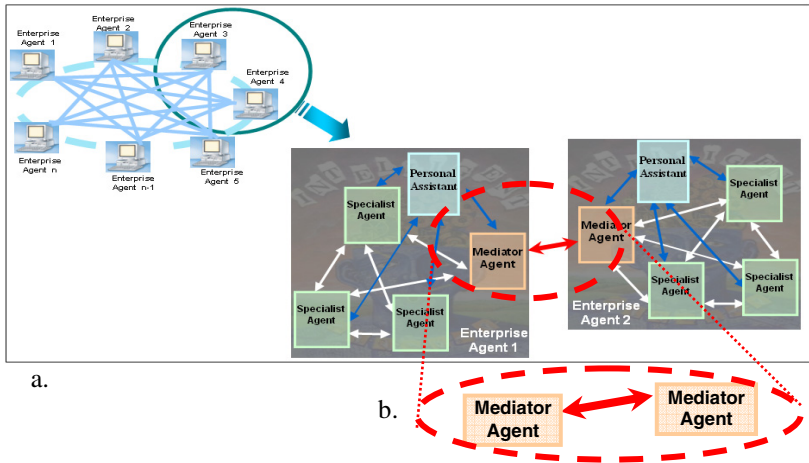


Fig. 1. a. Organizational model of MAIS-E². b. “Mediator” agent agency.

5 R²-IBN: Argumentation Framework

R²-IBN: Relationship-Role and Interest Based Negotiation framework is an extension of an existing one which is IBN proposed by Rahwan [15]. IBN uses mental attitudes: beliefs, desires, goals and planning rules as the primitives upon which argumentation are based [18]. Being in the context of the extended enterprise, concepts like agent roles and relationships play a crucial role by taking into consideration the heterogeneity of different actors. For that, such concepts are essential to be integrated. Hence, we propose the Relationship-Role and Interest Based Negotiation (R²-IBN) framework. We propose R²-IBN framework as a solution to the conflict resolution problem for the “Mediator” agent agency of MAIS-E² model already advocated in the previous section. In this paper, we focus only on the reasoning mechanism of one “Mediator” agent. So to present the description of R²-IBN framework, it is crucial to advocate that

in order to be able to negotiate with other agents, based on argumentation approach, a “Mediator” agent must be provided with three set of rules:

- Argument Evaluation Rules;
- Argument Generation Rules and
- Argument Selection Rules.

In this paper, and due to lack of space, we will describe the argument generation and selection rules, which constitute the main differences from the original framework IBN. For the argument evaluation rules, in R²-IBN we proceed in the same way as in IBN via three distinct parts of evaluation: one for reasoning about beliefs, another for arguing about what desires should be pursued and a third, for arguing about the best plan to intend in order to achieve these desires. We made some extensions by taking into consideration: the roles of the agents and their relationships in the reasoning mechanism of a “Mediator” agent.

Before detailing the generation and selection rules, it is important to introduce the definition of the proposed extensions: Role and relationship (advocated by the confidence level attributed to the peers).

Let \mathcal{L} be a propositional language with \wedge for conjunction, \rightarrow for implication and \neg for negation. Let \vdash denote classical inference.

Role

An agent role is defined by a set of agent capabilities².

$$RolAg = \{Cap_1, \dots, Cap_i, \dots, Cap_n\}$$

The addition and/or deletion of agent's capabilities can generate an evolution of the agent's role.

The “Mediator” agent has its basic capabilities defined as

$$B_{Cap} = \{(Cap_i, cap_i), i = 1, \dots, n\}$$

Where Cap_i is a propositional formula of \mathcal{L} and cap_i its degree of certainty.

In addition to that, it has the capabilities of their partners defined as:

$$B_{ACap} = \{(A_j, (Cap_i^{A_j}, cap_i^{A_j})), i = 1, \dots, n; j = 1, \dots, m\}$$

Where A_j is an agent, $Cap_i^{A_j}$ is a propositional formula of \mathcal{L} and $cap_i^{A_j}$ its degree of certainty.

Confidence Based relationship

Soh [19] define five Confidence parameters:

- The helpfulness of the peer: the satisfaction degree of requests to the peer.
- The reliance of the peer on the agent in terms of the ratio of receiving requests from the peer among all peers.

² We differentiate between the capabilities of communication: request, tell, confirm, ... and the capabilities of task achievement. In this definition of agent role we mean by the capabilities of task achievement, the role of the agent in the extended enterprise case.

- The tardiness degree: the communication delay between the agent and the peer.
- The hesitation degree: how readily the peer is to agree with a request.
- The availability degree of capability: whether the peer possesses the desired capability to solve task.

Following this list of parameters and according to their applicability to the extended enterprise characteristics, we can define an agent's confidence parameters as follows:

$Conf_{Para} = \{\text{helpfulness, reliance, tardiness degree, hesitation degree, availability degree}\}$.

We define the confidence base as follows:

$$B_{Conf} = \{(A_j, Conf^{A_j}), j=1, \dots, n\}$$

Where $Conf^{A_j}$: the confidence value attributed to the agent A_j .

The confidence value is defined by the function:

$$Conf: A \times A_j \rightarrow [0, 1]$$

$$Conf^{A_j} = W_R * Re\ put_{A_j} + W_C * \frac{nb_ParaConf \left(Conf_{A_j}^k \right)}{\sum_{k=1}^{nb_ParaConf} \left(Conf_{A_j}^k \right)} \tag{1}$$

Where:

- A_j : an agent.
- $Re\ put_{A_j} \in [0,1]$: The reputation value (value made by indirect interaction) of agent A_j passed by partner agents (for the first meeting).
- $Conf_{A_j}^k \in [0,1]$: The confidence value (value made by direct interaction) attributed to agent A_j for the k^{th} parameter ($k \in Conf_{Para}$).
- $nb_ParaConf$: The number of confidence parameters used.
- $w_R \in [0,1]$: The weight attributed to reputation.
- $w_C \in [0,1]$: The weight attributed to confidence (direct interaction).

Since, the reputation and direct confidence values have not the same impact on the total confidence value ($Conf^{A_j}$), we propose to handle this difference by assigning

different weights W_R and W_C to $Re\ put_{A_j}$ and $\frac{nb_ParaConf \left(Conf_{A_j}^k \right)}{\sum_{k=1}^{nb_ParaConf} \left(Conf_{A_j}^k \right)}$ respectively, such as

$W_R + W_C = 1$ and $W_R < W_C$. We also attributed to the $Re\ put_{A_j}$ a minor value via the

weight W_R because the reputation given by other peers is not always reliable and we attribute to the direct interaction a major value via the weight W_C since it is an experience made by the agent itself.

5.1 Argument Generation Rules

The argument generation process consists in generating candidate arguments, by a “Mediator” agent, to be presented to a negotiation counterpart (another “Mediator” agent). An argument is constituted by the following pair (Premises, Conclusion). The premises are constituted by believes, desires of the proponent and capabilities of the opponent. So, the agent role, via a set of its capabilities, is taken into account when generating arguments.

The argument generation process takes shape in the deductive reasoning context. We also need inference rules to infer from knowledge base. For that, we adopt the basic rules of reasoning used in classical logic: the Modus Ponens for forward reasoning and the Modus Tollens for backwards reasoning, whose symbolic expressions are:

Modus Ponens $P \rightarrow Q, P \vdash Q.$

Modus Tollens $P \rightarrow Q, \neg Q \vdash \neg P.$

5.2 Argument Selection Rules

The “Mediator” agent may generate several arguments for any specific situation; and only one argument should be used for each step of the negotiation. The key features that determine what argument to send are the following: utility of the proposal to the proponent (“Mediator” agent); the confidence degree³ (which describes the degree of trust that the proponent puts in the opponent); and the capability degree of the opponent in the specific task (according to the proponent belief). Since these characteristics are generally imperfect and uncertain. We propose a fuzzy rule based system as an intelligent method in order to better estimate the desirability degree of the argument to send [20]. Thus our fuzzy controller parameters are as follows:

Inputs parameters:

- Confidence degree: {Low, Medium and High}.
- Capability degree: {Low; Medium and High}.
- Proposal utility: {Low and High}.

Output parameter:

- Argument desirability degree: {Weak and Strong}.

All parameters are defined by trapezoidal membership functions illustrated by figure 2.

The rule base model is built by combination of inputs and output parameters. So, rules of the following form are encoded into our “Mediator” Agent:

Rule1: IF Confidence is low AND Capability is high AND Utility of the proposal is high THEN send an Argument with high degree of desirability.

Rule2: IF Confidence is high AND Capability is high AND Utility of the proposal is low THEN send an Argument with low degree of desirability.

For inference process, Mamdani (Max-Min) [20] inference method is used. The Center of Gravity (COG) method is used for defuzzification to determine the output value: The argument desirability degree.

³ Calculated and updated using function (1). In the experimental setting, $W_R = 0.2$ and $W_C = 0.8$.

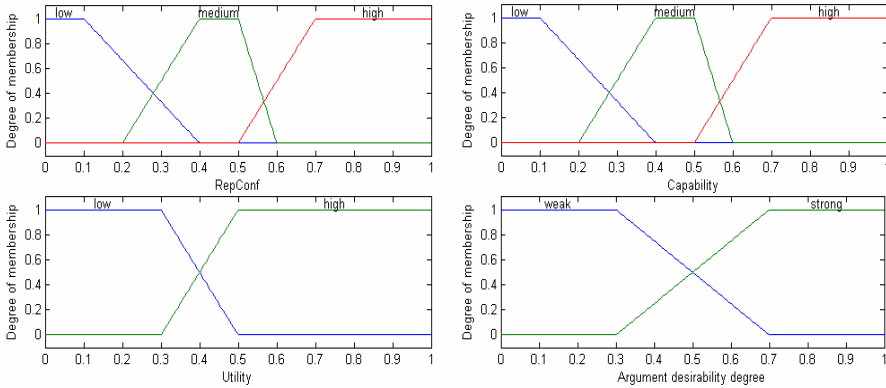


Fig. 2. Membership functions: Confidence, Capability, Proposal utility and Argument desirability respectively

Experimental Evaluation: The experiments aim to postulate about the efficiency: negotiation cycle and argument exchanged numbers to reach agreement. For that, at this stage of work, we restrict the negotiation to two “Mediator” Agents in harbor application, the set of the argument that can be uttered between 0 and 10 and the negotiation cycle per proposal between 1 and 10. A simulation run involves 1000 separate negotiation encounters. These encounters are paired between “Mediator” agents using the same argument selection mechanism; R²-IBN Fuzzy: using our mechanism, Ramchurn Fuzzy: using Ramchurn mechanism (follow a Rhetoric Approach) [21]; Ramping: using Kraus mechanism (use different kinds of arguments: appeals, rewards and threats) [22]; Random: choosing any available argument at random and Non-arguing: not using arguments. The simulation was repeated 20 times.

From figure 3 we can deduce that R²-IBN Fuzzy mechanism fares better than the other ones. The better performance of R²-IBN fuzzy selection mechanism is mainly provided by using fuzzy rules. Thus, the agent has more meaningful means in selecting its arguments than the other agents and hence does better. In addition, in its fuzzy rules, R²-IBN “Mediator” agent takes into consideration the certainty degree of the capability of the opponent and the confidence value attributed to this opponent. This

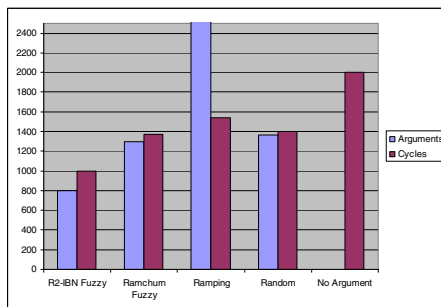


Fig. 3. Number of Arguments and Cycles to reach agreement

mechanism, improves the estimation of the desirability degree of the argument to be send which leads to optimize the negotiation cycle and argument exchanged numbers.

6 Conclusion

The choice of intelligent software agent provides a natural way to design extended enterprises (EE) because the intrinsic features of MAS correspond to those to be preserved in the hoped EE. MAIS-E², provides the cooperation of different actors in EE. In addition to that, the analysis of cooperation leads to the problem of conflict resolution. For that, ABN is an effective means of solving conflicts in MAS. In this way, we have already defined R²-IBN framework. R²-IBN argument selection mechanism has more meaningful means in selecting its arguments. However, it is still difficult to evaluate the number of arguments and cycles to reach agreement between more than two “Mediator” agents. In addition, each one can use its own selection mechanism.

The future research goals consist of: making deep experimental evaluation of the argument selection process, extending the negotiation between more than two “Mediator” agents, conceiving an ontology to allow “Mediator” Agents to communicate despite having semantic and understanding problems and the instantiation of the R²-IBN for harbor application, that was considered as an extended enterprise.

References

1. Tsung-Yi, C.: Knowledge sharing in virtual enterprises via an ontology-based access control approach. *Computer in industry* 59, 502–519 (2008)
2. Martinez, M.T., Fouletier, P., Park, K.H., Favrel, J.: Virtual enterprise-organisation, evolution and control. *I. J. of production economics* 74, 225–238 (2001)
3. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in Artificial Intelligence. *Artificial Intelligence* 171(10–15), 619–641 (2007)
4. Zadeh, L.A.: Fuzzy sets. *Inform. Control* 8, 338–358 (1965)
5. Casillas, J., Cordón, O., Herrera, F., Magdalena, L.: Interpretability issues in fuzzy modeling. In: *Studies in fuzziness and soft computing*, vol. 128. Springer, Heidelberg (2003)
6. Rahwan, I., Sonenberg, L., Dignum, F.: On interest-based negotiation. In: Dignum, F.P.M. (ed.) *ACL 2003. LNCS (LNAI)*, vol. 2922, pp. 383–401. Springer, Heidelberg (2004)
7. Von Neumann, J., Morgenstern, O.: *The Theory of Games and Economic Behaviour*. Princeton University Press, Princeton (1944)
8. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.J.: Automated negotiation: prospects, methods and challenges. *IJG of Decision and Negotiation* 10, 199–215 (2001)
9. Qiao, B., Jianying, Z.: *Agent-Based Intelligent Manufacturing System for the 21st Century* (1999), http://www.shaping-thefuture.de/pdf_www/152_paper.pdf
10. Kovac, G.L., Paganelli, P.: A planning and management infrastructure for large, complex, distributed projects—beyond ERP and SCM. *Computer in Industry* 51, 165–183 (2003)
11. McBurney, P., Parsons, S., Van Eijk, R., Amgoud, L.: A dialogue-game protocol for agent purchase negotiations. *J. of Autonomous Agents and Multi-Agent Systems* (2002)

12. Sadri, F., Toni, F., Torroni, P.: Dialogue for negotiation: agent varieties and dialogue sequences. In: Meyer, J.-J.C., Tambe, M. (eds.) ATAL 2001. LNCS (LNAI), vol. 2333. Springer, Heidelberg (2002)
13. Parsons, S., Sierra, C., Jennings, N.: Agents that reason and negotiate by arguing. *J. of Logic and Computation*. 8(3), 261–292 (1998)
14. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence* 34(1-3), 197–215 (2002)
15. Rahwan, I.: Interest-based Negotiation in Multi-Agent Systems. PhD Thesis, Department of Information Systems, University of Melbourne, Melbourne, Australia (2004)
16. Hsairi, L., Ghédira, K., Alimi, M.A., Ben Abdelhafid, A.: Resolution of Conflicts via Argument Based Negotiation: Extended Enterprise Case. In: IEEE/SSSM 2006. Université de Technologie de Troyes, pp. 828–833. IEEE Press, France (2006)
17. Ferber, J.: Les systèmes Multi-Agents Vers une intelligence collective. InterEditions, Paris (1995).
18. Rahwan, I., Pasquier, P., Sonenberg, L., Dignum, F.: On the Benefits of Exploiting Underlying Goals in Argument-based Negotiation. In: Proceedings of 22nd Conference on Artificial Intelligence (AAAI). AAAI Press, California (2007)
19. Soh, L., Li, X.: Adaptive, Confidence-based Multi-agent Negotiation Strategy. In: AAMAS 2004, New York, USA (2004)
20. Jang, J.: Neuro-Fuzzy and Soft Computing. Hall, P. (ed.) (1997)
21. Ramchurn, S., Jennings, N.R., Sierra, C.: Persuasive negotiation for autonomous agents: a rhetorical approach. In: IJCAI Workshop on Computational Models of Natural Argument, pp. 9–17. AAAI Press, Menlo Park (2003)
22. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence* 104, 1–69 (1998)

Extending Pattern Specification for Design the Collaborative Learning at Analysis Level

Jaime Muñoz Arteaga¹, Ma. De Lourdes Margain Fuentes², Fco. Álvarez Rodríguez¹,
and Carlos Alberto Ochoa Ortíz Zezzatti³

¹ Universidad Autónoma Aguascalientes, Av. Universidad # 940, Ags., CP. 20100 México
jmunozar@correo.uaa.mx, fjalvar@correo.uaa.mx

² Universidad Politécnica de Aguascalientes, Av. Mahatma Gandhi Km. 2, Ags. México
lmf@upa.edu.mx

³ Universidad Autónoma de Ciudad Juárez, México
megamax8@hotmail.com

Abstract. The collaborative learning offers for students a rich collaborative learning space before, during and after the sessions given for a teacher. In order to design and develop properly technological support for the collaborative learning with the learning objects, the designers require specification tools to take into account the different stakeholders notably students and teachers since the early phases. The analysis level is one of the engineering software phases, where the current work purpose at this phase the use of pattern paradigm as a specification tool in order to specify the best practices of collaborative learning and also as a common language between instructional and technological designers.

Keywords: Pattern Language, Collaborative Learning, Learning Objects Design.

1 Introduction

Collaborative learning is considered as an indispensable practice of social construction of learning. The social construction of learning makes it possible for Teachers and students to treat different problems in specific contexts, generating network learning, which allows for a richer and more cooperative didactic process. This way responds to the questions of where and when the human being learns [1]. In recent literature, different proposals of experts have been advanced for the development and use of learning objects, making it easy for experts in contents to design and implement didactic units called learning objects [3]. Moreover, others works have proposed the production of learning objects based on patterns [4] [10] and improve software usability through architectural patterns [5]. From the engineering software perspective, the development cycle of learning objects is done like a software life cycle with the cascade model. After requirements have been determined, an analysis stage is followed, this stage is one of the most important steps for learning objects because delimits the content and context in which design and development is going to apply. Finally, the life cycle concludes with the implementation level using SCORM stands for Sharable Content Object Reference Model. All the levels follow specifications

but, in this paper the authors only describe detail the analysis level for the reason that by this mode is feasible to model the collaborative learning and know if the students really learned in this way. This works consider that the specification of collaborative learning should be a process to take into account in methodologies of instructional engineering, due to the absence of a language that should help represent and implement the social process of learning. Also the use and re-use of learning objects for collaborative learning is required and could be representing through patterns. Reuse is one of the most widely followed practices in problem solving coming from software engineering [2].

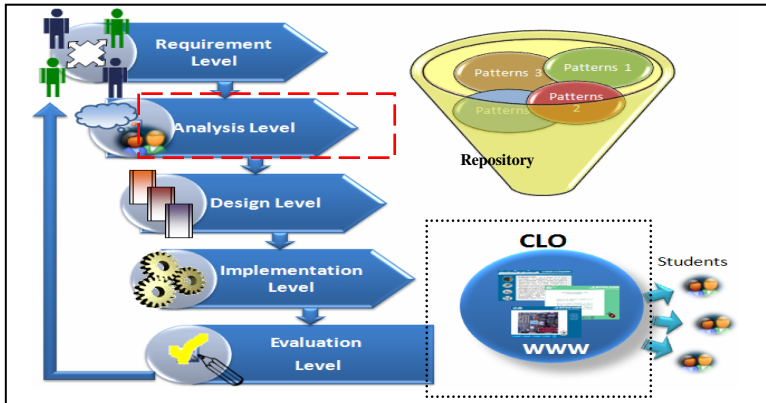


Fig. 1. MACOBA Methodology

According to Gamma et al. definition [6], a pattern is a solution to a problem that is used repeatedly in similar contexts with some variations in this implementation. For example, some authors have implemented interaction patterns for the design of usable web interfaces [7]. Others have used patterns to improve the usability of software utilizing the UML language [8]. This way, we can see that, through the creation of a language of patterns, the process of socio-construction of collaborative learning can be implemented, by distributing the specification of patterns at different abstraction levels to design and develop e-learning systems. This article purpose extends the pattern specification for design the collaborative learning at analysis level of the MACOBA methodology [9] (See figure 1). Furthermore, the authors present a learning object to learn topics about engineering hardware. This LO was developed by Polytechnic University of Aguascalientes.

2 Problem Outline

In software engineering is quite well known that different requirements could change continuously, so they are not foreseen in the initial design phases. Too often, a lack of constant communication between instructional designers and technological designers brings about failures in design, constant changes and, as a consequence, a rupture in the design which causes many problems during the design phase. Additionally, in the designing of collaborative learning, this problematic is accentuated due to the fact that

software also has to adapt to the great number of different profiles and styles of different participants [11]. We may say that the main source of the problematic can be seen during the phase of the planning of all the activities. Departing from this perspective, we see that instructional designers don't count on methodologies to communicate technological designers the practices required in collaborative learning. We can see that there is no common language between instructional designers and technological designers. Specifying at analysis level the collaborative learning using the pattern paradigm without loss the original participation of actors (Teacher and students) in such kind of learning, and additionally we have a common language between instructional designers and technological designers based on patterns.

3 Extending Patterns Specification at Analysis Level

The patterns represent simple and elegant solutions for solve no obvious and recurrent problems inside of the oriented objects design. In addition, they are solutions based on people experience. UML is a set of tools for model systems. These tools are the use case, classes diagram, states diagram, sequences diagram, activities diagram, collaborations diagram, components diagram and distribution diagram.

Table 1. Use Case Collaborative Learning Notation
















System name	
Simple Use Case	
Collaborative Use Case	
Communication between actor and use case	
System limit	
Actor on stand alone action	
Actor on collaborative action	

Table 2. Sequence Diagram Collaborative Learning Notation

Object	
Collaborative Object	
Simple messages	
Collaborative messages	
Time	
Actor on stand alone action	
Actor on collaborative action	
Collaboration point	

This work purposes to integrate the UML notation in order to specify the collaborative learning. This proposal includes in particular a new format for use case and a new notation for use case diagrams and sequence diagrams. This notation are related to specify the requirements without consider technical aspects and such a way to cover the lack of a common language between instructional designers and technological designers.

3.1 Use Case Collaborative Learning Pattern (UCCL)

The use case for collaborative learning initially has the purpose to organize the plan, explaining “what” must do a learning object (LO). The next level explains “how” a

LO will be designed. And finally, the use case also describes the interaction mode with the user. Fulfill the specifications of the UCCL must be simple, intelligible, clear and concise and must detail the stand alone or collaborative tasks of the actor. The UCCL is divided in three sections. In the first section the UCCL contains the specification of patten name context; use case name, author, date, and a full description of the use cases, actors, number of actors, rolls, preconditions, and normal flow.

The middle section contains a complete activity description including roles, cases and full description of cases. a) The role is the actor that makes that activity, for example: *“Teacher”*. b) The case is a specific activity that will be done by the actor for example: *“To prepare initial exploration.* c) The brief description, explains in this case the Teacher details for prepare the initial exploration and the resources used. For example: *“Browsing through the screens which will be explained through text, images, audio and video units of flash memory”*.

The last section contains the alternative flow like optional situations and the post conditions. All of this makes emphasis in the collaborative action. The UCCL patterns enclose a specification and a defined well format. In the next figure both are presented (See table 1). To show in a more simple way this uses cases for the instructional and technological designers are possible to model the scenario view including the roles and activities through illustration. In this sense is proposed the diagram for use case for collaborative learning UCCLD.

3.2 Use Case Collaborative Learning Diagram (UCCLD)

This part explains the basic elements for build the use case diagram for collaborative learning. First of all, is required to specify the system name, the actor’s relation (roles) and activities (cases). A single ellipse is proposed for simple use case, double ellipse for collaborative use case. That is, the student or teacher collaboration activity. Then, to represent the system limit; we use a square. Finally, the simple activity actors are represented by single picture and the collaborative, twice. (See table 2).

3.3 Sequence Diagram Collaborative Learning (SDCL)

When the scenario is modeled, the designers need to think about the interactions elements in order and sequence events. In this sense the use of Sequence Diagram is proposed for Collaborative Learning (SDCL). Similar than use case diagram, the sequence diagram need specify the elements, to represent the interaction of a collaborative learning. To illustrate the sequence diagram is essential first of all represent the objects. One box could represent a single object and two boxes a collaborative object. These objects require to be communicating through messages in the scene and could be represented by horizontal lines. Then, the messages are drawn chronological from the part superior of the diagram to the inferior part; the horizontal distribution of the objects is arbitrary. The vertical axis is represented by discontinuous lines and these represents the line time. Each object is connected with a vertical line, and the messages are represented by arrows. Finally the circles (one inside another one) represent a collaborative point. The time flows from top to bottom. In the same manner like in the use cases the actor should identify stand alone or collaborative actions.

4 Applying the Extending Pattern Specification

In this section, a hardware engineering case study in the computer science domain is exposed. The pattern for this use case describes the collaborative work with the users whom play students role and the teacher. The work is developed in groups of 4 or 5 students who collaborate in a collaborative tool (forum).

Table 3. Analysis Level – Use case collaborative learning pattern (UCCL)

Pattern Name		Use Cases
Context		Computer Science, Hardware Engineering
Use Case Name:		Memory Flash
Author:		L. Margain
Date:		10/02/2008
Description: This use case describes the collaborative work with students they work in groups of 4 or 5 students who collaborate and they could see videos. This use case covers the curricula and allows access in the case of use; students are located in different geographic places.		
Actors: Student Teacher	No. Actor 1 2	Roles: Student Teacher
Preconditions: The users are connected to internet from different geographic places. The users need right to use a collaborative tool (Forum as collaborative tool)		
Normal Flow		
Roles	Case	Description
Student	Initial exploration	Browsing through the screens which will be explained through text, images, audio and video units of flash memory.
Student Teacher	Forum Discussions	Explain in a forum opinion about the usefulness in the daily life of portable storage media in everyday life.
Student	Develop report	Develop a report about the knowledge learned.
Student	Conduct Questionnaire	Conducting a self-assessment using a questionnaire. This quiz will give the number of successes achieved.

Table 3. (continued)

Teacher	Rate Results	To analyze the strengths and weaknesses obtained from the student and provide suitably qualified.
Teacher	Conducting Feedback	Conduct feedback provided an analysis of the student's strengths and weaknesses, what will rise to the forum.
Alternative Flow: None.		
Pos conditions: The collaborative tool (forum) has been enabled.		

This work purposes the notation for a use case in term of pattern format, an example is shown in the figure 1, the construction of the use case diagram can start modeling single actions for the “student” role like “prepare initial exploration, to write pre-diagnosis, to discuss on line wiki, etc”. (See figure 2)

In order to shows how the learning object (LO) could be implemented derived from the previous specification at the analysis stage, a LO is shown in the figure 3 and 4.

The final LO shows the impact of the specification. For example, the impact on the cost of specification patterns is determined at each level for the cost of each process. This cost will be apportioned among the number of LO that are produced by

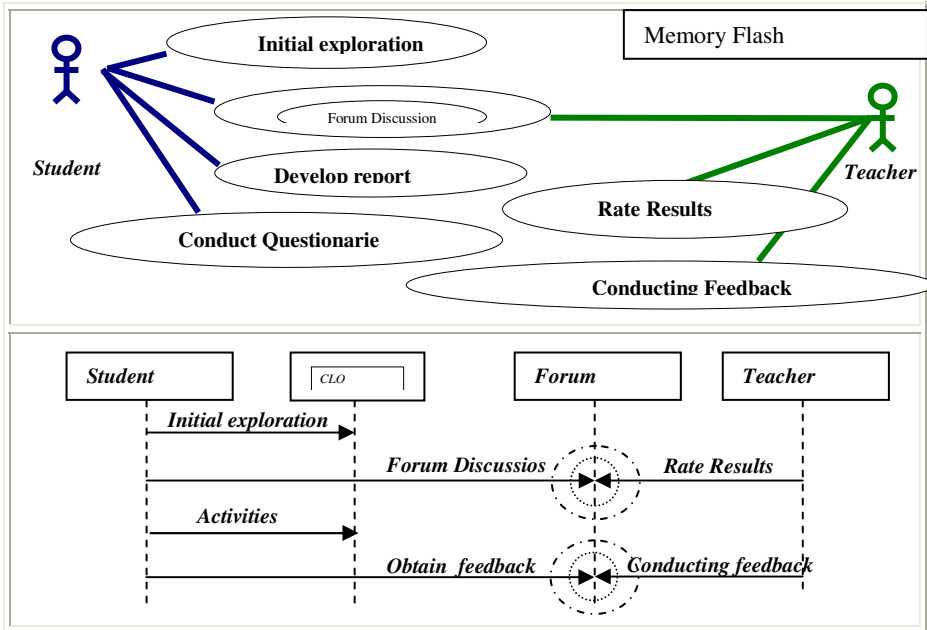


Fig. 2. Use Case Collaborative Learning & Sequence Diagram Collaborative Learning



Fig. 3. Initial Exploration



Fig. 4. LO Resources (Video)

Avatar	Fecha	Asignaturas	Mensajes	Último Mensaje
[Avatar]	10 Ago 2007 - 10:23	3	3	Via 10 Ago 2007 - 10:23 (100%)
REPORTE				
[Avatar]	10 Ago 2007 - 17:24	3	4	Via 10 Ago 2007 - 17:24 (100%)
[Avatar]	6 Ago 2007 - 0:12	2	2	Lun 6 Ago 2007 - 0:12 (100%)
[Avatar]	10 Ago 2007 - 10:12	1	1	Via 10 Ago 2007 - 10:12 (100%)

Fig. 5. Forum discussion

utilidad de las memorias USB - 07/08 a las 23:17

La memoria de las memorias USB son pequeñas y ligeras. Son populares entre personas que necesitan transportar datos entre la casa, escuela o lugar de trabajo. Actualmente, la memoria USB puede retener los datos durante unos 10 años y escribir un millón de veces.

Registrado: 04 Ago 2007
Mensaje: 1

utilidad de las memorias USB - 07/08 a las 23:29

Me parece que tu comentario es muy acertado, además cabe de señalar que aunque inicialmente fue concebido para guardar datos y documentos, se ha podido encontrar también en las memorias USB programas de utilidad que el usuario puede ejecutar directamente desde el dispositivo, una necesidad de realizar alguna instalación en el sistema operativo anfitrión.

Registrado: 20 Jul 2007
Mensaje: 6

Fig. 6. Conducting feedback

a certain pattern. Figures 3 and 4 show the initial exploration. In the figures 5 and 6, we can see immediately the participant roles. The roles changes in function of the teacher or the student role. Next we can see the collaboration dynamics in the course of the activities modeled in the analysis stage by sequence diagram collaborative learning (SDCL). The authors finally identify how to specify an intermediate language between the requirements and implementation levels that respond to teacher’s duty of promoting collaborative learning. The authors built a survey in relation to tree constructs: Utility, Relative advantage and Facility. The survey objective was to establish the collaborative tools contribution level. Four learning objects experts answered the survey. The experts are researches from Autonomous (UAA), Polytechnic (UPA) University of Aguascalientes and Juarez of Tabasco University (UJAT). About UTILITY criteria, the CLUC and CLSD tools at analysis level are good. About RELATIVE ADVANTAGE criteria, the contribution of the collaborative tools offer advantages like planning, preventing bad interpretation, preventing dual information and helping to evaluate the requirements stage. About FACILITY criteria, the collaborative tools facilitate the representation of the actor’s interactions and the pattern’s creation.

5 Conclusions

We can conclude the use of patterns in the analysis level offers a solution for the fundamental problem of the learning objects design: the lack of common language between instructional and technological designer. The difficulties of adaptation to that formal language will be disappear, because instructional designer attends the requirement level and technological designer attends the analysis level. The main advantage is the simple representation of the dynamics collaboration at analysis level between actors. Now, the designers could plan the learning objects according the best practices for collaborative learning and also to promote the reusing learning objects. It is possible to use object oriented tools in combination with the collaborative learning notations proposed here (UCCL, SDCL) in order to specify, execute and evaluate learning models. With these tools we have a communication mode between instructional designers and technological designers. In others works the authors explain the process for design and development learning objects and present the patterns. By the time, in the MACOBA methodology [9] all the levels are covered; then a new language is able for implement completely the learning social process based on patterns paradigm.

Acknowledgments

This work is partially supported by ALFA-CID & PROMEP-P/48-2007UAA projects.

References

1. Correa, Z., Luz, M.: Collaborative Learning: a new form of interpersonal dialogue and in network. Digital magazine of Education and New Technologies, Educative Context (28), Year V (2000)
2. Cox, B.: What if the is a silver bullet and the competition get it 1st. Dr. Dobb's journal (1992)
3. Sanchez, G., Polo, J., Hernández: A collaborative virtual system for the design, development and pursuit of didactic units based on the technology of learning objects (2007)
4. D Paloma Díaz, Susana Montero e Ignacio Aedo, "Ingeniería de la web y patrones de diseño". Editorial Pearson Educación (2005)
5. Juristo, N., Lopez, M., Moreno, A., Sánchez, M.: Improving software usability through architectural patterns. School of Computing - Universidad Complutense de Madrid, Spain and School of Computing - Universidad Carlos III de Madrid, Spain
6. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns. Addison Wesley, Reading (1995)
7. Martijn, V.W., Hallvard, T.: Interaction Patterns in User Interfaces, In: 7th.Pattern Languages of Programs Conference, Allerton, Illinois, USA (2000)
8. Moreno, A.M., Sánchez-Segura, M.: Patterns of Usability: Improvement the Software Usability from the moment of Architectonic. In: JISBD 2003, Alicante, Spain (2003)
9. Margain, L., Muñoz, J., Alvarez, F.: A Methodology for Design Collaborative Learning Objects. In: ICALT 2008, Spain (2008)
10. TELL Project. Design patterns for Teachers and educational (2005), <http://cosy.ted.unipi.gr/>
11. Cesarini, M., Monga, M., Tedesco, R.: Carrying on the e-learning process with a workflow management engine. In: Proc. of the ACM Symposium on Applied Computing SAC 2004, Nicosia, March 14-17, pp. 940-945. ACM Press, New York (2004)

Towards the Simulation of Social Interactions through Embodied Conversational Agents

María Lucila Morales-Rodríguez¹, Bernard Pavard², Juan J. González B.¹,
and José A. Martínez F.¹

¹ Instituto Tecnológico de Ciudad Madero,
Av. 1o. de Mayo esq. Sor Juana Inés de la Cruz S/N, Ciudad Madero Tamaulipas, México
{lmoralesrdz, jose.mtz}@gmail.com, jjgonzalezbarbosa@hotmail.com

² IRIT-GRIC, 118, route de Narbonne, 31062 Toulouse Cedex 4, France
Pavard@irit.fr

Abstract. In this paper, we present a pluridisciplinary approach in order to model the behavior of an embodied conversational agent that expresses emotional and social interactions. We present our methodology to reproduce credible social interactions. Particularly, we discuss the role of the context, the culture, and the emotions in the model of the management of speech acts. This model has been implemented in the context of virtual therapy to simulate the interaction between a therapist and a post-CVA patient.

Keywords: emotional and social interaction, situated cognition, embodied conversational agents, virtual therapy.

1 Introduction and Background

The simulation of social interactions in virtual universe opens new horizons in the field of information technologies. The introduction of virtual characters with a certain level of social interaction allows the development of many professional applications which could have only been developed until now in the field of human-human interaction.

The virtual characters and particularly the *Embodied Conversational Agent* [1] tries to capture the richness and the dynamics of the human behavior. To be able to produce this behavior, these life-like characters must communicate credibility in their emotional expressions and be able to express their indexical and reflexive behavior.

We consider that a virtual character able to produce human-like interactive social style through nonverbal expressions, social behaviors and empathy expressions must put the user in a situation close to the real dynamics of the physical and social environment and thus produce a sense of social and emotional immersion.

We believe that this kind of immersion could affect the subjective experience in the virtual universe and thus stimulate the participation of the user. This is really important in a therapy context and particularly in patients with brain injuries caused by cerebral vascular accidents (CVA). Patients of this kind are frequently depressed and have difficulties in evaluating their performances.

The paper is structured as follows. Section 2 provides an overview related to the social interaction and the approach applied in order to model it. Section 3 presents a) the methodology followed to understand the interactions and b) the influences of our interaction model and Section 4 describe the application developed in order to implement the interaction model.

2 Approach to the Simulation of Social Interaction

The main difficulty in creating a human-like character is to model the dynamic of the social, emotional and cognitive dimension of human interaction. The evolution of these dimensions is particularly observed in the face-to-face communication.

The face-to-face communication is made of interchanges between interlocutors. These interchanges are multi-channel and multimodal. In other words, the utterances that are co-produced by the interlocutors are communicated by verbal and nonverbal expressions [2]. These expressions reinforce the objective and intentions of the speech acts and the attitude.

The information conveyed during the face-to-face communication, is generally insufficient so that the addressee of the message could identify the effect looked at by the speaker. The interlocutor frequently uses the context of the statement to reconstruct the sense of what is communicated to him. This problem is treated by relevance theory [3], an inferential theory of communication, which aims to explain how the audience infers the communicator's intended meaning

Our aim is to create an intelligent emotional character which has the capability to communicate their objective and intentions in a believable interaction expressing and modifying its emotional states.

The theoretical background of our approach is related to the concept of situated interaction and constructivist theories. This background emphasizes the role of the context in the interpretation of behavior, and the fact that the meaningful part of the communication is co-constructed by the actors during the ongoing interaction [4].

We combined two complementary epistemological approaches in order to create our model for human-like behavior. We believe that the reproduction of an interaction that makes sense for the tasks and actions that must be achieved by the character's needs of (i) representational approaches for the production and expression of speech acts, and (ii) approaches of situated cognition in order to take into account the role of the context (objects and actors of the environment, their interrelation and history). Both approaches bring complementary aspects to reproduce the dynamic of a human interaction that makes sense for the tasks and actions that must be achieved by the character.

The representational approach enables us to describe the communication acts and emotional states in stable terms. We referred to theories of speech acts, dialogue games and nonverbal expression to deal with social interaction and express communication acts which transmit intentions.

In order to model the co-construction of the sense of the interactions, we referred to the social interaction theory of Goffman [5] and the ostensive-inferential communication theory developed by Sperber [3].

The Sperber theory helped us to understand how communicators succeed in understanding each other. It is because of any stimulus can be potentially interpreted against a huge range of contextual assumptions. Thus, an ostensive-inferential communication consists in intentionally providing evidence to an audience to make manifest a piece of information in order to arrive at certain conclusions. To sum up, the non-representational approaches were used to improve the behavior of the model in terms of context dependency.

We integrate these approaches in the conception of a social interaction model for the behavioral animation of ECA.

We will emphasize the role of the social interaction and its related feeling that we will call *social immersion*. This kind of immersion is strictly connected with the socio-cultural focus of presence proposed by Mantovani and Riva [6, 7]. According to these authors, experiencing presence and telepresence does not depend so much on the reproduction of the physical reality as on the capacity of simulation to produce a context in which social actors may communicate and cooperate.

3 Model of Interaction's Dynamics

We are interested in the cognitive couplings between actors in situation of co-presence. These couplings are socially and emotionally situated. That is to say, the character must in real time have the capacity to adjust simultaneously their facial and bodily emotional expressions within the context.

In order to understand this kind of problematic situation and to produce credible social interactions we referred to concepts of ethnomethodology [4, 8]. We put into practice an ethnomethodological analysis of interactions based on a video recording of the rehabilitation sessions. This analysis helped us to understand how interactions were context-dependent, and to structure and formalize the dynamic of the interaction.

We use the video in order to apply two approaches of the communication's analysis: (1) a discourse analysis in order to identify the different kind of speech acts and (2) a conversational analysis with the intention to discover the social interactions rules, the non verbal communication expressions (like gestures and body movements) and paraverbal activities (tone, pitch and pace of the voice).

These analyses show us the nonverbal production of the dialogue, the cultural differences in the expression of social interactions rules, the management of turn-taking, as well as the role of the gestures and indexical expressions.

These analyses enabled us to categorize the utterances of dialogue in phases, to identify their objectives and intentions, e.g. to order or to inform in order to guide, as well as their relationship among intentions, personality and the emotional state.

In order to elaborate on a model of the dynamics of the interaction we design a dialogue model to select the objectives and intentions of the therapist and a model for the update of attitude (see Fig. 1).

The model of dialogue management refers to the cognitive-emotional model for the selection of the communication acts. The model updates the state of the dialogue and selects the acts of language which express the objective (perlocutionary acts) e.g. to guide, to correct or to encourage) and intention (illocutionary acts) e.g. to order or to inform in order to guide of the character.

In order to understand the dynamic of objective and intentions, this model also refers to the Sperber notion about the communicator which is attributed two intentions: an intention to inform the hearer of something, which is called the *informative intention*; and the intention to inform the addressee of this informative intention, this latter intention being called the *communicative intention* [3].

The attitude and concept of emotional intelligence [9] are immersed in the dialogue model. They influence the selection of intentions, and the update of the character's goals.

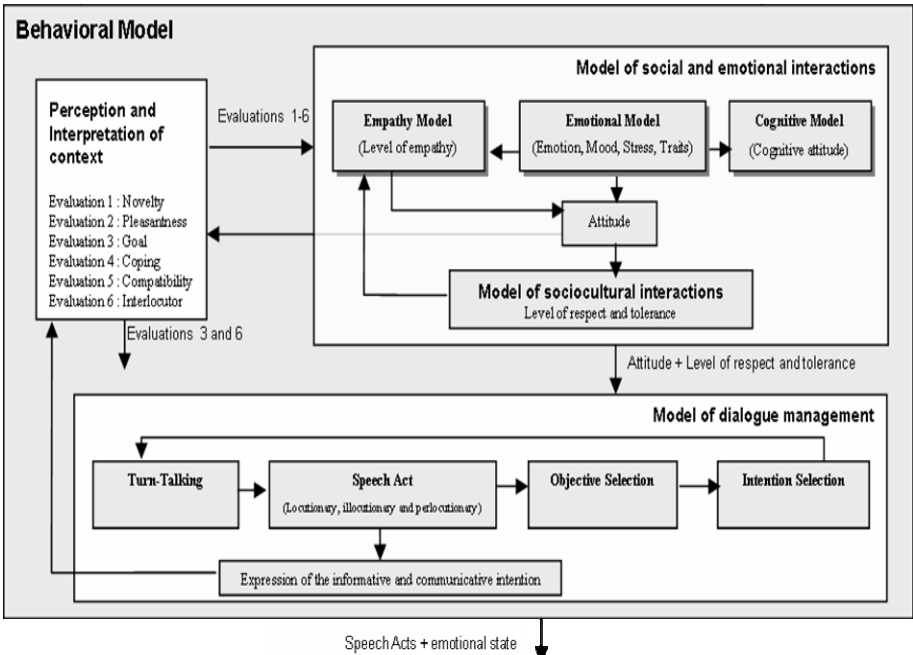


Fig. 1. The Behavioral model produces the dynamics of the interaction. This is expressed by the attitude and communication acts.

We defined the attitude as the predisposition to react in a positive or negative way to a stimulus, according to the feelings and ideas that it evokes. The attitude influences the selection of the acts of communication in the dialogue and defines a strategy for the dialogue management.

We describe the character's attitude in terms of four factors: (1) the personality features: openness, conscientiousness, extraversion, agreeableness, neuroticism, (2) the empathy level, (3) the stress level and (4) the mood.

The attitude is influenced by 1) the profile (traits of personality, culture, status and gender) and internal state of the character (cognitive state, emotional state, mood and empathy and stress level), 2) the perception and interpretation of the context (interlocutor actions and attitude) and 3) the interaction's games: dialogue games (objectives and intentions of character's task) and communication games (protocol of dialogue).

In order to express an attitude, the behavioral model manages the perception of the context and two models that delineate the behavior during the interaction: the model of social and emotional interaction and the model of management of the dialogue.

The social and emotional interaction model determines and updates the emotional state of the character (basic emotions such as joy, sadness, fear and anger), his mood, his level of stress, the personality traits and his level of respect and tolerance. This model is constructed on emotional and social dimensions that characterize the human being. In order to represent the emotional expression of behavior we had to adapt theories of emotions and personality in order to propose a dimensional model of emotions [10]. This model is influenced by the traits of personality (Five-Factor Model) [11] and the Scherer appraisal theory [12].

4 Social Interaction in Virtual Therapy

The behavioral model is implemented through a virtual character in the context of virtual therapy. We have developed in collaboration with the INSERM¹ a virtual simulation environment for providing therapy to the rehabilitation of the active extension of the wrist to brain injured people caused by CVA (see Fig. 2).



Fig. 2. During the therapy session, the patient sees at the same time his clone's arms and his virtual therapist. The activity of virtual therapist is based on the experimental protocol as well as on the actions and attitude of the character that represent the patient.

Virtual reality technology has been used for several decades for a variety of psychosocial applications [13]. Concerning therapy applications, virtual reality makes possible the creation of an interactive environment, where the exercises could be controlled, quantified and modulated, reinforcing the cognitive abilities of the patient and minimizing the cost of rehabilitation.

Our virtual environment, shows at the same time the patient clone's arms and a virtual therapist (see Fig. 2). The patient receives instructions concerning the exercise from the virtual therapist (verbal, paraverbal and nonverbal). The activity of the virtual therapist consists of guiding the patient in his exercises and encouraging and correcting him. An example of the utterances is the following : "Follow me, raise your left hand", "Let's go!", "Please, try again!", "No!!".

¹ French National Institute of Health and Medical Research.

The behavioral animation of the virtual therapist takes into account an emotional and intentional model of the patient–therapist relationship. For example, according to the traits of personality and the development of the session, the virtual therapist could be neutral or empathic, directing or suggestive, attentive or detached, etc. Table 1 shows an example of speech acts that could be selected by the dialogue model according to the attitude, objective and intention of the character.

Table 1. Examples of Speech Acts Characterization, according to the Attitude, Objective and Intention of the Character

Locutionary Act	Attitude	Perlocutionary Act	Illocutionary Act
Raise your left hand, please	Empathic	To guide	To inform
Follow me, raise your left hand	Empathic	To guide	To order
Very good!	Empathic	To encourage	To inform
Come On, Let's go	Empathic	To encourage	To order
Please, try again	Directive	To encourage	To inform
No	Directive	To correct	To inform
Do your job	Directive	To guide	To order
Fine, try again	Directive	To encourage	To order
No	Directive	To correct	To order

The preliminary evaluations of the virtual environment show that the users are sensitive to the fact that the therapist expresses signs of life during the pauses: movement of the eyes and head, modification of posture, breathing, etc.

We observed that the coherence of the actions of the therapist compared to the context seem important in the simulation of social interaction. The user expects an evolution of therapist behavior according to their performances. He expects to be corrected if he carries out a bad movement. If the therapist does not react whereas the user even perceives his own dysfunction, he can interpret this non-response as a mark of carelessness.

We also noticed the importance of a coupling between the nonverbal and paraverbal expression. For example, if the prosody of the locutionary act transmits dissatisfaction, the facial expression must be coherent.

5 Conclusion

In this paper, we present our approach to create an intelligent emotional character capable of communicate their objective and intentions. This character models credible social interactions in a situated context : the motor rehabilitation of post-CVA patients.

The preliminary evaluations provided by the users during their interaction with the virtual therapist enable us to get some conclusions about the simulation of social and emotional interactions. We thought that a virtual character able to produce and maintain a sense of social and emotional immersion could to improve the advantages of software application, and approach a believable human-computer interaction.

References

1. Cassell, J., et al. (eds.): *Embodied Conversational Agents*, p. 426. MIT Press, Cambridge (2000)
2. Cosnier, J.: *Communication non verbale et langage*. *Psychologie Medicale* 9(11), 2033–2047 (1977)
3. Sperber, D., Wilson, D.: *La Pertinence, communication et cognition*, p. 397. Les editions de minuit, Paris (1989)
4. Garfinkel, H.: *Studies in Ethnomethodology*. Cliffs, E. (ed.), Prentice-Hall, NJ (1967)
5. Goffman, E.: *Les rites d'interaction. Le sens commun.*, p. 236. Les editions de minuit, Paris (1974)
6. Riva, G., Waterworth, J.A.: *Presence and the Self: A cognitive neuroscience approach*. In: *Presence-Connect.*, vol. 3 (2003), <http://presence.cs.ucl.ac.uk/presenceconnect/articles/Apr2003/jwworthApr72003114532/jwworthApr72003114532.html>
7. Mantovani, G., Riva, G.: "Real" presence: How different ontologies generate different criteria for presence, telepresence, and virtual presence. *Presence: Teleoperators and Virtual Environments* 8(5), 538–548 (1999)
8. Garfinkel, H., Sacks, H.: *On formal Structures of Practical Actions*. In: Tiryakian, J.M.a.E. (ed.) *Theoretical sociology: Perspectives and developments*. Appleton Century Crofts, New York (1970)
9. Salovey, P., Mayer, J.D.: *Emotional Intelligence. Imagination, Cognition and Personality* 9(3), 185–211 (1990)
10. Morales Rodríguez, M.L.: *Modèle d'interaction sociale pour des agents conversationnels animés. Application à la rééducation de patients cérébro-lésés*. PhD Thesis. Toulouse, Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier, 108 p. (2007)
11. McCrae, R.R., John, O.P.: *An introduction to the Five-Factor Model and Its Applications*. *Journal of Personality* 60, 175–215 (1992)
12. Scherer, K.R.: *Toward a dynamic theory of emotion: The component process model of affective states*. *Geneva Studies in Emotion and Communication* 1(1), 1–72 (1987)
13. Riva, G. (ed.): *Virtual Reality in Neuro-psycho-physiology Cognitive, clinical and methodological issues in assessment and treatment*. In: *Studies in Health Technology and Informatics*, vol. 44, p. 220. IOS Press, Amsterdam (1997)

Ontology-Based Approach for Semi-automatic Generation of Subcategorization Frames for Spanish Verbs

Rodolfo A. Pazos R., José A. Martínez F., Javier González B.,
María Lucila Morales-Rodríguez, Gladis M. Galiana B., and Alberto Castro H.

Instituto Tecnológico de Cd. Madero
Av. 1o. de Mayo esq. Sor Juana Inés de la Cruz s/n, 89440, Cd. Madero, Mexico
{r_pazos_r, gmgaliana}@yahoo.com.mx,
{jose.mtz, lmoralesrdz}@gmail.com,
{jjgonzalezbarbosa, a_castro_h}@hotmail.com

Abstract. This work deals with the semi-automatic generation of subcategorization frames (SCFs) of Spanish verbs; specifically, given a set of verbs in Spanish and their respective sense, their SCFs are obtained. The acquisition of SCFs in Spanish has been approached in different works: in some the frames are generated manually, while in others they are obtained semi-automatically from a tagged corpus; unfortunately in this case, the results depend on the characteristics of the texts used. The method proposed in this document combines an ontology-based approach (through lexical relations of verbs) and linguistic knowledge (functional class of verbs). The relations among base verbs and other verbs were obtained from the Spanish WordNet ontology, which contains lexical relations among words. Also, the existing relation between the SCF and the functional class of verbs was used to generate the SCFs. In order to evaluate the method the SCFs for 44 base verbs were generated manually, from which 239 SCFs were semi-automatically generated and validated, yielding an accuracy of 89.38%.

1 Introduction

Most Natural Language processing applications use syntactic parsing. There exist several theories for parsing. The most used ones are constituent parsing and dependency parsing. Some of the most important dependency parsing techniques are based on the Meaning-Text Theory (MTT). The use of MTT allows to describe Spanish characteristics, as the establishment of relations among syntactic and semantic valencies.

Verb subcategorization frames (SCF) represent one of the most important elements of lexical/grammatical knowledge for efficient and reliable parsing [1]. The implementation of an MTT parser requires creating a base of linguistic knowledge that contains the SCFs of verbs that allow to know how words are related in a sentence.

At the Centro Nacional de Investigación y Desarrollo Tecnológico a syntactic parser for a natural language interface was developed, which uses an MTT dependency parser. This parser uses a base of SCFs for verbs, which does not contain enough linguistic information for parsing.

The parser [2] uses a base of SCFs as linguistic knowledge for filtering syntactic structures and discarding invalid ones. The database of the linguistic knowledge for the SCFs is implemented in PostgreSQL 7.2.1 and consists of five tables: Verbs, Classes, Sentences, SCFs and Combinations. Figure 1 shows the tables of the database of SCFs and their relations.

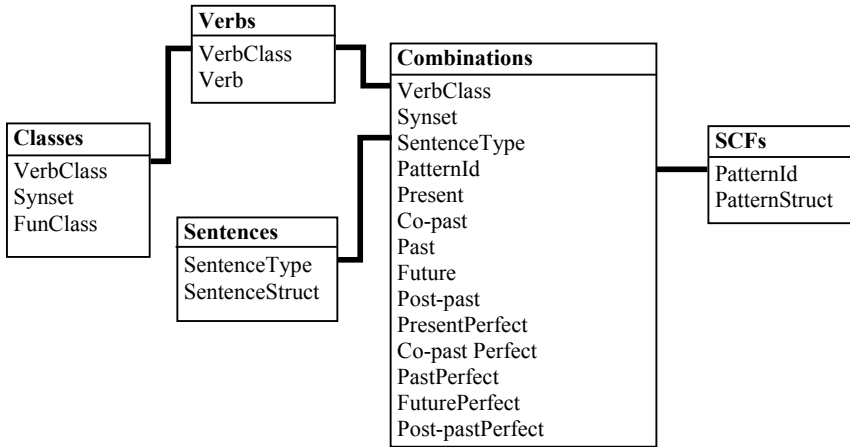


Fig. 1. Schema of the database of subcategorization frames

2 Previous Work

In [1] a statistical-based mechanism for automatically acquiring SCFs and their frequencies based on tagged corpora is described. The authors showed that combining syntactic and statistical analysis, SCF frequencies can be estimated with large accuracy. Unfortunately, the frequencies of just six SCFs were measured, but they claim that their method is general and can be extended to much more SCFS.

[3] presents a cooperative machine learning system, ASIUM, which is able to acquire SCFs with restrictions of selection and ontologies for specific domains from syntactically parsed technical texts in natural language. Experiments on corpora of cooking recipes in French, and patents in English, have shown the applicability of the method to texts in restricted and technical domains and the usefulness of the cooperative approach for such knowledge acquisition.

In [4] a linguistic-based work is presented that proposes a method for adding semantic annotation to verb predicate arguments for a lexicon called Penn English TreeBank. The work relies considerably on recent works on linguistics about word classification that has more semantic orientation, such as the Levin classes and WordNet. The document mentions that 1000 SCFs have been manually generated, and hints at the possibility of automatic frame generation; for example, it mentions that a large number of SCFs for repetition or negated actions (verbs) can be generated (e.g., state/restate, load/unload).

In [5] a linguistic-based work is presented, which proposes an extension of 57 new classes to Levin's verb taxonomy. An experiment was carried out in the context of automatic SCF corpus-based acquisition for determining the usefulness of the new classes proposed, which yielded an accuracy of 71.0% (the percentage of SCF types that the system proposes which are correct). Though several classifications are currently available for English verbs, they are restricted to certain class types and many of them have few instances in each class. The document mentions that some experiments recently reported indicate that it should be possible in the future to automatically supplement existing classifications with original/new verb classes and member verbs from corpus data.

In [6] a probabilistic classifier (based on C5.0) is described, which can automatically classify a set of verbs into argument-structure classes with a reasonable error rate (33.4%). The method exploited the distributions of selected features from the local context of the verb, which were extracted from a 23-million word corpus from the Wall Street Journal.

In [7] a method is explained, which uses unsupervised learning to learn equivalence classes for verbs in Spanish. Specifically, the approach consists of using the Expectation Maximization algorithm for clustering verbs according to their contexts of occurrence, which would permit extrapolating the behavior of known verbs to unknown ones. The authors claim having found a good clustering solution that distinguishes verbs with clearly different SCFs.

3 General Description of the Method

The method proposed for generating SCFs consists of two phases:

- Generate manually the SCFs of a group of verbs/sense denominated *base verbs/sense*. The steps carried out for generating manually the SCFs of the verbs/sense were the following:
 - Select a verb/sense from the base verbs/sense set.
 - Generate manually the SCF of the selected verb/sense according to its meaning.
 - Create the syntactic structure of a sentence with respect to the verb, according to the examples presented in the gloss (if it exists).
 - Check the SCF of the verb/sense against sentences in the corpus published by the Real Academia Española [8].
 - Verify the SCFs by an expert in linguistics.
- Using the SCFs of the base verbs/sense, automatically obtain the SCFs of other verbs/sense with related meaning.

For developing a filling program based on the preceding method, two programs were designed: the *Automatic Overall Filling Sub-module*, which uses the base verbs/sense to fill the base of SCFs automatically for all the verbs; and the *Automatic Levelwise Filling Sub-module*, which uses a file that contains a verb/sense that considers as base, and obtains the SCFs of the hyponyms of the verb/sense. Both algorithms use the *Verb Verification Submodule*, which verifies that the verb/sense exists in the Spanish WordNet ontology. If it exists, it is processed by the *Querying, Assignment and Frame Elaboration Sub-module*, for generating its SCF.

4 Lexical Relations

In order to relate the base verbs/sense with other verbs, it was necessary to obtain the lexical relations that exist among them. To this end the hyponymy relation (has_hyponym) contained in the Spanish WordNet ontology 1.0 was used. This relation allows to connect synsets of the ontology in a hierarchy, which is important to obtain the SCFs.

For example, Figure 2 depicts the semantic relation of verbal hyponymy of the verb/sense *respirar*₁ (breathe). We hypothesize that if one knows the SCF of the verb/sense *respirar*₁ (breathe), one can generate the SCF of the verb/sense *espirar*₁, *exhalar*₁ (breathe out, exhale), because this verb/sense is hyponym of *respirar*₁. Also, if one knows the SCF of this verb/sense, one can generate the SCF of the verb/sense *soplar*₁ (blow) for the same reason. And this one, in turn, will serve to generate the SCF of the verb/sense *resoplar*₂ (puff).

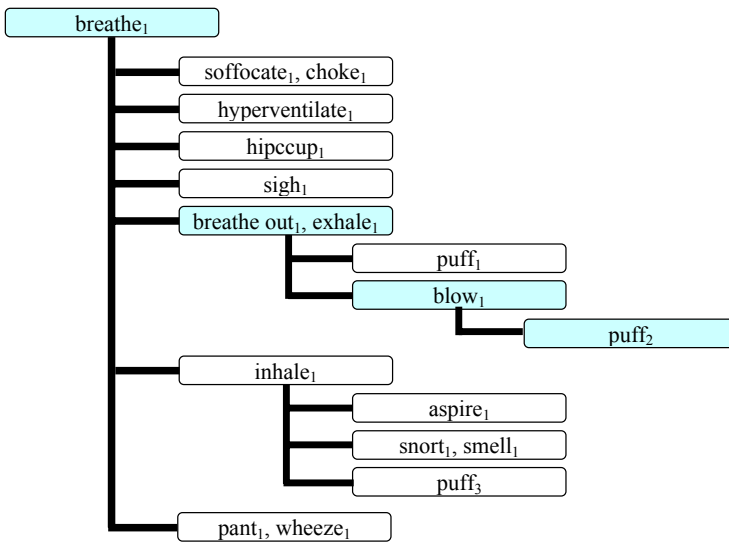


Fig. 2. Example of lexical relation of verbal hyponymy in Spanish WordNet 1.0

5 Subcategorization Frames of Base Verbs/Sense

In Spanish WordNet there exist 366 synsets in the highest level of the ontology. It is important to mention that each synset contains one or more verbs/sense. These verbs/sense were denominated base verbs/sense or level-1 verbs/sense. Since the manual generation of more than 366 SCFs of level-1 verbs/sense is an arduous task that requires a lot of time, we worked with a sample of 44 level-1 verbs/sense, which were generated manually.

Figure 3 shows an example of the structure of the SCF for verb inhalar (inhale).

00004020
*inhalar*₁ (inhale)
 (Medicine) X draw in air

X = I
1. SN

C₁₁: SN indicates a nominal syntagm with animate noun.

Combinations	
C _{1,1}	{ <i>Un buen hombre</i> } <i>inhala</i> [, <i>exhala y acomoda sus ideas</i>] { <i>A good man</i> } inhales [, exhales and accommodates his ideas]

Fig. 3. Subcategorization frame of the verb *inhalar* (inhale)

It is important to mention that in this work, we consider that the inclusion of the Spanish preposition *a* (to) in a direct complement, which defines the animacy of the complement, depends on the semantics of the noun that it precedes. Therefore, the verb does not determine if its direct complement includes preposition *a* (to) [9].

6 Experimentation

6.1 Test Cases

The sample of 44 base verbs/sense that was used for the tests, belong to level 1 of the Spanish WordNet ontology. From these, the first test cases were obtained, which correspond to the verbs/sense of level-2 synsets (hyponyms of level-1 synsets), for which their SCFs were generated (derived or assigned). These SCFs served as a basis to obtain the SCFs of the verbs/sense of level-3 synsets (hyponyms of level-2 synsets). This process can be continued up to the last level.

It is necessary to mention that, each time that the generation of the SCFs of the verb/sense of some level begins, it has to be verified that the SCFs of their hypernym synsets are correct.

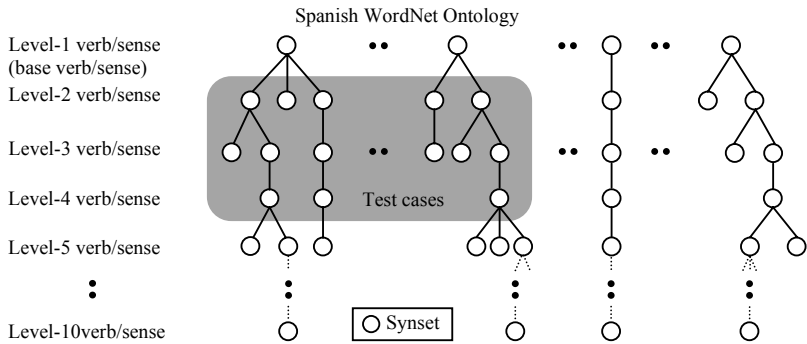


Fig. 4. Test cases

As can be seen in Figure 4, the test cases correspond to the shaded area of Spanish WordNet.

6.2 Experimentation

The experimentation consisted on three tests; and in order to verify that the 195 SCFs were correctly obtained by the SCFs generation method, these were previously generated by hand. The experiments conducted were the following:

1. The first experiment consisted of elaborating or assigning the SCFs for 100 level-2 verbs/sense that are hyponyms of level-1 verbs/sense.
2. The second experiment consisted of elaborating or assigning the SCFs for 60 level-3 verbs/sense that are hyponyms of level-2 verbs/sense. The SCFs for level-2 verbs/sense were previously checked, and corrected for those verbs/sense whose SCFs were incorrectly generated in the first experiment.
3. The third experiment was similar to the second, except that in this case SCFs were generated for 35 level-4 verbs/sense.

6.3 Analysis of Results

In order to asses the effectiveness of the generation method of SCFs, the percentage of correct results was calculated for the SCFs generated.

The percentage of success (PS) of a verb/sense was obtained comparing each one of the possible combinations of the correct SCF with the possible combinations of the corresponding obtained SCF, and using the following formula:

$$PS = \frac{\text{No. of correct combinations generated}}{\text{No. of correct combinations possible}} \times 100$$

In the preceding formula the number of correct combinations generated (NCCG) is divided by the number of correct combinations possible (NCCP). Additionally, the number of incorrect combinations generated (NICG) and the number of over-generated combinations (NOC) were calculated.

Table 1 shows the obtained results, which includes the average of PA and the percentage of NICG for each experiment, and the overall averages for all the experiments.

Table 1. Success percentage from the experiments

Experiment	Average of PS	Percentage of NOC
1	81.00%	23.53%
2	90.00%	0.00%
3	97.14%	0.00%
Overall average	89.38%	7.84%

The experiments yielded some errors in the derivation/assignment of the SCFs. The errors found are explained hereupon:

- Errors generated in the elaboration mode:

- Verb/sense without necessary complements. This error occurred when generating a SCF that does not accept Prepositional Complements (PC) or Attributes (Atr) that are necessary for some verbs. For example, the SCF generated for the verb *convertirse*₁ (become) allows the incorrect sentence (5).

(5) [_{Subj} *La plática* (The chat)] [_V *se convirtió* (became)].

Whereas the correct SCF must have a prepositional complement as shown in sentence (6).

(6) [_{Subj} *La plática* (The chat)] [_V *se convirtió* (became)] [_{PC} *en un desastre* (a disaster)].

The SCF of the verb/sense *convertirse*₁ (become) was obtained from the SCF [Subj(NS)] V for the verb/sense *cambiar*₂ (change). This SCF does not include a PC (*en* NS); and therefore, the SCF elaborated for the verb/sense *convertirse*₁ (become) is incorrect.

- Transitive verb/sense with incorrect complement. This error occurred when generating the following SCF for transitive verbs: [Subj(S)] V DC(*a* NIPro). Some complement(s) is(are) added to this SCF as Circumstantial Complement (CC), which is obtained from its hypernym. Such kind of complement is incorrect for some transitive verbs. For example, the SCF generated for the verb/sense *hacer*₁ allows the incorrect sentence (7).

(7) [_{Subj} *Samuel*] [_V *hizo* (made)] [_{DC} *a su novia* (his girlfriend)][_{CC} *bien* (good)].

Whereas the correct SCF accepts an attribute as in the following sentence.

(8) [_{Subj} *Samuel*] [_V *hizo* (made)] [_{DC} *a su novia* (his girlfriend)] [_{Atr} *feliz* (happy)].

The SCF of the verb/sense *hacer*₁ (make) was obtained from the SCF [Subj (NS)] Pro-V CC(Adv) for the verb/sense *cambiar*₂ (change). This SCF includes the DC(Adv) and it doesn't include Atr(Adj); and therefore, the SCF that was elaborated for the verb/sense *hacer*₁ (make) is incorrect.

- Errors generated in the assignment mode:

- Transitive verb/sense with optional Direct Complement. This error occurred when copying the SCF to a transitive verb, whose DC is optional. For example, the SCF generated for the verb *ensanchar*₄ (widen) is incorrect because it allows the absence of a DC as in sentence (9).

(9) [_{Subj} *Los trabajadores* (The workers)] [_V *ensancharon* (widened)].

When the Direct Complement must be mandatory; for example, sentence (10).

(10) [_{Subj} *Los trabajadores* (The workers)] [_v *ensancharon* (widened)] [_{CD} *la carretera* (the highway)].

The SCF of the verb/sense *ensanchar*₄ was obtained from the SCF [Subj(NS)] V DC(NS) for the verb/sense *cambiar*₃ (change). This SCF includes an optional DC; therefore, the SCF assigned to the verb/sense *ensanchar*₄ (widen) is incorrect.

- Verb/sense with incorrect complements. This error occurred when copying the SCF from a hypernym verb to a hyponym verb, whose complement (indirect, prepositional or attribute) is incorrect for the last verb. An example of this error occurs with the verb/sense *fanfarronear*₁ (brag) in sentence (11).

(11) [_{Subj} *Juan* (John)] [_v *fanfarronea* (brags)] [_{CC} *mal* (wrong)].

Whereas the correct form of the previous sentence is (12).

(12) [_{Subj} *Juan* (John)] [_v *fanfarronea* (brags)].

The SCF of the verb/sense *fanfarronear*₁ (brag) was obtained from the SCF [Subj(NS)] V DC(Adv) for the verb/sense *actuar*₁ (act). This SCF includes the DC(Adv); and therefore, the SCF that is assigned to the verb/sense *fanfarronear*₁ is incorrect.

- Verb/sense without necessary complements. This error occurred when copying a SCF to a verb, which does not include Prepositional Complements (PC) or Circumstantial Complements (CC), which are necessary for some verbs. For example, the SCF generated for the verb/sense *comportarse*₂ (behave) does not allow the correct sentence (13).

(13) [_{Subj} *Luis*] [_v *se comportó* (behaved)] [_{CC} *como loco* (as a lunatic)].

The SCF of the verb/sense *comportarse*₂ (behave) was obtained from the SCF [Subj(NS)] V CC(Adv) for the verb/sense *actuar*₁ (act). This SCF does not include the CC(*como* NS|Adv); and therefore, the SCF that is assigned to the verb/sense *comportarse*₂ (behave) is incorrect.

7 Conclusions

A total of 239 SCFs of Spanish verbs were generated manually and validated. A program was developed based on the proposed method for SCF generation. The results obtained on the generation of the SCFs of verbs/sense in three levels (level 2, level 3 and level 4) of the Spanish WordNet ontology yielded an average success of 89.38% and an average percentage of 7.84% for over-generated combinations, which compares favorably with the 71.0% reported in [5].

The experiments revealed that the percentage of success of the SCFs generated improves as the level of depth in the ontology increases. A possible explanation for this

behavior is that the meaning of these verbs makes it possible that less verb valencies are required.

References

1. Ushioda, A., Evans, D.A., Gibson, T., Waibel, A.: The Automatic Acquisition of Frequencies of Verb Subcategorization Frames from Tagged Corpora. In: SIGLEX ACL Workshop on The Acquisition of Lexical Knowledge from Text, Columbus, Ohio, pp. 95–106 (1993)
2. Cervantes, A.: Diseño e Implementación de un Analizador Sintáctico para las Oraciones en Español Usando el Método de Dependencias. Tesis de maestría. Centro Nacional de Investigación y Desarrollo Tecnológico (2005)
3. Faure, D., Nedellec, C.: Knowledge Acquisition of Predicate Argument Structures from Technical Texts Using Machine Learning: The System ASIUM. In: Proc. of the 11th European Workshop on Knowledge Acquisition, Modeling and Management, pp. 329–334 (1999)
4. Kingsbury, P., Marcus, M., Palmer, M.: Adding semantic annotation to the Penn Tree-Bank. In: Proc. of the Human Language Technology Conference (HLT), San Diego, CA (2002)
5. Korhonen, A.: Assigning Verbs to Semantic Classes via WordNet. In: Proc. of the SemaNet 2002: Building and Using Semantic Networks, Taipei, Taiwan, pp. 1–7 (2002)
6. Sarkar, A., Tripasai, W.: Learning Verb Argument Structure from Minimally Annotated Corpora. In: Proc. of the Int. Conf. on Computational Linguistics, Taipei, Taiwan, pp. 1–8 (2002)
7. Castellón, I., Alemany, L.A., Tincheva, N.T.: A Procedure to Automatically Enrich Verbal Lexica with Subcategorization Frames. *Revista Iberoamericana de Inteligencia Artificial* 12(37), 45–53 (2008)
8. Real Academia Española, Banco de datos (CREA), Corpus de referencia del español actual (2006), <http://www.rae.es>
9. Galicia, S.: Análisis Sintáctico Conducido por un Diccionario de Patrones de Manejo Sintáctico para Lenguaje Español. Tesis doctoral. Centro de Investigación en Computación, Instituto Politécnico Nacional (2000)

Diffusion of Domestic Water Conservation Technologies in an ABM-GIS Integrated Model

José M. Galán¹, Ricardo del Olmo¹, and Adolfo López-Paredes²

¹ Universidad de Burgos – INSISOC Group, C/ Villadiego s/n
09001 Burgos, Spain
{jmgalan, rdelolmo}@ubu.es

² Universidad de Valladolid – INSISOC Group, Paseo del cauce s/n,
47011 Valladolid, Spain
adolfo@insisoc.org

Abstract. The aim of this work is to analyze the effect of adoption of domestic water conservation technologies under different assumptions and scenarios in emergent metropolitan areas. For that purpose, we present an agent based model linked to a geographic information system to explore the domestic water demand in the Valladolid metropolitan area (Spain). Specifically, we adapt and integrate diverse social submodels –models of urban dynamics, water consumption, and technological and opinion diffusion– with a statistical model that characterizes the agents’ water consumption behaviour. The results show the temporal impact on water consumption and technological adoption of several parameters, and the importance of the joint effect of both diffusion mechanisms.

Keywords: agent based modelling, GIS-ABM integration, domestic water management, computer simulation, water conservation technologies.

1 Introduction

Agent-based modelling (ABM) is one of multiple techniques used to conceptualise social systems. This methodology is characterised by a direct correspondence between the entities in the system to be modelled and the computational agents that represent such entities in a computer model.

During the last two decades, this technique has become a recognised research tool in a wide range of scientific disciplines, *e.g.* Economics [1], Ecology and Resource Management [2], Political Science [3], Anthropology [4], Computer Science [5] or Sociology [6]. Regardless of some drawbacks associated to its use, this expansion phenomenon is not surprising. ABM offers alternative and supplementary insights in many scientific domains. It enables us to incorporate heterogeneity, participatory processes, explicit spatial topologies, model integration or agent’s decision in multiple levels [7], all of them attractive modelling characteristics.

Natural resource management and specifically water management, is one of the fields where ABM has been more widely used [2]. The influence of the social dimension in this domain has forced to integrate some socioeconomic aspects in the models.

Precisely, due to the holistic nature of the methodology, ABM is considered a suitable tool to include explicitly the human interaction [8]. As a matter of fact, the increasing awareness of the significant role of these factors in urban water use has given as consequence several ABM models focused on the topic –*e.g.* on the Thames in UK [9–10], on Barcelona [11], on Thessaloniki [12], and also more general models as those developed by Tillman [13] or Kotz and Hiessl [14] –.

Given that the territorial model is one of the factors with more impact in domestic water consumption, many of the developed models include an explicit representation of the environment. However, most of these representations are often based on very simple cellular automata. In order to capture the spatial influence in the consumption in a realistic fashion, in this work we have integrated an agent based model with a vector based geographic information system (GIS). The developed model enables us to analyse some aspects that are very difficult to model with other traditional forecasting methods of water demand, especially those that involve integration and spatial interaction.

Within this framework, we have parameterised the model in the metropolitan area of Valladolid (Spain), using statistical consumption models of the families in the region to characterize the water consumption behaviour of the agents. We have explored the influence of the adoption of technological water saving devices under different scenarios.

The paper is organized as follows: in the next section, we present the general structure of the model and its different submodels. Thereafter we point out the most relevant aspects of the parameterisation process. Subsequently we define the analysed scenarios and discuss the obtained results, and finally, conclusions are presented in the last section.

2 Model Structure

The model comprises different subcomponents that capture diverse socioeconomic aspects with influence in domestic water demand in metropolitan areas. Two types of entities are represented in the model: the environment and the agents. The environment, imported from a GIS, represents each one of the buildings and dwellings in a geographic area. Computational agents are placed on this layer representing families. The location of the agents is done according to the socioeconomic information retrieved from the georeferenced databases of the region.

This infrastructure is used to integrate four different submodels (Fig. 1): a statistical model that characterizes the agents' behaviour regarding to domestic water demand, linked to an agent based model (hybrid approach) that, in turn, includes models of urban dynamics, water consumption, and technological and opinion diffusion.

The urban dynamics model captures the differences in consumption caused by the change of territorial model in some metropolitan areas: the migratory movement of families in the metropolitan space from the compact city to the diffuse city and vice versa. The opinion and behaviour diffusion models are incorporated to model the effect of social education and awareness campaigns and the influence of social imitation and media pressure. Another aspect that may be important in the urban water demand is the role of technology. Since the adoption and diffusion of any new

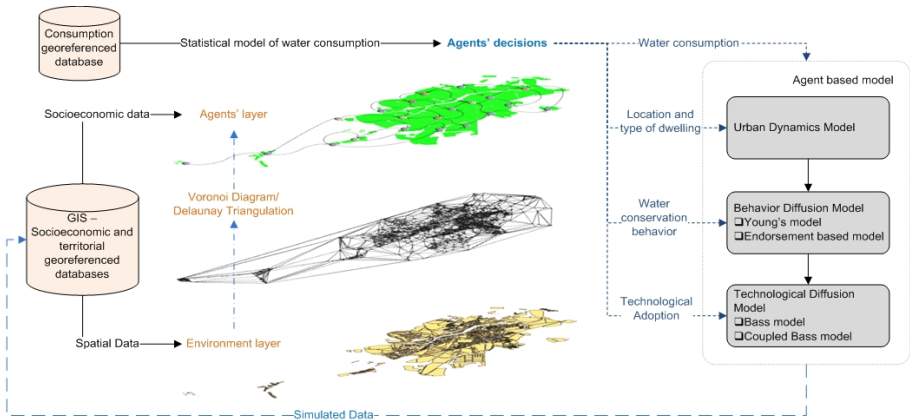


Fig. 1. General structure of the model

(potentially water saving) technological device is not an immediate process, and it depends on agents' interactions, we have included a technological diffusion model for assessing its influence.

The urban dynamics model is based on Benenson's model [15-16]. The foundation of his model lies in the stress-resistance hypothesis as the dynamic engine of urban movement. The model assumes that the migration decision of the agents depends on the dwelling's features and on the socioeconomic characteristics of the neighbouring population.

His original model [15] is defined on a cellular automaton (CA), a type of bottom-up model often used in social sciences. These models can be considered as multiagent systems based on local interaction structures. Nevertheless, this approach is sometimes perceived as too idealised and abstract. Moreover, one of the most restrictive hypotheses of many CA models is to be based on regular rectangular grids, which implicitly assume geographic homogeneity, when very often the interesting issue is to analyse the effect of the spatial heterogeneity of real contexts.

In order to generalize Benenson's model to irregular space distributions and to integrate it with a GIS, we have used the concept of adjacency of Voronoi polygons. In a similar way as Benenson's adaptation [16], the generator points considered to tessellate the space are the centroids of the building blocks. In our model two dwellings are neighbours if they belong to the same building block or if they have a common edge in the Voronoi diagram (or are joined in the Delaunay triangulation, dual problem of the Voronoi diagram) and the distance between their centroids is below certain threshold.

As behaviour diffusion model we have included a reversible stochastic model based on Young's model [17], model that has been adopted and adapted by Edwards et al. [18] to be used in water management. We have included an alternative adaptation of the model based on the endorsement mechanism [19]. This mechanism enables us to assess the effect of heterogeneity in intentions, influences and motivations among the agent population. The endorsement system [19] gives different weight to the different information sources used by the agents to decide their behaviour related

to the resource consumption. Thus, we will have global sourced, local sourced or self sourced agents.

At last, we have integrated a non-reversible technological diffusion model in the process. For this purpose, we have adapted de Bass model [20], model usually expressed as a set of differential equations, to be included as a discrete event mechanism in the general agent based model dynamics. To this effect we have followed the proposal suggested by Borschev and Filippov [21] to translate system dynamics models in ABM. We have included the model in two different versions, as an independent model and as a model coupled with the behaviour and opinion diffusion model. In this latter case we assume an additional probability of adoption in the population with environmentalist behaviour if the technology means water consumption savings.

3 Model Parameterisation

The urban infrastructure, the environment of the model, is constructed using GIS information provided by the Centre of Geographic Information of the Valladolid City Council (2006) and included in its Urban Development Plan (2003). The socioeconomic characteristics of the population are retrieved from the 2006 municipal register of the Valladolid City Council.

The statistical model of the region has been derived from data of demand per plumbing connection to the water net. Since the model is focused on the domestic water, we have just used information about residential demand, ignoring commercial and public uses.

The consumption per person has been calculated by a multiple linear regression, considering as independent variable the water demand and as dependent variables the socioeconomic data of the population and the typology of households where they live. We have used a stepwise method based on F-tests for variable selection with $F \leq 0.05$ as inclusion criteria and $F \geq 0.10$ as exclusion criteria, obtaining a model that just depends on three categorical variables (the different typologies of dwellings) with an adjusted- R^2 of 0.837, and maximum p-value of 0.001 for the significance of the independent variables and the overall regression.

4 Results: Effect of the Technological Diffusion

Since this model is complex, with many parameters, it is very difficult (in practice almost impossible) to analyse it completely for every possible parameter combination. In any case, this is not the objective of the model. The model is aimed at providing a tool that enables to understand some aspects about the global behaviour of the system in those situations that can be considered of greater interest (because of their plausibility, their criticalness or any other reason that justifies a detailed analysis), the scenarios.

We have scaled down the simulations. Thus, even though there are about 125,000 families in the case study, our simulations have been run with 12,500 agents. The timeframe of the simulation is 10 years. One time-step in the simulation represents three months basically because the consumption data is given with this frequency.

To show the potential of the approach we have analysed three scenarios. Each scenario is defined by the assumptions that determine the behaviour of the urban dynamic submodel. In each one, we study the effect of the different parameters and policies with the rest of the integrated components of the model. The first scenario assumes minimum exogenous interference in the urban infrastructure, providing a benchmark against which the rest of scenarios will be compared. The second scenario analyses the impact on water consumption of a constant flow of immigrant population to the metropolitan area. The third has been created to investigate the effect of housing price inertia in city centres of Spanish metropolitan areas.

In the present article we analyse the effect of the adoption and diffusion of a water saving (subtracting) technological device as a low flow shower head under different diffusion models and for several sets of parameters.

Firstly, we have analysed the effect of the technological innovation on water consumption under different profiles of adoption (slow, medium and fast profiles) within the Bass model [20] framework. We have considered that the diffusion process begins in the first time step. The obtained results are consistent with the results expected by the differential Bass model. The evolution of the adoption follows the classical S-Shape function as predicted by theory, of course where the speed of diffusion is clearly affected by the profile (see series 1, 8 and 15 in Fig. 2).

The integration of the Bass model with the rest of the model is more interesting when we analyse the joint effect of the technological diffusion spread together with a behaviour and opinion diffusion process based on a local social network. Thus, an increase in pressure about the awareness of the resource (modelled by the parameter e_s in the diffusion models [17-18]) may affect to the technological adoption, making unnecessary specific high adoption profiles for technological innovation.

In other words, it may happen that the general effect of raise the consciousness of population about the resource could indirectly affect the adoption of water saving technologies, even if the specific media pressure about this particular measure is not very high. To verify our hypothesis we have design a set of simulations with different parameters and under the assumption of the coupled Bass model. We have analysed in each scenario the effect of three profiles of adoption (slow, medium and fast) under both diffusion models (the Young model and the endorsement based model), and in each case with values of e_s (awareness pressure parameter) of 0.0, 0.4 and 0.8. The average results are shown in Fig 2.

Simulations show a change in the convexity of adoption curves under the assumption of a joint effect of both diffusion models. The opinion and behaviour diffusion models cause high proportion of population with environmentalist behaviour and hence a higher rate of technological adoption. This fact implies a double positive effect: a direct effect of higher adoption, but also a second indirect effect through the imitation contagious rate of the Bass model.

In the same figure we can observe that the higher rates of adoption are those with higher value of e_s for each behaviour diffusion model. This effect can be explained assuming that higher values of the parameter imply a higher adoption of environmentalist behaviour and hence higher chance to adopt the technological innovation. This

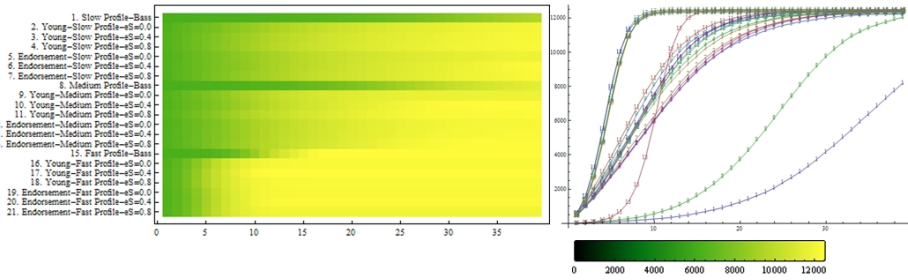


Fig. 2. Temporal evolution of the number of adopters in the Bass and the coupled Bass models as a function of different parameters. The picture on the left represents the number of adopters in a colour scale.

fact, in turn, generates a higher speed of adoption by imitation. We also can notice that the rate of adoption is higher under the Young-Edwards model than under the endorsement based model, and this effect is more severe for higher values of e_s . A possible explanation is that the higher inertia generated by the self sourced behaviour in the endorsement based model can also slow down the adoption process.

Figure 3 shows the average savings percentages corresponding to each profile compared to the scenario without technological adoption. We can observe that except in the cases of slow profiles, the widespread adoption of the saving technology is able to compensate the effects of moderate immigration and territorial change, and even to reduce de consumption in 15% (the accumulated effect would be greater).

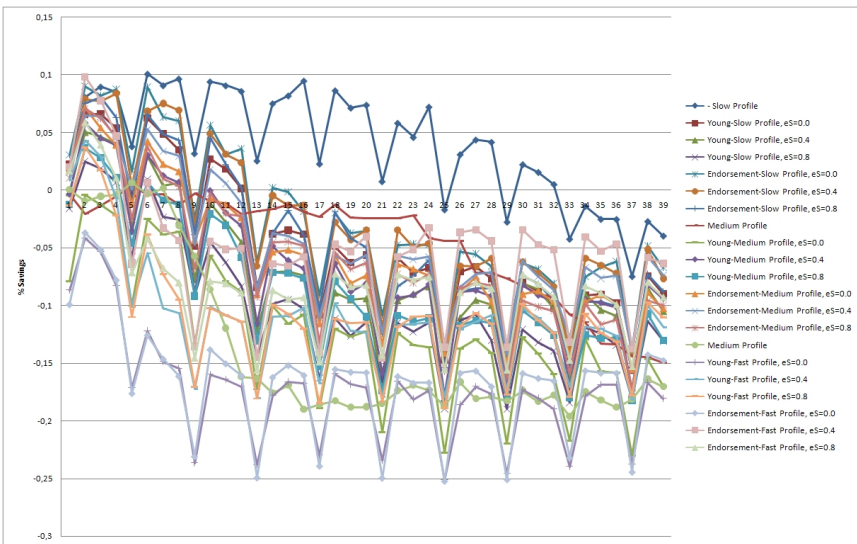


Fig. 3. Temporal evolution of average water savings as consequence of the technological adoption compared to the first scenario without adoption measure

5 Conclusions

In this work we have proved the usefulness of the agent based approach as a tool for modelling and simulation of real systems. In particular, in the context of domestic water management, we have developed an exploring tool that complements the traditional water demand forecasting methodologies, enabling integration of multiple dimensions of the problem and a detailed representation of spatial influences.

Within this framework, we have analysed the effect of diffusion of technological devices with impact in water saving. Simulations corroborate that it is not possible to estimate the saving produced by a domestic conservation technology without linked it to a temporal scale.

We think that the adoption of conservation measures is ruled, in general, by the same diffusion processes than most of the products. Notwithstanding, we also consider that in the particular case of ecological measures, it is possible that the adoption will be influenced by the individual's general attitude toward conservation. In the implemented model we have included this last assumption and we have analysed its influence in the general process. In doing so we have confirmed that there is a coupling between the innovation diffusion model and the behaviour diffusion model (based on local interaction) and hence there is an explicit spatial influence.

The model also shows that the widespread adoption of this type of conservation measures will be able to compensate, in some cases, the increment of water consumption caused, for instance, by changes in the territorial model. Moreover, the necessity of estimating the relative speed of both processes justifies the use of dynamic approaches as agent based modelling.

Acknowledgements. This work is funded by the Spanish Ministry of Education and Science through Projects DPI2004-06590 and DPI2005-05676. We would like to thank Aguas de Valladolid (water Supplier Company of the region), the Valladolid City Council and the Department of Geography of the University of Valladolid for their help and willingness to this research.

References

1. Izquierdo, S.S., Izquierdo, L.R., Galán, J.M., Hernández, C.: Market Failure Caused by Quality Uncertainty. In: Mathieu, P., Beaufils, B., Brandouy, O. (eds.) *Artificial Economics. A Symposium in Agent-based Computational Methods in Finance, Game Theory and their applications*. LNEMS, vol. 564, pp. 203–214. Springer, Heidelberg (2006)
2. López-Paredes, A., Hernández, C.: *Agent-based Modelling in Natural Resource Management*. Insisoc, Madrid (2008)
3. Johnson, P.E.: Simulation modeling in political science. *The American Behavioral Scientist* 42, 1509–1530 (1999)
4. Kohler, T., Gumerman, G.J.: *Dynamics in human and primate societies: Agent-based modeling of social and spatial processes*. Oxford University Press/Santa Fe Institute, New York (2000)
5. Santos, J.I., Galán, J.M., del Olmo, R.: An Agent-Based Model of Personal Web Communities. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) *IDEAL 2006*. LNCS, vol. 4224, pp. 1242–1249. Springer, Heidelberg (2006)

6. Gilbert, N., Troitzsch, K.G.: *Simulation for the social scientist*. Open University Press, Buckingham (1999)
7. Galán, J.M., Santos, J.I., Izquierdo, S.S., Pascual, J.A., del Olmo, R., López-Paredes, A.: Agent-Based Modelling in Domestic Water Management: Barcelona and Valladolid case studies. In: López-Paredes, A., Hernández, C. (eds.) *Agent-Based Modelling in Natural Resource Management*, pp. 93–109. Insisoc, Madrid (2008)
8. Downing, T.E., Moss, S., Pahl-Wostl, C.: Understanding climate policy using participatory agent-based social simulation. In: Moss, S., Davidsson, P. (eds.) *MABS 2000*. LNCS (LNAI), vol. 1979, pp. 198–213. Springer, Heidelberg (2001)
9. Moss, S.: Agent Based Modelling for Integrated Assessment. *Integrated Assessment* 3, 63–77 (2002)
10. Barthélemy, O.: Water demand Policies. In: López-Paredes, A., Hernández, C. (eds.) *Agent-Based Modelling in Natural Resource Management*, pp. 75–91. Insisoc, Madrid (2008)
11. López-Paredes, A., Saurí, D., Galán, J.M.: Urban water management with artificial societies of agents: The FIRMABAR simulator. *Simulation* 81, 189–199 (2005)
12. Athanasiadis, I.N., Mentes, A.K., Mitkas, P.A., Mylopoulos, Y.A.: A hybrid agent-based model for estimating residential water demand. *Simulation* 81, 175–187 (2005)
13. Tillman, D., Larsen, T.A., Pahl-Wostl, C., Gujer, W.: Modeling the actors in water supply systems. *Water Science and Technology* 39, 203–211 (1999)
14. Kotz, C., Hiessl, H.: Analysis of system innovation in urban water infrastructure systems: An agent-based modelling approach. *Water Science and Technology: Water Supply* 5, 135–144 (2005)
15. Benenson, I.: Multi-Agent Simulations of Residential Dynamics in the City. *Computing, Environment and Urban Systems* 22, 25–42 (1998)
16. Benenson, I., Omer, I., Hatna, E.: Entity-based modeling of urban residential dynamics: the case of Yaffo, Tel Aviv. *Environment & Planning B: Planning & Design* 29, 491–512 (2002)
17. Young, H.P.: *Diffusion in Social Networks*. CSED Working Paper No. 2 (1999)
18. Edwards, M., Ferrand, N., Goreaud, F., Huet, S.: The relevance of aggregating a water consumption model cannot be disconnected from the choice of information available on the resource. *Simulation Modelling Practice and Theory* 13, 287–307 (2005)
19. Cohen, P.R.: *Heuristic Reasoning about Uncertainty: an Artificial Intelligence Approach*. Pitman Advanced Publishing Program, Boston (1985)
20. Bass, F.M.: A new product growth for model consumer durables. *Management Science* 15, 215–227 (1969)
21. Borshchev, A., Filippov, A.: From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. In: Kennedy, M., Winch, W.G., Langer, R.S., Rowe, J.I., Yanni, J.M. (eds.) *Proceedings of the 22nd International Conference of the System Dynamics Society*. Systems Dynamics Society, Albany (2004)

Hybrid IT2 NSFLS-1 Used to Predict the Uncertain MXNUSD Exchange Rate

Gerardo M. Mendez and Angeles Hernandez

Instituto Tecnológico de Nuevo Leon
Av. Eloy Cavazos #2001, CP 67170, Cd. Guadalupe, NL, México
gmm_paper@yahoo.com.mx, ahernandez@yturria.com.mx

Abstract. This paper presents a novel application of the interval type-1 non-singleton type-2 fuzzy logic system (FLS) for one step ahead prediction of the daily exchange rate between Mexican Peso and US Dollar (MXNUSD) using the recursive least-squared (RLS)-back-propagation (BP) hybrid learning method. Experiments show that the exchange rate is predictable. A non-singleton type-1 FLS and an interval type-1 non-singleton type-2 FLS, both using only BP learning method, are used as a benchmarking systems to compare the results of the hybrid interval type-1 non-singleton type-2 FLS (RLS-BP) forecaster.

Keywords: Type-2 fuzzy inference systems, type-2 neuro-fuzzy systems, hybrid learning, uncertain rule-based fuzzy logic systems.

1 Introduction

Interval type-2 fuzzy logic systems (IT2 FLS) constitute an emerging technology [1]. As in hot strip mill process [2], [3], in autonomous mobile robots [4], and in plant monitoring and diagnostics [5], [6], financial systems are characterized by high uncertainty, nonlinearity and time varying behavior [7], [8]. This makes it very difficult to forecast financial variables such as exchange rate, closing prices of stock indexes and inflation. The ability of comprehending as well as predicting the movements of economic indices could result in significant profit, stable economic environments and careful financial planning. Neural networks are very popular in financial applications and a lot of work has been done in exchange rate and stock markets predictions described elsewhere [9]–[11].

T2 fuzzy sets (FS) let us model the effects of uncertainties, and minimize them by optimizing the parameters of the IT2 FS during a learning process [12]–[14]. Although some econometricians [7], [8], claim that the raw data used to train and test the FLS forecasters should not be directly used in the modelling process, since it is time varying and contains non-stationary noise, they were used by us to directly train the IT2 FLS. The inputs were modeled as type-1 (T1) non-singletons. The interval type-1 non-singleton type-2 fuzzy logic system (IT2 NSFLS-1) forecaster accounts for all of the uncertainties that are present in the initial antecedents and consequents of the fuzzy rule base.

2 MXNUSD Exchange Rate Prediction

The data used to train the three forecasters cover a period of nine years and four and a half months from 01/02/1997 to 06/17/2006 whereas the test data cover five months from 06/18/2006 to 10/12/2006. The daily closing price of MXNUSD exchange rate was found on the Web site: <http://pacific.commerce.ubc.ca/xr/>. It was a set of $N = 2452$ data, $s(1), s(2), \dots, s(2452)$. It is assumed that the noise-free sampled MXNUSD exchange rate, $s(k)$, can be corrupted by uniformly distributed stationary additive noise $n(k)$, so that

$$x(k) = s(k) + n(k) \quad k = 1, 2, \dots, 2452. \quad (1)$$

and that signal to noise ratio (SNR) is equal to 0 dB. In this application $n(k)$ was fixed to 0. Figure 1, shows the trend of the raw noise-free data $s(k)$. The first 2352 noisy-free data were used for training, and the remaining 100 data were used for testing the forecasters. Four antecedents were used as noise-free inputs, $x(k-3)$, $x(k-2)$, $x(k-1)$ and $x(k)$, to predict the noise-free output, $y = x(k+1)$.

2.1 IT2 NSFLS-1 Design

The architecture of the IT2 NSFLS-1 was established in such a way that its parameters were continuously optimized. The number of rule-antecedents (inputs) was fixed to four. Each antecedent-input space was divided into three FS, resulting in 81 rules. Gaussian primary membership functions (MF) of uncertain means were chosen for the antecedents. Each rule of the IT2 NSFLS-1 was characterized by 12 antecedent MF parameters (two for left-hand and right-hand bounds of the mean and one for standard deviation, for the FS of each of the four antecedents), and two consequent parameters (one for left-hand and one for right-hand end points of the consequent type-1 FS), giving a total of 14 parameters per rule. Each input value had one standard deviation parameter, giving 4 additional parameters.

The resulting IT2 NSFLS-1 used type-1 (T1) non-singleton fuzzification, join under maximum t-conorm, meet under product t-norm, product implication, and center-of-sets type-reduction.

The IT2 NSFLS-1 was trained using two main learning mechanisms: the back-propagation (BP) method for tuning of both antecedent and consequent parameters, and the hybrid training method using recursive least-squared method (RLS) for tuning of consequent parameters as well as the BP method for tuning of antecedent parameters. In this work, the former is referred to as IT2 NSFLS-1 (BP), and the latter as hybrid IT2 NSFLS-1 (RLS-BP).

2.2 IT2 Fuzzy Rule Base

The IT2 NSFLS-1 fuzzy rule base consists of a set of IF-THEN rules that represents the model of the system. This IT2 NSFLS-1 has four inputs $x_1 \in X_1$,

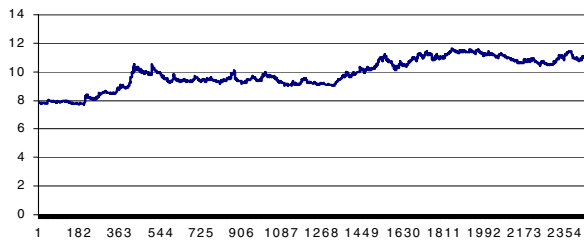


Fig. 1. The daily closing price of MXNUSD exchange rate

$x_2 \in X_2, x_3 \in X_3, x_4 \in X_4$ and one output $y \in Y$. The rule base has $M = 81$ rules of the form

$$R^i : IF \ x_1 \text{ is } \tilde{F}_1^i \ \text{and} \ x_2 \text{ is } \tilde{F}_2^i, \ \text{and} \ x_3 \text{ is } \tilde{F}_3^i \ \text{and} \ x_4 \text{ is } \tilde{F}_4^i, \ \text{THEN} \ y \ \text{is} \ \tilde{G}^i. \quad (2)$$

where $i = 1,2,3,\dots,81$ rules; $\tilde{F}_1^i, \tilde{F}_2^i, \tilde{F}_3^i$ and \tilde{F}_4^i are the antecedent type-2 fuzzy sets, and \tilde{G}^i is the consequent type-2 fuzzy set of rule i .

2.3 Antecedent Membership Functions

The primary MF for each antecedent is an IT2 FS described by Gaussian primary MF with uncertain means:

$$\mu_k^i(x_k) = \exp \left[-\frac{1}{2} \left[\frac{x_k - m_k^i}{\sigma_k^i} \right]^2 \right]. \quad (3)$$

where $m_k^i \in [m_{k1}^i, m_{k2}^i]$ is the uncertain mean, and σ_k^i is the standard deviation, with $k = 1,2,3,4$ (the number of antecedents) and $i = 1,2,\dots,81$ (the number of M rules). Table I shows the values of the three FSs for each antecedent.

Table 1. Intervals of uncertainty for all inputs

	m_{k1}	m_{k2}	σ_k^i
1	7.58	7.78	1.04
2	9.68	9.88	1.04
3	11.68	11.88	1.04

Fig. 2 shows the initial FSs of each antecedent, being the same for all inputs.

2.4 Consequent Membership Functions

The primary membership function for each consequent is a Gaussian with uncertain means, as defined in equation (3). Because the centre-of-sets type-reducer replaces

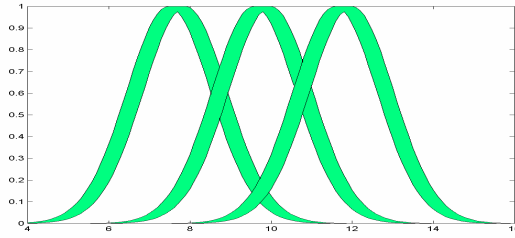


Fig. 2. Three FSs for each of the four antecedents

each consequent set \tilde{G}^i by its centroid, then y_l^i and y_r^i (the M left-points and right points of interval consequent centroids) are the consequent parameters.

The initial values of the centroid parameters y_l^i and y_r^i are such that the corresponding MFs uniformly cover the output space, from 4.0 to 16.0.

3 Modeling Results

One T1 NSFLS and two IT2 NSFLS-1 FLSs were trained and used to predict the daily MXNUSD. The BP training method was used to train both the base line T1 NSFLS and IT2 NSFLS-1 forecasters, while the hybrid RLS-BP method was used to train the hybrid IT2 NSFLS-1 forecaster. For each of the two training methods, BP and RLS-BP, we ran 25 epoch computations; 1138 parameters were tuned using 2352 input-output training data pairs per epoch.

The performance evaluation for the learning methods was based on root mean-squared error (RMSE) benchmarking criteria as in [1]:

$$RMSE_{S,2}(\ast) = \sqrt{\frac{1}{n} \sum_{k=1}^n [Y(k) - f_{nS,2}(\mathbf{x}^{(k)})]^2} \quad (4)$$

where $Y(k)$ is the output from the 100 testing data pairs, $f_{nS,2}(\mathbf{x}^{(k)})$ is the predicted output of the type-1 non-singleton type-2 forecaster, evaluated using the input data $\mathbf{x}^{(k)}$, $k = 1, 2, \dots, 100$, $RMSE_{S,2}(\ast)$ stands for $RMSE_{S,2}(BP)$ [the RMSE of the IT2 NSFLS-1 (BP)], and for $RMSE_{S,2}(RLS - BP)$ [the RMSE of the IT2 NSFLS-1 (RLS-BP)]. The same formula was used in order to calculate the $RMSE_{S,1}(BP)$ of a base line T1 NSFLS (BP).

Fig. 3, shows RMSEs of the two IT2 NSFLS-1 systems and the base line T1 NSFLS for 25 epochs of training. For the T1 NSFLS (BP), there is a substantial performance improvement at epoch two, and after twenty-five epochs of training, the performance is still improving. For the IT2 NSFLS-1 (BP), there is small improvement in performance after five epochs of training. For the case of the hybrid IT2 NSFLS-1 (RLS-BP), its performance still improves after twenty-five epoch of training.

Observe that the IT2 NSFLS-1 (BP) only has better performance than T1 NSFLS-1 during the first twenty epochs of training.

In other hand, after one epoch of training, the hybrid IT2 NSFLS-1 (RLS-BP) has better performance than both T1 NSFLS (BP) and IT2 NSFLS-1 (BP) for all the twenty-five epochs. The hybrid IT2 NSFLS-1 (RLS-BP) outperforms both the T1 NSFLS (BP) and the IT2 NSFLS-1 (BP) at each epoch. The average of the $RMSE_{S,2}(BP)$ is 92.86 percent of the average of the $RMSE_{S,1}(BP)$, representing an improvement of 7.14 percent. The $RMSE_{S,2}(RLS-BP)$ improves the $RMSE_{S,1}(BP)$ by 83.52 percent and the $RMSE_{S,2}(BP)$ by 82.25 percent, as can be observed in Table 2.

Table 2. Performance efficiency of the three fuzzy forecasters

Forecasters	Performance ratio
T1 NSFLS (BP)	1.00
IT2 NSFLS (BP)	7.14
IT2 NSFLS-1 (RLS-BP)	82.25

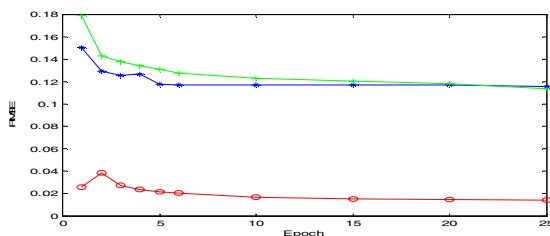


Fig. 3. RMSE errors of the three forecasters - T1 NSFLS (BP) (+), IT2 NSFLS-1 (BP) (*), IT2 NSFLS-1 (RLS-BP) (o)

A comparison between the predicted exchange rate after one epoch and twenty-five epochs of training using T1 NSFLS (BP), and the real price, are depicted in Figures 4 and 5, respectively. Figures 6 and 7 respectively show the IT2 NSFLS-1 (BP) predictions after one epoch and twenty-five epochs of training. Observe that IT2 NSFLS-1 (BP) prediction after epoch one takes better into account the variations of the real data than its counterpart T1 NSFLS-1 (B). The predicted exchange rate using hybrid IT2 NSFLS-1 (RLS-BP) after one epoch and twenty-five epochs of training, are depicted in Figures 8 and 9, respectively. The predicted value is in all figures one step ahead in order to compare with its real value. After only one epoch of training, the hybrid IT2 NSFLS-1 (RLS-BP) predictions are very close to the real values, meaning that it is feasible its application to predict and to control critical systems where there is only one chance of training. The reason is that the uncertainties presented in the training data are almost totally incorporated into the 81 rules of the IT2 NSFLS-1 forecaster, and that the fast convergence of the predicted error is improved by the RLS method.

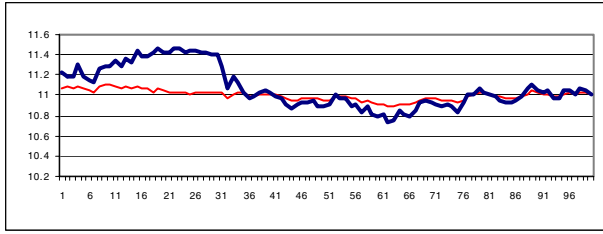


Fig. 4. (Blue) Real historical and (Red) predicted prices of MXNUSD exchange rate. Predictions of the T1 NSFLS (BP) forecaster after one epoch of training.

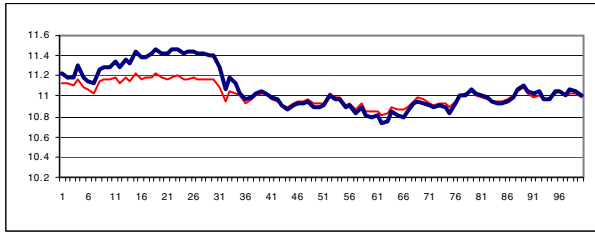


Fig. 5. (Blue) Real historical and (red) predicted prices of MXNUSD exchange rate. Predictions after twenty-five epochs of training of T1 NSFLS (BP).

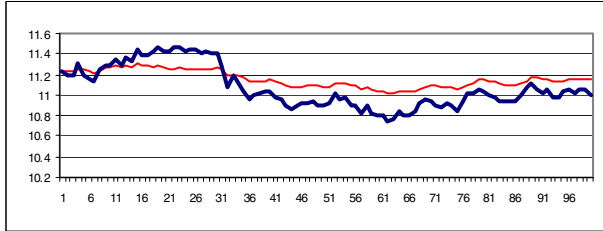


Fig. 6. (Blue) Real historical and (red) predicted prices of MXNUSD exchange rate. Predictions after one epoch of training of IT2 NSFLS-1 (BP).

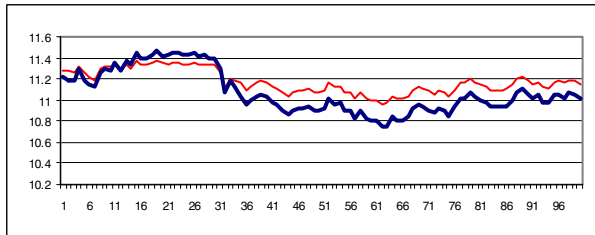


Fig. 7. (Blue) Real historical and (red) predicted prices of MXNUSD exchange rate. Predictions after twenty-five epochs of training of IT2 SFLS-1 (BP).

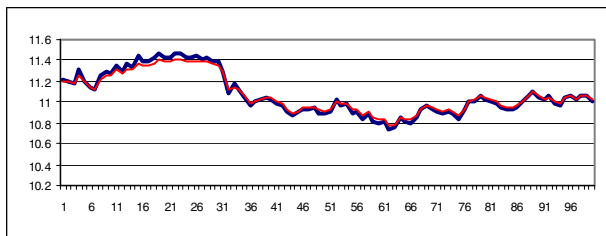


Fig. 8. (Blue) Historical and (red) predicted prices of MXNUSD exchange rate. Predictions after one epoch of training of hybrid IT2 NSFLS-1 (RLS-BP).

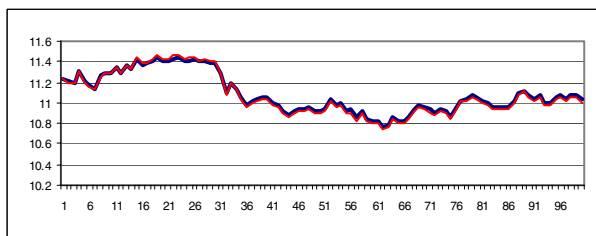


Fig. 9. (Blue) Historical and (red) predicted prices of MXNUSD exchange rate. Predictions after twenty-five epochs of training of hybrid IT2 NSFLS-1 (RLS-BP).

4 Conclusions

An IT2 NSFLS-1 using the hybrid learning method RLS-BP was tested and compared for forecasting the daily exchange rate between Mexican Peso and U.S. Dollar (MXNUSD). The results showed that the hybrid IT2 NSFLS-1 (RLS-BP) forecaster provided the best performance, and the base line T1 NSFLS-1 (BP) provided the worst performance. We conclude, therefore, that it is possible to directly use the daily data of exchange rate to train an IT2 NSFLS-1, in order to model and predict MXNUSD exchange rate one day in advance. It was observed that IT2 NSFLS-1 forecasters efficiently managed the raw historical data.

References

1. Mendel, J.M.: Uncertain rule-based fuzzy Logic systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
2. Mendez, G., Cavazos, A., Leduc, L., Soto, R.: Hot strip temperature prediction using hybrid learning for interval singleton type-2 FLS. In: Proceedings of the IASTED International Conference on Modelling and Simulation, Palm Springs, CA, pp. 380–385 (2003)
3. Mendez, G.: Orthogonal-back propagation hybrid learning algorithm for type-2 fuzzy logic systems. In: IEEE Proceedings of the NAFIPS 2004 International Conference on Fuzzy Sets, Banff Alberta, Canada, vol. 2, pp. 899–902 (2004)
4. Hagrais: A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Transactions on Fuzzy Systems 12(4), 524–539 (2004)

5. Castillo, O., Melin, P.: A new hybrid approach for plant monitoring and diagnostics using type-2 fuzzy logic and fractal theory. In: Proceedings of the International Conference FUZZ 2003, pp. 102–107 (2003)
6. Castillo, O., Huesca, G., Valdez, F.: Evolutionary computing for optimizing type-2 fuzzy systems in intelligent control of non-linear dynamic plants. In: IEEE Proceedings of the NAFIPS 2005 International Conference, pp. 247–251 (2005)
7. Anastasakis, L., Mort, N.: Prediction of the GSPUSD exchange rate using statistical and neural network models. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Benalmadena, España, pp. 493–498 (2003)
8. Lendasse, A., de Boot, E., Wertz, V., Verleysen, M.: Non-linear financial time series forecasting – application to the bel 20 stock market index. *European Journal of Economics and Social Systems* 14, 81–91 (2000)
9. Ito, Y.: Neuro-fuzzy phillips-type stabilization policy for stock process and exchange-rates models. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Benalmadena, España, pp. 499–504 (2003)
10. Saad, E.W., Prokhorov, D.V., Wunch II., D.C.: Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEE Trans. On Neural Networks* 9, 1456–1470 (1998)
11. Leigh, W., Paz, M., Purvis, R.: An analysis of a hybrid neural network and pattern recognition technique for short term increases in the NYSE composite index. *The International Journal of Management Science* 30, 169–176 (2002)
12. Liang, Q., Mendel, J.M.: Interval type-2 fuzzy logic systems: theory and design. *Trans. on Fuzzy Systems* 8, 535–550 (2000)
13. John, R.I.: Embedded interval valued type-2 fuzzy sets. In: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, Honolulu Hawaii, vol. 1 & 2, pp. 1316–1321 (2002)
14. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems* 10, 117–127 (2002)

Minimizing Energy Consumption in Heating Systems under Uncertainty

José Ramón Villar^{1,*}, Enrique de la Cal¹, and Javier Sedano²

¹ University of Oviedo, Computer Science department, Edificio Departamental 1,
Campus de Viesques s/n, 69121 Gijón, Spain
{villarjose, delacal}@uniovi.es

² University of Burgos, Electromechanic department,
Escuela Politécnica Superior, Campus Vena, B2
09006 Burgos, Spain
jsedano@uniovi.es

Abstract. Energy saving systems are needed to reduce the energy taxes, so the electric energy remains balanced. In Spain, a local company that produces electric heaters needs an energy saving device to be integrated with the heaters. The building regulations in Spain introduce five different climate zones, so such device must meet all of them. In previous works the uncertainty in the process was shown, and the different configurations that must be included to accomplish the Spanish regulations were established. It was proven that a hybrid artificial intelligent systems (HAIS) could afford the energy saving reasonably, even though some improvements must be introduced. This work proposes a modified solution to relax the hardware restrictions and to solve the lack of distribution observed. The modified energy saving HAIS is detailed and compared with that obtained in previous works.

Keywords: Fuzzy systems, Hybrid Artificial Intelligence Systems, Real World Applications, Electric Energy saving.

1 Introduction

As stated in [2,12], only Greece, Portugal and Spain have energy taxes deficit because the price of the energy in the energy market is partially liberalized. In Spain, the deficit is growing since year 2000 –the first year the deficit was accepted; and it is estimated to reach the amount of 15000 millions of euros in 2008.

The deficit is caused by the difference between the electric generation costs and the electric energy price to consumers. With the raise of both the petroleum price and the costs of the CO₂ emissions, the problem is unresolved. It is clear that the electric taxes for small consumers will increase in the next years. Consumers must be aware of energy saving methods and devices, in particular, energy saving methods in the heating systems –specially if they are based on electric energy–, which represents the main contribution to the energy consumption.

* Corresponding author.

In Spain, a local company has developed a new dry electric heaters production line. Just because of previous reasons, a device for electric energy saving is to be included in the catalogue. In previous works, the development of such a device has been analyzed, and a multi agent hybrid fuzzy system has been proposed [7,8,9]. The solution integrates the electric heaters and a Central Control Unit (CCU). The CCU is the responsible for distributing the available electric energy to the heaters based in the energy balance concept and with the objective of keeping the user defined comfort level in the house. Although the proposal was proven to make a suitable distribution of the available electric energy, it suffers some deficiencies. Specifically, some bias was found in the steady state. Also, there was a lack of stability in the electric energy output due to fluctuations that must be faced.

In this work, a new proposal to improve the energy saving system is detailed. In this approach, the CCU is fully integrated with the heaters, which allow that the energy distribution algorithm (EDA) could be simplified. The EDA considers the energy balance and the comfort variables to propose the electric power to be spent in heating in the next period. The EDA makes use of a fuzzy controller to obtain a suitable energy rate for each room. The energy requirement for all the rooms is now obtained directly from the power controller in each of the heaters installed in the room. The pre-process stage is now modified to include the dynamics of each room. Finally, a multi objective algorithm is proposed to train the fuzzy controller in the EDA.

This work is organized as follows. The next section deals with the Spanish regulations and the problem definition. Section 3 details the energy saving system proposed in this work. An analysis of the behaviour of this energy saving system in a specific configuration is carried out in Section 4, and a comparison with previous works results is preformed. Finally, conclusions and future work are presented in Section 5.

2 The Spanish Regulations and the Problem Description

In the first quarter of the year 2007, the Spanish parliament approved a new building regulation [10] (RITE). This new regulation had many consequences, as it determined how the new buildings had to be accomplished: materials, isolation, energy efficiency, etc. Up to five different winter climate zones were defined for Spain –from A to E–, each zone represents specific weather conditions that make the climate more or less severe. Therefore, today, not only the builders are affected but also all the products related to energy consumption, as they are to be installed in those new buildings.

In this case, a local enterprise released a new catalogue of electric heaters, which accomplish Spanish regulations. In Spain, the program LIDER [11] validates the heating power installed in a new building. On the other hand, the electric energy provider establishes the energy consumption contract limits. A problem arises when the total power installed in a new building surpasses the contracted energy consumption limit. A common sense solution is to include energy distributors to limit the energy consumption. This kind of devices do not consider the comfort in the house, only the energy consumption, so the local company looks for a new device that considers both the energy consumption and the comfort in the house.

The design of the new device must consider the new building specifications in order to establish the comfort prerequisites and the items related to energy efficiency.

The objective aimed in the energy efficiency is to limit the energy consumption while keeping the thermal comfort in the houses. A study of the thermal system that a house represents is needed to determine the thermal energy requisites, which will depend on the local weather, the season of the year, the isolation, the use of the building and the air purity.

The local company, which produces electric heaters, wants to produce an energy saving device to be included in its catalogue. The electric heaters and the energy saving device are to be installed in houses. The range of the power rates for the heaters are 500, 1000, 1200 and 1500 Watts. The houses must agree with the Spanish regulations enumerated above, and in Figure 1 an example of a Condo house with a possible indoor partition is shown. The main goal is the design of a system for saving electrical energy while keeping the comfort level of the house considering the current electric consumption (excluding the heating installation) and the contracted electrical power. The comfort level in a house is defined as the ambient variables that the user fixes for each room, in the case of this project, only the temperature in each room is considered. The market area of the product is constrained to Spain, so the product must agree with the Spanish regulations.

There will be a CCU responsible of the energy saving algorithm. Each room in the house will have at least one heater, with a given power rate. It is assumed that each heater will collaborate with the CCU in order to reduce the consumed energy while keeping the comfort in the house. The comfort specifications for each room must be fixed, setting the temperature set point for every hour and day of the week. Moreover, the electrical energy contract for each house limits the amount of energy that could drain in a house.

Besides, the sum of the electrical power of each heater plus the power of the rest of the electrical devices in the house must not exceed the contracted power limit. Finally, the whole saving energy system must be economically viable so that consumers can afford it. Also, the system must be robust so failures in the CCU do not collapse the system.

3 An HAIS Solution to Energy Saving

The objective of this energy saving system is to reduce the energy consumption while keeping the comfort level. Comfort is defined using the temperature in each room. The closer the temperature is to the room temperature set point, the higher the comfort is.

Different to the preliminary works, in which the heaters and the CCU work all together to reach the objective, in this work the CCU and the heaters are fully integrated. This means that all the data from each heater can be used, specifically, the room temperature, the heater power rate and the heater power output –from its internal PID–. The energy consumption measurement – whose sensor is wired to the CCU–, should be known so the energy supplier contracted power limit could not be exceeded.

The same design reasoning presented in preliminary works applied in this work [7,8,9]. Interested readers would find in those papers an in-depth description of the Zigbee microcontroller network based Multi Agent System (MAS) approach used. The main difference is that, as long as the heaters and CCU are fully integrated and

the power output of all the heater controllers is accessible, there is no need for estimating the power requirement. In the following Figure 1 the energy saving system (ESS) is outlined. There are two stages: the design stage and the run stage. In the design stage a wide range fuzzy controller is trained and validated, while in the run stage the whole algorithm is executed in the embedded hardware. A detailed description of all the parts of each stage follows.

It is called a *configuration* the pair of values <climate zone, building topology>. As stated before, the Spanish regulations define 5 climate zones, from the less severe – A– to the most severe –E–. On the other hand, the ESS must deal with all kind of building: a detached house, a condominium house, a company headquarters, etc. In each building type there is a wide variety in the number of rooms, the orientation, etc. To deal with such different spread of diversity, the EES is designed with a common part –the energy distribution algorithm (EDA)– and a specific part for each room –a fuzzy controller FC–. The EDA concerns with arranging the available power between the heaters and makes use of the fuzzy controllers to propose a heating power rate for each heater; the calculations are based in the energy deficit in the room. The FC is a mamdani FC with 3 membership functions per variable, and a complete set of rules, designed using an Input/Output mapping.

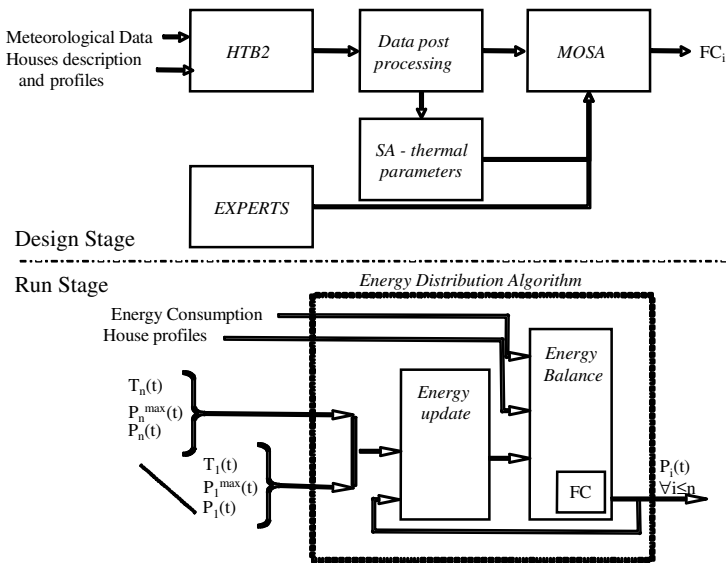


Fig. 1. The ESS two stage procedure. A fuzzy controller is trained for each configuration pair of climate zone and building topology. The run stage represents the energy saving EDA algorithm.

The Design stage deals with the generation of a FC for each configuration. This stage includes several steps: the HTB2 simulation, the data post-processing and the generation of the best suite FC. All of the heaters will have the same initial FC, but the ESS includes an auto-tuning facility to fit the FC to its room requirements. The HTB2 simulation software [5], in following HTB2, is a well known tool suitable to analyze the dynamics of heating systems and the energy consumption in buildings

using the concentrated parameter model [1,4]. The HTB2 is a totally parameterized simulation tool, its main output is the heating power requirements and temperature for each room in the modeled house, both logged at each time interval.

For generating the dataset, all the topologies of the houses and the geographical zones to be covered must be defined, which are market decisions. Then, each building topology and geographical zone must be defined for the HTB2, and a set of simulations for each season must be carried out using realistic profiles of occupancy, set point temperature, small power devices consume, etc. Also a simulation for determining the dynamics of each room in the building is needed. This simulation is the *step response of the house*, where a change in the temperature set point while the rest of parameters are kept constant is analyzed.

The post-processing stage should select two subsets, each one containing ten significant parts from the HTB2 output datasets for a certain configuration: this will represent the training and validation datasets to be used in the third step. Finally, with the step response of the house the thermal resistance and the thermal capacitance for each room are determined [3]. Using the simulated annealing heuristic the best pair of thermal parameters for each room are found.

Finally, the MOSA step is the 10 k fold cross validation multi-objective simulated annealing for training the fuzzy controller for each configuration. The MOSA is the multi-objective adaptation of the simulated annealing presented in [6]. Each individual is the consequent part of the fuzzy controller, and the fitness function is the EDA algorithm. The two objectives to reach are minimizing the energy consumption and keep the house in comfort by means of the EDA. The initial individual has been obtained from expert knowledge, and the antecedents and rules are kept the same for all individuals. At the end of the Design stage a base fuzzy controller for each configuration is obtained. The fuzzy controller will be used in the energy distribution algorithm as follows.

The Run stage is the energy distribution algorithm, named EDA. This algorithm is to be executed in a microcontroller unit, so its complexity should be kept low to avoid getting the CCU collapsed. The EDA request all the heaters for the room temperature, the power rate and the power output they would accomplished if they would were stand alone. The latter is called the *required power*, and the integral of the required power is the *required energy*. The EDA will propose to all the heaters a *heating power*; its integration represents the *heating energy*. The integrals are computed in using a 20 periods window. The *energy deficit* is calculated as the difference between the required energy and the heating energy. Using the room temperature error –the temperature set point minus the room temperature– and the energy deficit the FC proposed a heating power for each heater. The heating power for all the heaters is then fitted to do not surpass the available power neither the heaters power rate. The available power is the power contract limit minus the small power devices consumption –that is, the power that is available for heating purposes–. Also a minimum heating power is assigned to all the heaters with energy deficit to keep them warm.

4 Experiments and Commented Results

This work concerns the evaluation of the best approach to energy saving and deals with the drawbacks found in preliminary works. The experimentation will determine,

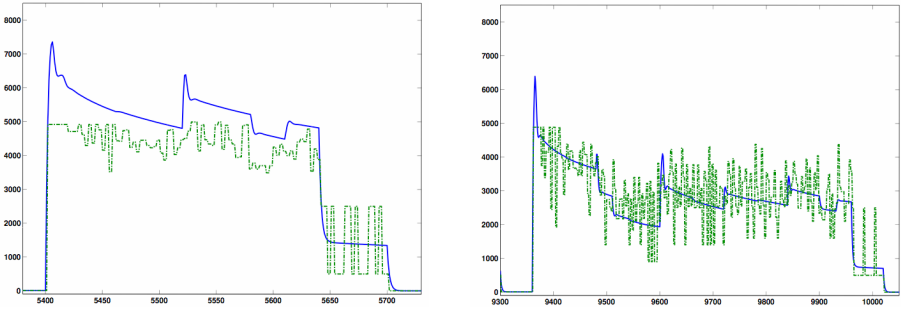


Fig. 2. Aggregated heating power evolution for the house

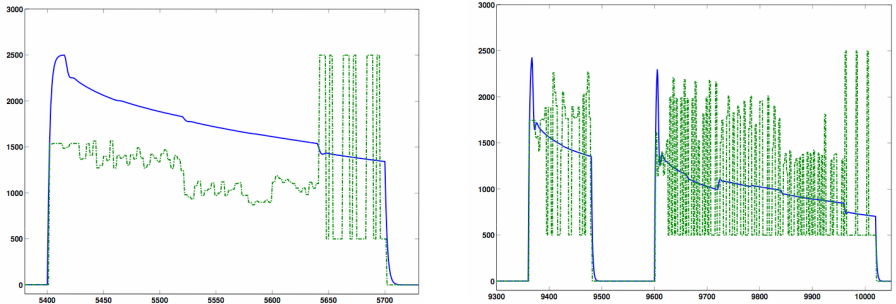


Fig. 3. Heating power evolution for a room in the house

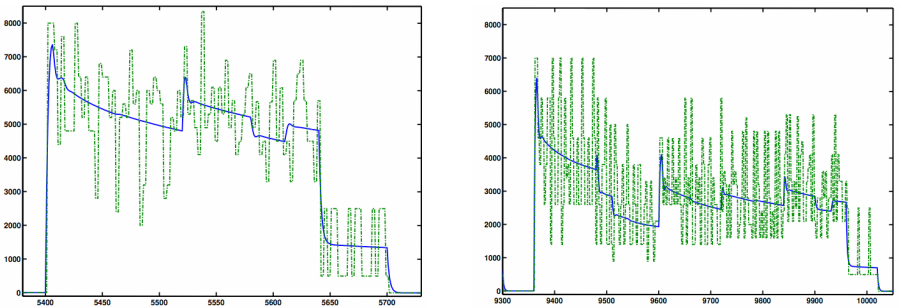


Fig. 4. Aggregated heating power evolution for the house when no Contracted Power Limits is used

if the current approach does fulfill better the two objectives than the preliminary approaches: to minimize the energy consumption and to keep the comfort level in the house. The ESS market area is Spain, thus it must accomplish with the Spanish regulations. Before analyzing the ESS with the whole set of different configurations the system should be valid for only one of the possible situations for a given configuration. The climate zone chosen is the city of Lugo, a D1 city, and the building topology is the condominium topology, in particular, a three bedrooms condo house, with

orientation North-South. The original FC will be used: the MOSA step is to be implemented once this approach gets validated.

The set of configuration files for the HTB2 simulations can be designed once the configuration and the specific situation are chosen. This configuration files must represent real and realistic data. The environmental data such as the outdoor temperature or the solar radiation have been gathered from statistical data for the city of Lugo in three different seasons of year 2007: mid of autumn, mid of winter and mid of spring. Other files, as occupancy file, temperature set point profile for each room in the house, small power devices profile, etc. have been designed attending to realistic profiles. Finally, files like ventilation file have been generated attending to the regulations.

The HTB2 simulation produces two datasets: the step response of the house and the evolution of the heating power required in the house to keep the comfort level. The former is used to obtain the thermal dynamics of the house, the latter for evaluating the EDA. The last step in the Design stage is not used in this work as long as it is desired to compare results with preliminary works, which make use of the FC generated by the experts for calculating the heating power. The thermal dynamics of the house is used to calculate the evolution of the temperature in each room with the heating power proposed by the CCU.

The simulation of the EDA has been carried out with the HTB2 datasets and the thermal dynamics of the house. The results are shown in Figure 2 to Figure 5, solid line is the required power and the dash and dotted line is the outcome of the EDA; left side corresponds with the house being cold, right side corresponds with a warm state of the house. As can be seen in Figures 2 to Figure 4, the lack of stability in the heating power output has been eliminated, not only for the aggregated heating power but also for the heating power corresponding to each room. Also in Figure 3 and Figure 4 it is shown that the bias in the steady state has been fixed. Finally, in Figure 5 the ability of the FC to calculate a heating power based in the error temperature and in the energy deficit under the variability of situations –the different rooms and their geometrics and uses– has been proven, and must be tested with all of the possible situations of all the configurations.

5 Conclusions and Future Work

This work applies different AI techniques in energy saving. The use of a specific software to train and validate the FC, the simulate annealing to obtain the thermal parameters that define the dynamic of each room, the multi-objective simulate annealing, a fuzzy controller in the EDA and the MAS approach. The results have proven the ESS to be valid if only a situation is faced. The ESS has reached all the challenges that preliminary works has left, and the fluctuation heating power and the errors in the steady state has been fixed while the comfort in the house is kept.

Future work includes the training and validating of the FC for each configuration and the validation of the ESS in all the configurations. Finally, the implementation of the MAS and the ESS in the embedded hardware accomplishing the time and computing complexities needed to make a competitive product.

Acknowledgements

This research work is been granted by *Gonzalez Soriano, S.A.* –by means of the the CN-08-028-IE07-60 FICYT research project– and by Spanish M. of Education, under the grant TIN2005-08386-C05-05.

References

- [1] Bojic, M., Despotovic, M., Malesevic, J., Sokovic, D.: Evaluation of the impact of internal partitions on energy conservation for residential buildings in Serbia. *Building and Environment* 42, 1644–1653 (2007)
- [2] CNE, Comisión Nacional de Energía, <http://www.cne.es/cne/Home>
- [3] Davidsson, P., Boman, M.: Distributed monitoring and control of office buildings by embedded agents. *Information Sciences* 171, 293–307 (2005)
- [4] Koroneos, C., Kottas, G.: Energy consumption modeling analysis and environmental impact assessment of model house in thessaloniki (Greece). *Building and Environment* 42, 122–138 (2007)
- [5] Lewis, P., Alexander, D.K.: HTB2: A flexible model for dynamic building simulation. *Building and Environment* 1, 7–16 (1990)
- [6] Sánchez, L., Villar, J.R.: Obtaining transparent models of chaotic systems with multiobjective simulated annealing algorithms. *Information Sciences* 178(4), 952–970 (2008)
- [7] Villar, J.R., de la Cal, E., Sedano, J.: Energy saving by means of fuzzy systems. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) *IDEAL 2007*. LNCS, vol. 4881, pp. 155–167. Springer, Heidelberg (2007)
- [8] Villar, J.R., de la Cal, E., Sedano, J.: Energy Savings by means of Multi Agent Systems and Fuzzy Systems. In: *International Workshop on Practical Applications of Agents and Multi-agent Systems* (2007)
- [9] Villar, J.R., de la Cal, E., Sedano, J.: A Fuzzy Logic efficient Energy Saving system. *Integrated Computer Aided Engineering* (submitted, 2008)
- [10] de la Presidencia, M., B.O.E.: num 207, Real Decreto 1027/2007, de 20 de julio, por el que se aprueba el Reglamento de Instalaciones Térmicas en edificios, Agosto (2007), <http://www.boe.es>
- [11] de Vivienda, M.: Dirección General de Arquitectura y Política de Vivienda, Documento Básico de Ahorro de Energía. Limitación de la demanda de Energía (2005)
- [12] UNESA, Asociación Española de la Industria Eléctrica, <http://www.unesa.es>

Learning to Trade with Incremental Support Vector Regression Experts

Giovanni Montana and Francesco Parrella

Department of Mathematics
Imperial College London

Abstract. Support vector regression (SVR) is an established non-linear regression technique that has been applied successfully to a variety of predictive problems arising in computational finance, such as forecasting asset returns and volatilities. In real-time applications with streaming data two major issues that need particular care are the inefficiency of batch-mode learning, and the arduous task of training the learning machine in presence of non-stationary behavior. We tackle these issues in the context of algorithmic trading, where sequential decisions need to be made quickly as new data points arrive, and where the data generating process may change continuously with time. We propose a master algorithm that evolves a pool of on-line SVR experts and learns to trade by dynamically weighting the experts' opinions. We report on risk-adjusted returns generated by the hybrid algorithm for two large exchange-traded funds, the iShare S&P 500 and Dow Jones EuroStoxx 50.

Keywords: Incremental support vector regression, subspace tracking, ensemble learning, computational finance, algorithmic trading.

1 Introduction

Support vector regressions (SVRs) are state-of-art learning machines for solving linear and non-linear regression problems. Over the last few years, SVRs have been successfully applied to a number of different tasks in computational finance, most notably the prediction of future price movements and asset returns. A variety of SVR architectures have been proposed with applications to different asset classes including equities [19,11] and foreign exchange rates [3,12]. In some cases, hybrid approaches combine standard econometric forecasting models with more flexible SVR algorithms that are able to capture potential non-linear patterns in historical data [10]. Other than expected returns, some initial work has also been done towards forecasting the volatility of an asset [6,9] as well as the direction of its change [11].

A major difficulty in dealing with financial time series representing asset prices is their non-stationary behavior (or concept drift), i.e. the fact that the underlying data generating mechanism keeps changing over time. Some adaptive SVR algorithms have been proposed [23], where more recent errors are penalized more heavily by appropriately modifying the ϵ -insensitive loss function. However, these

approaches are based on algorithms that learn from the data in *batch-mode*, so that when new data points are made available, the learning machine has to be retrained again. Batch learning can be highly inefficient and can introduce costly delays in the decision making process, particularly so in time-aware applications such as algorithmic trading where the data streams used as inputs by the learning machine may be imported at very high sampling frequencies (e.g. every minute or less). In real-time forecasting and trading applications one is often interested in *on-line learning*, a situation where the regression function is updated following the arrival of each new sample. Various approaches for incremental learning with support vector machines have been proposed in the literature, for both classification [4,13,27] and regression problems [15,26,16].

In this paper we propose a hybrid learning algorithm in the context of algorithmic equity trading. The algorithm decides how many shares of a given security to buy or *short sell* at any given point in time. Short selling is the practice of borrowing a security for an immediate sale, with the understanding that it will be bought back later. By using short selling, the trading machine can benefit from decreases, not just increases, in the prices, and could achieve financial returns that are only mildly correlated with the market returns. This class of strategies is often referred to as *market neutral* or *relative value* [20]. At the core of our approach there is an on-line SVR algorithm that learns the relationship between the observed security price and the underlying market conditions, and then suggests in real-time whether the security being traded is currently underpriced or overpriced, relative to the market. The market conditions are represented by *patterns*, in the form of latent factors, that are dynamically extracted from a (possibly very large) collection of cross-sectionally observed financial data streams. Section 2 contains a concise description of our on-line SVR algorithm.

In practice, the assumed relationship between the marker factors and the target security constantly evolves over the time, and failing to capture these changes can crucially affect the performance of the trading algorithm. We propose an ensemble learning approach. Our solution consists of evolving an ensemble of SVR *experts*, where each expert is characterized by a specific set of hyperparameters. A master algorithm, detailed in Section 3, is then responsible for weighting these expert opinions and taking a final trading decision. In Section 4, the meta algorithm is empirically shown to outperform the best SVR expert in the ensemble in terms of risk-adjusted returns.

2 Asset Pricing with Incremental SVR

In this section we describe how SVR is applied to price a security using only market data, in an incremental fashion. Suppose that, on each trading day i , the security price is denoted by y_i , and we have made t observations, respectively. We envisage a situation where n cross-sectional financial data streams representative of the target market are observed and collected, at each discrete time, in a vector s_i . Since n may be very large, we project sequentially the data onto the

directions of maximal variability in order to extract an input vector or pattern x_i of dimension $k < n$ (see also Section 3). With the most recently arrived input vector x_t at hand, we then wish to obtain an updated estimate of y_t , say \hat{y}_t . In our approach, this estimate is obtained as $f(x_t) = w\phi(x_t) + b$, where w is a weight vector, $\phi(\cdot)$ is a non-linear mapping from input space to some high-dimensional feature space, and b is a scalar representing the bias.

In order to perform the estimation task above, we adopt the SVR framework based on the ϵ -insensitive loss function [24]. Accordingly, we are willing to accept that price estimates that are within $\pm\epsilon$ of the observed price y_t are always considered *fair* prices, for a given positive scalar ϵ . Estimation errors less than ϵ are ignored, whereas errors larger than ϵ result in additional linear losses [7]. Introducing slack variables ξ_i, ξ_i^* quantifying estimation errors greater than ϵ , the price estimation task can be formulated as a constrained optimization problem: we search for w and b that minimize the regularized loss

$$\frac{1}{2} w^T w + C \sum_{i=1}^t (\xi_i + \xi_i^*) \tag{1}$$

subject to constraints $-y_i + (w^T \phi(x_i) + b) + \epsilon + \xi_i \geq 0$, $y_i - (w^T \phi(x_i) + b) + \epsilon + \xi_i^* \geq 0$ and $\xi_i, \xi_i^* \geq 0$ for all $i = 1, \dots, t$, where a penalization on the norm of the weights has been imposed and is controlled by the parameter C . As is done commonly, we transform this primal optimization problem to its dual quadratic optimization problem. The Representer Theorem [25] assures us that the solution to the resulting optimization problem may be expressed as

$$f(x) = \sum_{i=1}^t \theta_i K(x_i, x) \tag{2}$$

where θ_i is defined in terms of the dual variables and $K(\cdot, \cdot)$ is a kernel function [22]. In our application, we adopt a Gaussian kernel depending upon the specification of a hyperparameter σ . Our objective is to minimize (1) *incrementally*, without having to retrain the model entirely at each point in time.

Following [4,15], an incremental version of SVR can be obtained by first classifying all training points into three distinct auxiliary sets according to the KKT conditions that define the current optimal solution. Specifically, after defining the margin function $h_i(x_i)$ as the difference $f(x_i) - y_i$ for all time points $i = 1, \dots, t$, the KKT conditions can be expressed in terms of $\theta_i, h_i(x_i), \epsilon$ and C . Accordingly, each data point falls into one of the following sets,

$$\begin{aligned} \mathcal{S} &= \{i \mid (\theta_i \in [0, +C] \wedge h_i(x_i) = -\epsilon) \vee (\theta_i \in [-C, 0] \wedge h_i(x_i) = +\epsilon)\} \\ \mathcal{E} &= \{i \mid (\theta_i = -C \wedge h_i(x_i) \geq +\epsilon) \vee (\theta_i = +C \wedge h_i(x_i) \leq -\epsilon)\} \\ \mathcal{R} &= \{i \mid \theta_i = 0 \wedge |h_i(x_i)| \leq \epsilon\} \end{aligned} \tag{3}$$

Set \mathcal{S} contains the support vectors, i.e. the data points contributing to the current solution, whereas the points that do not contribute are contained in \mathcal{R} . All other points are in \mathcal{E} . When a new data point arrives, the incremental

algorithm relabels all points in an iterative way until optimality is reached again. Similarly, old data points can be discarded easily without having to retrain the machine. Our own implementation in C++ mostly differs from [15] in the definition of the auxiliary sets (3). Full details can be found in [21].

3 Learning to Trade with Expert Advice

Let us suppose that we were able to select the hyperparameter triple (ϵ, C, σ) yielding a regression function $f(x_t)$ of form (2) that truly describes the relationship between the current market patterns x_t and the fair price of the target security. Based on historical data up to date, this regression model would produce an estimate \hat{y}_t of the fair price, and a corresponding estimate $\hat{m}_t = y_t - \hat{y}_t$ of the discrepancy between observed and estimated price. We can now make the following observations. If the relationship above was not affected by noise, and if markets were always perfectly efficient, we would have that $\hat{m}_t = 0$ at all times. On the other hand, when $|\hat{m}_t| > 0$, an *arbitrage* opportunity arises. For instance, a negative \hat{m}_t would indicate that the security is under-valued, and a sensible decision would be to buy shares of the security, which is compatibly with the expectation that markets will promptly react to this temporary inefficiency by moving the price up. Analogously, a positive \hat{m}_t would suggest to short sell a number of shares in accordance with the expectation that the price will be decreasing over the following time interval. Further insights and explanations regarding this *relative value* trading strategy can be found in [3,21,8].

In a realistic setting, tuning the hyperparameter triple is an arduous task. Moreover, an optimal choice may soon become sub-optimal, due to the highly dynamic nature of the data under consideration. For this reason, we have chosen to pursue a hybrid approach consisting of a collection of SVR experts. On each trading day, a master algorithm adopting the weighted majority voting principle [14] collects the opinion of E experts, each one suggesting whether to buy or short sell the security, and takes a final decision. In summary, the following operations are performed at any given time t , after observing the most recent data pair (s_t, y_t) :

Pattern extraction: A pattern x_t is extracted from the data stream vector s_t by projecting s_t onto the eigenvector corresponding to the largest eigenvalue. An estimate of the eigenvector at time t , denoted \hat{h}_t , is given by a recursive updating equation,

$$\hat{h}_t = \frac{t-1}{t} \hat{h}_{t-1} + \frac{1}{t} s_t s_t^T \frac{\hat{h}_{t-1}}{\|\hat{h}_{t-1}\|}$$

Note how this solution does not require the explicit computation of the covariance matrix (see [28] for derivations).

Expert opinion: Each expert in the pool produces a mispricing estimate

$$\hat{m}_t^{(e)} = y_t - \hat{y}_t^{(e)} \quad e = 1, \dots, E \tag{4}$$

where the estimated fair price $\hat{y}_t^{(e)}$ is given by the SVR expert as in Eq. (2). Each expert is defined by a specific parameter value in a grid of points: ϵ varies

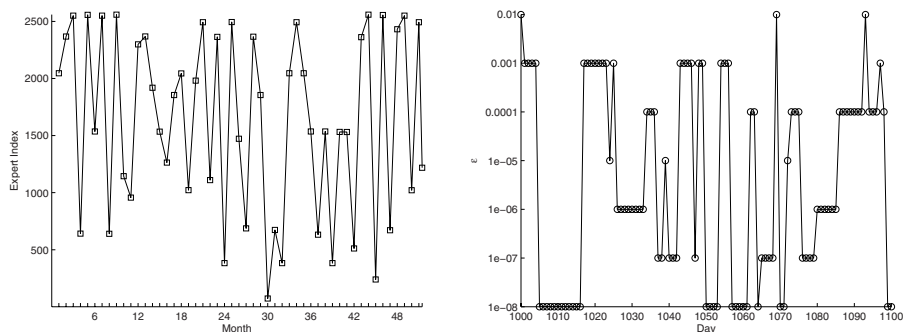


Fig. 1. DJ EuroStoxx 50: time-dependence of the best expert (left) and time-dependence of the ϵ hyperparameter corresponding to the best expert for a few selected trading days (right). The criterion chosen to select the best expert is the Sharpe Ratio, a measure of risk-adjusted returns.

between 10^{-1} and 10^{-8} , while both C and σ vary between 0.0001 and 1000. In total there are 2560 experts. The trading signal generated by each expert is given by $d_t^{(e)}$: this is set to 1 (buy) if $\hat{m}_t^{(e)} > 0$ and to 0 (short sell) if $\hat{m}_t^{(e)} < 0$.

Ensemble-based decision: The meta algorithm produces a final binary trading decision expressed as $d_t = 1$ (buy) if $q_{t0} < q_{t1}$ and $d_t = 0$ (short sell) otherwise, where

$$q_{t0} = \sum_{e:d_t^{(e)}=0} \omega_t^{(e)} \quad \text{and} \quad q_{t1} = \sum_{e:d_t^{(e)}=1} \omega_t^{(e)}$$

At the beginning of the next time period, $t + 1$, all experts whose decision turned out to be incorrect are penalized by downgrading their weights to $\omega_{t+1}^{(e)} = \beta \omega_t^{(e)}$, for a given user-defined positive β . We will refer to this algorithm as WMV. Its theoretical properties have been extensively studied [5].

4 Experimental Results

We have tested the master algorithm described in Section 3 using data from two major exchange-traded funds (ETFs). An ETF is a portfolio of securities that trades like an individual stock on a major stock exchange, and is constructed to track closely the performance of an index of securities. Exactly like stocks, ETFs can be bought or sold throughout the trading day. The analysis presented here demonstrates that our trading algorithm can detect and exploit arbitrage opportunities. The two ETFs we use are the iShare S&P 500 and the Dow Jones EuroStoxx 50. They were introduced on the market on 19/05/2000 and 21/10/2002, respectively. We collected daily historical prices until 13/09/2007, which resulted in data sets with sample sizes of 1856 and 1279 observations. The

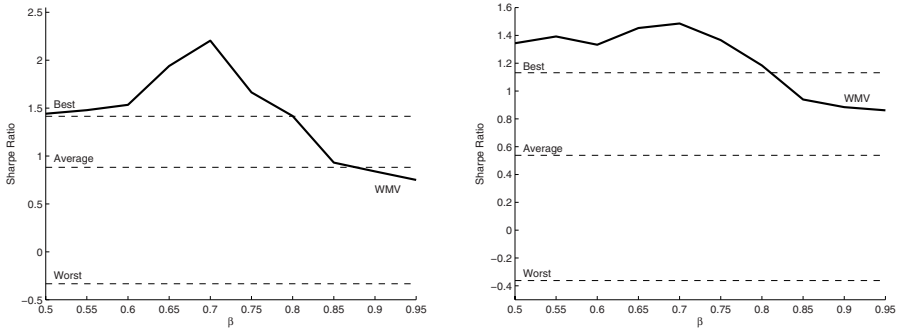


Fig. 2. Sharpe Ratio obtained by the WMV algorithm as a function of β for the two data sets, DJ EuroStoxx 50 (left) and iShare S&P 500 (right), using all the historical data

first 250 data points were used for learning purposes only. A trade is made on each day.

The quality of an expert was evaluated by computing the Sharpe Ratio, a measure of risk-adjusted returns. Figure 1 clearly illustrates how the best performing expert changes over time using a sliding window of 22 days (a trading month). Different window sizes produced very similar patterns. The same figure also illustrates the time-varying nature of the ϵ parameter corresponding to the best expert for a randomly chosen sub-period. It clearly emerges that the overall performance may be improved by dynamically selecting or combining experts. In fact, Figure 2 shows that WMV consistently outperforms the average expert for most values of β . More surprisingly, for a wide range of β values, WMV

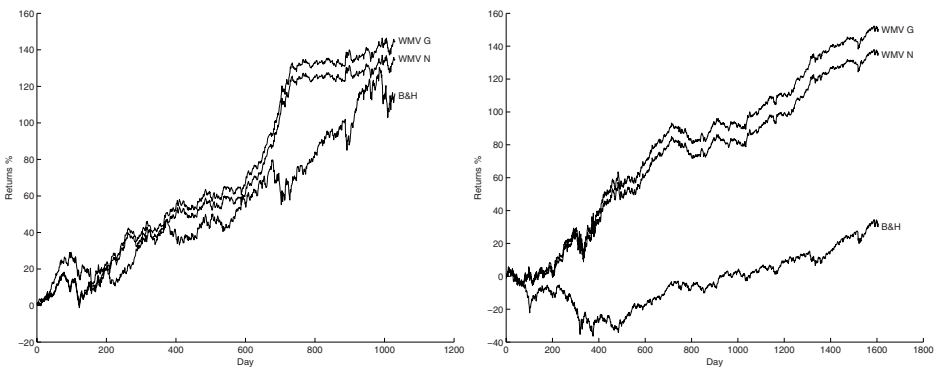


Fig. 3. Gross (G) and net (N) returns generated by the WMV algorithm compared to a standard buy-and-hold (B&H) investment strategy for the two data sets, EuroStoxx 50 (left) and iShare S&P 500 (right), using all the available historical data. B&H is a passive, long-only strategy that consists in buying shares of the ETF at the beginning of the investment period and holding them till the end of the period.

outperforms the best performing expert in the pool by a large margin. Clearly, the meta algorithm is able to strategically combine the expert opinion in a dynamic way. As our ultimate measure of profitability, we compared financial returns generated by WMV with returns generated by a simple *Buy-and-Hold* (B&H) investment strategy, for both data sets, in Figure 3. Net returns are obtained by taking into account fixed transaction costs and slippage (10 USD and 2 bps, respectively). The figure also illustrates the typical market neutral behavior of the trading algorithm: overall, the correlation coefficient between the returns generated by WMV and the market returns is 0.579 for iShare and 0.521 for EuroStoxx.

5 Conclusions

We have suggested a hybrid algorithm that combines the flexibility of SVR in discovering potential price discrepancies with the robustness of the ensemble learning approach. The final WMV algorithm only depends upon one adjustable parameter, β , for which a default value of 0.7 has proved to work well with very different data sets. Further improvements may be obtained by explicitly forecasting the mispricing series (4), as in (17), and by adopting alternative on-line learning schemes (e.g. (29)). Preliminary analyses seem to indicate that the algorithm's performance degrades only marginally when measurable transaction costs, such as bid-ask spreads and commission fees, are taken into account.

References

1. Bao, Y., Liu, Z., Wang, W.: Forecasting stock composite index by fuzzy support vector machine regression. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics (2005)
2. Burgess, A.N.: Applied quantitative methods for trading and investment, chapter Using Cointegration to Hedge and Trade International Equities Finance, pp. 41–69. Wiley, Chichester (2003)
3. Cao, D., Pang, S., Bai, Y.: Forecasting exchange rate using support vector machines. In: Fourth International Conference on Machine Learning and Cybernetics (2005)
4. Cauwenberghs, G., Poggio, T.: Incremental and decremental support vector machine learning, Cambridge, vol. 13, pp. 409–123 (2001)
5. Cesa-Bianchi, N., Lugosi, G.: Prediction, learning, and games. Cambridge University Press, Cambridge (2006)
6. Chang, B.R., Tsai, H.F.: Forecast approach using neural network adaptation to support vector regression grey model and generalized auto-regressive conditional heteroscedasticity. Expert Systems with Applications: An International Journal 34, 925–934 (2008)
7. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
8. Elliott, R.J., van der Hoek, J., Malcolm, W.P.: Pairs trading. Quantitative Finance, 271–276 (2005)

9. Gavrishchaka, V.V., Banerjee, S.: Support vector machine as an efficient framework for stock market volatility forecasting. *Computational Management Science* 3(2), 147–160 (2006)
10. He, Y., Zhu, Y., Duan, D.: Research on hybrid arima and support vector machine model in short term load forecasting. In: *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications* (2006)
11. Huang, W., Nakamori, Y., Wang, S.: Forecasting stock market movement direction with support vector machine. *Computers & Operations Research* (2004)
12. Ince, H., Trafalis, T.B.: A hybrid model for exchange rate prediction. *Decision Support Systems* 42, 1054–1062 (2006)
13. Laskov, P., Gehl, C., Kruger, S.: Incremental support vector learning: analysis, implementation and applications. *Journal of machine learning research* 7, 1909–1936 (2006)
14. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. *Information and Computation* 108, 212–226 (1994)
15. Ma, J., Theiler, J., Perkins, S.: Accurate on-line support vector regression. *Neural Computation* 15 (2003)
16. Martin, M.: On-line support vector machine regression. In: *13th European Conference on Machine Learning* (2002)
17. Montana, G., Triantafyllopoulos, K., Tsagaris, T.: Data stream mining for market-neutral algorithmic trading. In: *Proceedings of the ACM Symposium on Applied Computing*, pp. 966–970 (2008)
18. Montana, G., Triantafyllopoulos, K., Tsagaris, T.: Flexible least squares for temporal data mining and statistical arbitrage. *Expert Systems with Applications* (2008) doi:10.1016/j.eswa.2008.01.062
19. Nalbantov, G., Bauer, R., Sprinkhuizen-Kuyper, I.: Equity style timing using support vector regressions. *Applied Financial Economics* 16, 1095–1111 (2006)
20. Nicholas, J.G.: *Market-Neutral Investing: Long/Short Hedge Fund Strategies*. Bloomberg Professional Library (2000)
21. Parrella, F., Montana, G.: A note on incremental support vector regression. Technical report, Imperial College London (2008)
22. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge (2004)
23. Tay, F., Cao, L.: ϵ -descending support vector machines for financial time series forecasting. *Neural Processing Letters* 15, 179–195 (2002)
24. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, Heidelberg (1995)
25. Wabha, G.: *Spline models for observational data*. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 59. SIAM, Philadelphia (1990)
26. Wang, W.: An incremental learning strategy for support vector regression. *Neural Processing Letters* 21, 175–188 (2005)
27. Wen, Y., Lu, B.: Incremental Learning of Support Vector Machines by Classifier Combining. In: *Advances in Knowledge Discovery and Data Mining*, pp. 904–911. Springer, Heidelberg (2007)
28. Weng, J., Zhang, Y., Hwang, W.S.: Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(8), 1034–1040 (2003)
29. Yaroshinsky, R., El-Yaniv, R., Seiden, S.: How to better use expert advice. *Machine Learning* 55(3), 271–309 (2004)

Craniofacial Superimposition Based on Genetic Algorithms and Fuzzy Location of Cephalometric Landmarks

Oscar Ibáñez¹, Oscar Cordon¹, Sergio Damas¹, and Jose Santamaría²

¹ European Centre for Soft Computing, Asturias, Spain
{oscar.ibanez, oscar.cordon, sergio.damas}@softcomputing.es

² Dpt. Computer Science, University of Jaén. Jaén, Spain
jslopez@ujaen.es

Abstract. Craniofacial superimposition is the second stage of a complex forensic technique that aims to identify a missing person from a photograph (or video shot) and the skull found. This specific task is devoted to find the most appropriate pose of the skull to be projected onto the photograph. The process is guided by a number of landmarks identified both in the skull (craniometric landmarks) and in the photograph (cephalometric landmarks). In this contribution we extend our previous genetic algorithm-based approach to the problem by considering the uncertainty involved in the location of the cephalometric landmarks. This proposal is tested over two real cases solved by the Physical Anthropology lab at the University of Granada (Spain).

Keywords: Craniofacial superimposition, genetic algorithms, fuzzy landmarks, image registration.

1 Introduction

Photographic supra-projection [1] is a forensic process where photographs or video shots of a missing person are compared with the skull that is found. By projecting both photographs on top of each other (or, even better, matching a scanned three-dimensional skull model against the face photo/video shot), the forensic anthropologist can try to establish whether that is the same person.

To do so, an accurate 3D model of the skull is first demanded. Next, the matching of two sets of radiometric points (facial anthropometric landmarks in the subject photograph, and cranial anthropometric landmarks in the obtained skull model) is considered to guide the superimposition of the skull 3D model and the photograph [1]. Then, a decision making stage starts by analyzing the different kinds of achieved matches between landmarks. Some of them will perfectly match, some will partially do so, and finally some others will not. After the whole process, the forensic expert must declare whether the analyzed skull corresponds to the missing person or not.

This procedure is very time consuming and there is no systematic methodology but every expert often applies a particular process. Hence, there is a strong interest in designing automatic methods to support the forensic anthropologist to put it into effect.

In a previous proposal [2], we tackled the second stage of the process, i.e. the craniofacial superimposition, by using a real-coded genetic algorithm (GA), assuming no uncertainty was involved in the process. However, two sources of uncertainty can actually be identified in the craniofacial superimposition problem. On the one hand, the *matching uncertainty* (not modeled in this contribution) will refer to the uncertainty involved in the matching of the landmarks that correspond to the two different objects: the photograph and the skull (as said, there is a clear partial matching situation). On the other hand, the *location uncertainty* is related to the extremely difficult task of locating the landmarks [9] in a invariable place, with the accuracy required for this application. Indeed, every forensic anthropologist is prone to locate the landmarks in a slightly different place. The ambiguity may also arise from reasons like variation in shade distribution depending on light condition during photographing, unsuitable camera focusing, poor image quality, etc.

Obviously, the location uncertainty affects any of the landmarks involved in the craniofacial superimposition process, either if they belong to the photograph or the skull model. For the sake of simplicity, in this contribution we will concentrate our study on the uncertainty related to cephalometric landmarks as a first approach.

This contribution aims to extend our work in [2] with a first approach to tackle the location uncertainty by defining different fuzzy areas where the experts should locate every landmark. The resulting genetic fuzzy system [3] is tested on two superimposition problems of real-world identification cases solved by the Physical Anthropology lab at the University of Granada.

The structure of the contribution is as follows. Section 2 is devoted to review our previous proposal on craniofacial superimposition. Our proposal is described in Section 3. Then, Section 4 presents the experimental study. Finally, Section 5 collects some concluding remarks and future works.

2 Real Coded Genetic Algorithm for Craniofacial Superimposition

In [4], we reviewed the state of the art in craniofacial superimposition, remarking there were few automatic proposals to accomplish this task. Nickerson et al.'s method [5] used a GA to find the optimal parameters of the similarity and perspective transformation that overlays the 3D skull model on the face photograph. Because of the lack of space, we cannot describe those geometric transformations. The interested reader is referred to [2]. However, it didn't achieve acceptable results on our data (either synthetic or real ones). Hence, we proposed a new design for the GA considering Nickerson et al.'s proposal as a base.

The problem we are tackling aims to find the best perspective transformation that, when applied to the skull 3D model, projects it properly onto the 2D photograph. The search for the best transformation is guided by a number of pairings between craniometric and cephalometric landmarks located in the skull and the face, respectively. This transformation is defined by a number of real parameters. Hence, our first improvement with respect to the Nickerson's proposal was based on using a real-coded GA, instead of a binary coding, where the transformation parameters were encoded as

a chromosome comprised by 12 real genes. Different definitions of the objective function were studied and the one achieving the best results was the mean square error:

$$MSE = \frac{\sum_{i=1}^N \|f(C_i) - F_i\|^2}{N} \quad (1)$$

where $\|\cdot\|$ is the 2D Euclidean distance, N is the number of considered landmarks (only four in Nickerson et al.'s proposal), C_i corresponds to every 3D landmark, F_i refers to every photo 2D landmark, and $f(C_i)$ represents the position of the transformed skull 3D landmark C_i in the projection plane.

We also designed more efficient genetic operators. Indeed, we used the tournament selection [6], the blend crossover (BLX- α) [7] and the random mutation operators [8].

3 Fuzzy Location of Landmarks for Craniofacial Superimposition

The use of fuzzy landmarks in this contribution aims to face the *location uncertainty* in the photograph of the missing person. According to this problem, each forensic expert could place each landmark in different positions in the 2D image. Thus, the higher the uncertainty related to a landmark, the broader the region where the forensic experts will locate the landmark.

Based on the work by Sinha [10], we consider the cephalometric landmarks as rectangular zones, instead of using crisp locations. We will refer to these rectangular zones as fuzzy landmarks. Every fuzzy landmark is given by a rectangular array of pixels defined by diagonally opposite corners (x_1, y_1) and (x_2, y_2) (say $x_2 > x_1$ and $y_2 > y_1$). Hence, the bigger the rectangle, the higher the uncertainty associated to the landmark.

Shina [10] introduced these concepts to provide more robustness and toleration to a Neural Network (NN) to match 2D facial images. Ghosh and Sinha [11] adapted the original proposal to tackle 2D craniofacial superimposition. In these works, crisp points were substituted by rectangles, to avoid human error due to image ambiguity. Each fuzzy landmark was then temporarily defuzzied by taking the centroid as a crisp target feature. Using the centroids for a first NN weights adaptation, there was a later stage where the rectangle landmarks were adapted (reduced) by means of the NN responses. Solving the problem of craniofacial superimposition, this work has two important limitations, already pointed out by the authors: a part of the cranial image that would never be properly mapped, and the need of a frontal view photograph. Moreover, this method was not fully applicable because of its long computation time, and the need of separately applying two different networks to the upper skull contour and to the frontal view cranial features, where the overlaying found by the first one could be disrupted by the second.

In contrast to Sinha's work, which used these rectangular zones to train a NN, we will model this uncertainty to guide our GA-based craniofacial superimposition procedure. In our case, this is done with the aim of avoiding local minima by prioritizing some landmarks (more precisely located) rather than others (fuzzily located).

Therefore, we have modified the previous definition of the objective function (see section 2) as follows:

$$Fitness = \frac{\sum_{i=1}^N \sqrt{[u_i(x'_{ci} - x_{fi})]^2 + [v_i(y'_{ci} - y_{fi})]^2}}{N} \tag{2}$$

where x'_{ci} and y'_{ci} are respectively the coordinates of the transformed skull 3D landmark Ci in the projection plane, x_{fi} and y_{fi} are the centroids of the fuzzy landmark of every photo 2D landmark, and N is the number of considered landmarks. The terms u_i , v_i are used to represent the uncertainty around each landmark. Each value depends on the size of the rectangular zone, such that,

$$u_i = \frac{1}{1 + |x_2 - x_1|} \qquad v_i = \frac{1}{1 + |y_2 - y_1|} \tag{3}$$

where (x_1, y_1) and (x_2, y_2) are diagonally opposite corners. In this formulation $(x_2 - x_1)$ and $(y_2 - y_1)$ are, respectively, measures of X and Y axis uncertainty.

According to this formulation, when the rectangle defining the fuzzy landmark is bigger (i.e., lower value of u_i and/or v_i), the corresponding weight in the fitness function will be lower. Thus, the more uncertainty is around a landmark, the less important this landmark is for the optimization process (error minimization).

On the other hand, instead of providing weighting factors to the localization of each component of the Euclidean distance (Eq. 2), we propose to weight each component of the distance (mainly based in experimental results) as follows:

$$Fitness_2 = \frac{\sum_{i=1}^N \sqrt{u_i (x'_{ci} - x_{fi})^2 + v_i (y'_{ci} - y_{fi})^2}}{N} \tag{4}$$

Notice that $Fitness_2$ is an alternative way to model the existing location uncertainty that strengthen it (note that the component square distance is multiplied by u_i/v_i here while by u_i^2/v_i^2 in Eq. 2).

4 Experiments

After explaining the sources of uncertainty and our proposal to deal with the location uncertainty, we will study its performance. Section 4.1 presents the experimental design. Sections 4.2 and 4.3 are devoted to the analysis of two real-world case studies.

4.1 Experimental Design

According to what landmarks are more sensitive to uncertainty and the characteristics of the image (size, resolution, contrast, etc.), we use different sizes of rectangles to differentiate the influence of subsets of fuzzy landmarks. Besides, we simulate the process followed by ten different anthropologists to manually identify the cephalometric landmarks in the two cases of study presented in this contribution. In particular, we assume a uniform probability distribution inside the rectangle defining every fuzzy landmark. Hence, each landmark is simulated by randomly selecting a point inside such rectangle as a crisp landmark. Then, we use these ten sets of landmarks to

solve the corresponding superimposition problems, by means of our previous GA (see Section 2). We compare these results with those reached by the current GA-based craniofacial superimposition based on fuzzy location of cephalometric landmarks.

We base our work on positive and negative identification cases addressed by the staff of the Physical Anthropology lab at the University of Granada in collaboration with the Spanish scientific police. We will consider a positive and a negative case, both solved following a manual approach. We will consider the 2D photographs and skull 3D models acquired at the lab using their Konica-Minolta 3D Lasserscanner VI-910. Experiments consider both variants of the fitness function (Section 3). The best performing GA parameter values [2] are used (Table 1).

Table 1. Parameter settings of the genetic algorithm used for the craniofacial superimposition

GA Parameter settings			
Generations	600	Population size	600
Tournament size	2	BLX- α parameter	0.7
Crossover probability	0.9	Mutation probability	0.2

For all the tests, 30 runs of the GA are considered. Tables 2 and 3 show the minimum (m), maximum (M), mean (μ) and standard deviation (σ) of MSE relative to the Euclidean distances between cephalometric landmarks and the transformed craniometric landmarks using the best solution achieved by the GA. Notice that in the case of crisp landmarks, the fitness function is defined as the MSE. However, when we use fuzzy landmarks the fitness function is not the MSE, but a weighted sum of the distances between the transformed skull landmarks and the centroids of the fuzzy regions. Owing to that, numerical results are not fully comparable. Nevertheless, we have used MSE as the only way of providing an error measurement.

4.2 Positive Case Study

The facial photographs of this positive case were provided by the family. The identity was confirmed using manual photographic supra-projection. We studied this real case with the consent of the relatives. The 2D image used by anthropologists is a 290 x 371 color image. The 3D model of the skull comprises 243.202 points.

The anthropologists identified seven cephalometric landmarks. However, they had to not consider the zygion landmark to be able to achieve a reasonable solution following a manual photographic supra-projection (see [2] for a detailed description of the manual approach). In our approach, we will consider all the seven landmarks the experts identified. The corresponding craniometric ones were manually extracted from the skull 3D model. Due to the pose of the face and the quality and resolution of the photograph, three sizes of fuzzy landmarks have been defined. The four fuzzy landmarks in the vertical line dividing the face in two symmetrical sides (Nasion, Subnasale, Labiale superios, Gnathion) are defined by a 2 x 3 rectangle. The two fuzzy landmarks in the center point of the ocular cavity (Ectocanthions) are given by an 8 x 8 rectangle. Finally, the landmark close to the right ear (Zygion) is defined by a 12x12 rectangle. The closer the landmark to the vertical line the smaller the rectangle defining the corresponding fuzzy landmark.

Table 2 presents the MSE for craniofacial superimpositions, distinguishing crisp and fuzzy location. Notice that the values correspond to 30 runs of the GA. In the case of crisp landmarks, the values correspond to ten different sets. It is important to remark that results are not fully comparable since the superimposition process using fuzzy landmarks does not minimize the MSE but a different function. Figure 1 presents the superimposition results of the crisp and the fuzzy approaches.

Table 2. Case study 1. Superimposition results obtained using fuzzy and crisp locations.

Test	MSE			
	m	M	μ	σ
Fuzzy Landmarks (Eq. 2)	0.020	0.321	0.129	0.101
Fuzzy Landmarks (Eq. 4)	0.020	0.210	0.053	0.046
Crisp Landmarks (10 “forensics”)	0.014	0.135	0.031	0.017



Fig. 1. Positive case study . From left to right. The best and worst results achieved using crisp landmarks. On the right, the best solution considering fuzzy landmarks.

On the one hand, the best and worst results achieved by ten forensic experts¹ are in the left and center of figure 1, respectively. We can observe the high variability between both results because of the different set of landmarks used. On the other hand, the best result of the genetic fuzzy superimposition is presented on the right. These results correspond to the best of the 30 runs of the GA.

4.3 Negative Case Study

The second case is a negative one, that is, the person in the picture does not correspond with the skull used for the superimposition. The skull 3D model (comprising 327.641 points) corresponds to a young woman and the 2D photograph is a 760 x 1050 RGB color image of a similar age woman. As in the previous case, the

¹ As said, we are emulating the behavior of ten different forensic experts when locating the set of cephalometric landmarks (section 5.1).

anthropologists identified seven cephalometric landmarks and the corresponding craniometric ones were manually extracted from the skull 3D model.

Owing to the already mentioned source of uncertainties and provided that, there are not extremely laterally landmarks (as the Zygion landmark seen in the first case study), only two sizes of fuzzy landmarks have been defined. Furthermore, in the definition of their sizes, we have taken into account the highest resolution of the source image. The four landmarks in the aforementioned vertical line are defined by a 5×7 rectangle. The three landmarks in the inner and outer parts of the ocular cavities (Ectocanthions and Endocanthions) are defined by a 20×20 rectangle.

Table 3 presents the MSE for craniofacial superimpositions results, as in Table 2. Since the superimposition process using fuzzy landmarks does not minimize MSE but a different function, these results are not fully comparable. However, MSE from both methods are very similar in this case.

Table 3. Case study 2. Superimposition results obtained using fuzzy and crisp locations.

Test	MSE			
	m	M	μ	σ
Fuzzy Landmarks (Eq.2)	0.0074	0.0682	0.0319	0.0207
Fuzzy Landmarks (Eq.4)	0.0077	0.0577	0.0270	0.0165
10 Forensic anthropologists	0.0067	0.0706	0.0225	0.0144

Figure 2 comprises the superimposition results of the crisp and the fuzzy approaches for this second case. We can observe again the high variability between the best and worst results achieved using crisp landmarks because of the different sets of landmarks used. Anyhow, these results are clearly worse than the superimposition achieved using fuzzy landmarks. The skull pose, shape and size of the fuzzy solution are much more suitable to the photograph characteristics than the crisp ones. Indeed, the superimposition achieved following the fuzzy approach is close to an ideal one.

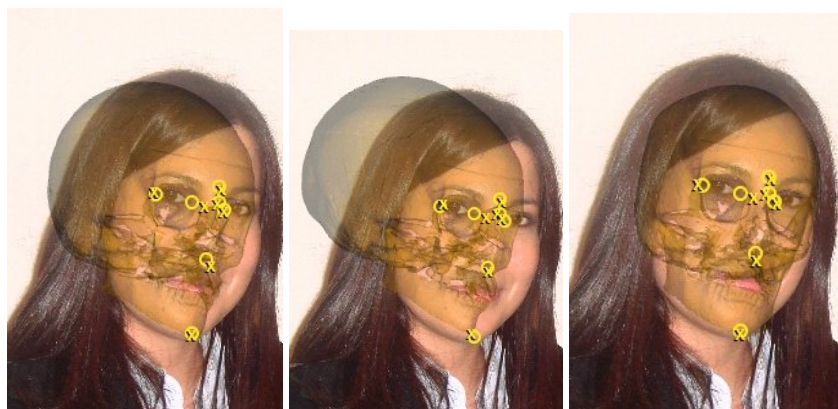


Fig. 2. Negative case study. From left to right. The best and worst results achieved using crisp landmarks. On the right, the best solution considering fuzzy landmarks (with Eq. 4 as fitness).

Notice that, a good superimposition in a negative case does not mean a bad performance of the algorithm but it is just the opposite. In the forensic identification process applied, the second and third stages are fully independent, and a good superimposition must be obtained in order to be able to properly determine if the skull and the face actually correspond to the same subject. Thus, we aim to achieve the best possible superimposition in any scenario. The decision making, corresponding to the next photographic supra-projection stage, will determine if the photo and the skull belong to the same person.

On the other hand, this negative case highlights the advantages of the approach based on the fuzzy location of landmarks. Besides the smaller variability of the results, the method is more robust to deal with different initializations of landmarks provided by a number of forensic experts.

5 Concluding Remarks and Future Works

We have proposed the use of fuzzy landmarks to tackle the uncertainty related to the landmark location task for craniofacial superimposition. We have presented and discussed superimposition results obtained on a positive and a negative identification case. We have simulated 10 forensic anthropologists, who would have manually extracted a set of landmarks from a 2D-image of a missing person. We have compared the automatic superimpositions considering crisp landmarks and those achieved using fuzzy landmarks. Promising results have been achieved, especially using the fitness given by equation 4. The superimposition results show that the fuzzy approach is able to model the inherent uncertainty involved in the location of the landmarks, i.e., none of the ten emulated experts are able to achieve a better result than the fuzzy approach. Thus, our proposal becomes more robust when tackling different locations of the landmarks. However, these results need to be confirmed in a more extensive study, with a larger number of cases.

As a future work, we aim to determine transformation parameters of the manual superimposition achieved by the forensic expert. We will use those parameters as the ground truth solution to measure the accuracy of the automatic superimposition processes. In addition, we are planning to introduce the already said matching uncertainty. We will make a poll between different forensic anthropologists in order to define the most appropriate shape and size of the fuzzy landmarks and for having sets of crisp landmarks. We also plan to use the extracted landmarks as sets of crisp landmarks for comparison purposes.

Finally, we aim to tackle the identification stage, i.e. the decision making by using fuzzy logic, in order to assist the forensic expert in the final identification decision.

Acknowledgements

This work was partially supported by the Spain's Ministerio de Educación y Ciencia (ref. TIN2006-00829) and by the Andalusian Dpto. de Innovación, Ciencia y Empresa (ref. TIC1619), both including EDRF fundings.

References

1. Iscan, M.Y.: Introduction to techniques for photographic comparison. In: Iscan, M.Y., Helmer, R. (eds.) *Forensic Analysis of the Skull*, pp. 57–90. Wiley, Chichester (1993)
2. Ballerini, L., Córdón, O., Damas, S., Santamaría, J.: Craniofacial superimposition in Forensic identification using genetic algorithms. Technical report ECSC AFE 2008-03. European Center for Soft Computing (2008)
3. Córdón, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L.: Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets and Systems* 141(1), 5–31 (2004)
4. Ballerini, L., Córdón, O., Damas, S., Santamaría, J., Alemán, I., Botella, M.: Identification by computer-aided photographic supra-projection: a survey. Technical report ECSC AFE 2007-04. European Center for Soft Computing (2007)
5. Nickerson, B.A., Fitzhorn, P.A., Koch, S.K., Charney, M.: A methodology for near-optimal computational superimposition of two-dimensional digital facial photographs and three-dimensional cranial surface meshes. *Journal of Forensic Sciences* 36(2), 480–500 (1991)
6. Blickle, T.: Tournament selection. In: Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.) *Handbook of evolutionary computation*, ch. 2.3. IOP Publishing Ltd/Oxford University Press (1997)
7. Eshelman, L.J.: Real-coded genetic algorithms and interval schemata. In: Whitley, L.D. (ed.) *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202. Morgan Kaufmann, San Mateo (1993)
8. Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.): *Handbook of evolutionary computation*. IOP Publishing Ltd/Oxford University Press (1997)
9. Richtsmeier, J., Paik, C., Elfert, P., Cole, T.M., Dahlman, F.: Precision, repeatability and validation of the localization of cranial landmarks using computed tomography scans. *Cleft Palate Craniofac J.* 32, 217–227 (1995)
10. Sinha, P.: A symmetry perceiving adaptive neural network and facial image recognition. *Forensic Science International* 98(1-2), 67–89 (1998)
11. Ghosh, A., Sinha, P.: An economized craniofacial identification system. *Forensic Science International* 117(1-2), 109–119 (2001)

A Minimum Risk Wrapper Algorithm for Genetically Selecting Imprecisely Observed Features, Applied to the Early Diagnosis of Dyslexia

Luciano Sánchez¹, Ana Palacios², and Inés Couso³

¹ Univ. Oviedo, D. Informática. 33071 Gijón, Asturias
luciano@uniovi.es

² Univ. Oviedo, D. Informática. 33071 Gijón, Asturias
anapalaciosg@gmail.com

³ Univ. Oviedo, D. Estadística 33071 Oviedo, Asturias
couso@uniovi.es

Abstract. A wrapper-type evolutionary feature selection algorithm, able to use imprecise data, is proposed. This algorithm is based on a new definition of a minimum Bayesian risk k-NN estimator for vague data. Our information about the risk is assumed to be fuzzy. Therefore, the risk is minimized by means of a modified multicriteria Genetic Algorithm, able to optimize fuzzy valued fitness functions.

Our algorithm has been applied to interval-valued data, collected in a study about the early diagnosis of dyslexia. We were able to select a low number of tests that are relevant for the diagnosis, and compared this selection of factors to those sets obtained by other crisp and imprecise data-based feature selection algorithms.

Keywords: Imprecise Data, Evolutionary Feature Selection, Wrapper-based Feature Selection, Dyslexia.

1 Introduction

Most of the feature selection algorithms used in the design of classification systems are suitable for precise, numerical data, without observation error neither missing values [2]. On the contrary, in this paper we address a problem where the data has low quality and most methods in the literature cannot be applied. In previous works, we have proposed a filter-type evolutionary algorithm called FMIFS [7] and also a wrapper-type evolutionary feature selection algorithm called FSSGA [8]. These two algorithms are, to our best knowledge, the only algorithms able to discover the most relevant features in classification problems with interval-valued or fuzzy observations of either the features or the class. In [8] we concluded that the wrapper algorithm did not improve the filter approach, suggesting that the instrumental classifier used in FSSGA (our own generalization of the k-NN rule to vague data) was suboptimal. In this paper we propose a different extension of this rule, that takes into account different costs of misclassification, and analyze the impact of the new definition in a practical problem: the early diagnosis of dyslexia.

1.1 Early Diagnosis of Dyslexia

Dyslexia is a learning disability in people with normal intellectual coefficient, and without further physical or psychological problems that can explain such disability. It has been estimated that between 4% and 5% of schoolchildren have dyslexia, with reading and writing problems [1]. The average number of children in a Spanish classroom is 25, therefore most classrooms have dyslexic children. Dyslexia may become apparent in early childhood, with difficulty putting together sentences and a family history. Recognition of the problem is very important in order to give the infant an appropriate teaching.

We intend to design a fuzzy rule-based testing system that can be used by unqualified personnel while screening the children for dyslexia. As we will show later, the diagnosis will be based on the responses of children to certain tests. Many of these tests are based on drawings, whose interpretation is subjective. The psychologist does not make a crisp classification, either, as she labels each individual with an interval of values between 0 and 4. The same happens with the results of the tests, which can be assigned linguistic or interval values. Therefore, the main difficulty with this study is dealing with low quality data (incomplete items, intervals, lists, subjective values, etc.) Moreover, there are some hundreds of different tests that can be evaluated, thus one of the first challenges of our study is to select a small, representative subset of tests that can be used to diagnose the disability.

1.2 Objectives and Summary

From a theoretical point of view, the main novelty in this work (beside the use of evolutionary techniques for selecting features of imprecisely perceived data) is in the definition of the instrumental classifier which is used in the genetic search. In this paper, we will not use a fuzzy extension of the standard k-NN, as we did in [8], but we will derive a new rule based on the minimum Bayesian risk. In words, we want to use a classifier where the cost of labeling a child depends on the amount of error: the cost of assigning the class '4' to an individual of type '1' has to be higher than the cost of assigning the class '0' to it.

The study of this new classifier will be carried on in Section 2. In Section 3, we will briefly introduce the practical problem and show some preliminary results. Lastly, Section 4 concludes the paper. It is remarked that the implementation of the genetic algorithm used in the feature selection is identical to that explained in [8], and, because of space reasons, it is not repeated in this paper.

2 The k-NN Rule with Uncertain Data and the Minimum Bayesian Risk Classifier

The most frequent use of the term “fuzzy k-NN” is described in [4]. In that paper, a membership value for each crisp example in the training dataset is introduced, and the class of the object is assigned to the class with higher certainty, in a procedure similar to a statistical kernel classifier. It is remarked that, in that reference, all features were

assumed to be crisp. In [9], the problem where the class of an object is a fuzzy value was addressed. Furthermore, in [8], a k-NN algorithm able to use imprecise features was defined. In the following section, we extend this last result to the case where the costs of misclassifying objects depend on the class.

2.1 An Extended Definition of the k-NN Criterion for Imprecise Data and Misclassification Costs

Let the class of an object ω be $\text{class}(\omega)$. We want to define a decision rule that maps any set of measurements $X(\omega)$ to $\text{class}(\omega)$, with the lowest risk. Let C_{ij} the cost of classifying an object of class j as of being of class i , and $C_{ii} = 0$. The risk of classifying the object ω in class i is

$$r_i(x) = \sum_j C_{ij} P(\text{class}(\omega) = j \mid X(\omega) = x). \tag{1}$$

The minimum risk rule is

$$c(x) = \arg \min_{i=1,\dots,M} r_i(\omega) \tag{2}$$

where M is the number of classes.

Let us suppose now we are given a sample of objects $\Omega = \{\omega_1, \dots, \omega_N\}$. N_j elements of this sample belong to the j -th class, thus $N = \sum_{j=1,\dots,M} N_j$. For estimating $P(j|x)$ from data, we assume that $f(x|j)$ is constant in an small neighborhood of x , as follows: let V be the smallest ball, centered in x , that contains k objects of the sample. Let $n_j, k = \sum_{j=1,\dots,M} n_j$ be the number of objects of the j -th class contained in V . If V is small enough, then

$$r_i(x) = \sum_j C_{ij} \frac{f(x|j)p(j)}{f(x)} \approx \frac{\sum_j C_{ij} \frac{n_j}{N_j V} \frac{N_j}{N}}{\frac{k}{NV}} = \frac{\sum_j C_{ij} n_j}{k}. \tag{3}$$

Let us assume that $k = 1$, i.e. V is the smallest volume, centered in x , that contains one point of the sample. Let q be the class of this point. Applying eq. (2), x is assigned the class

$$c(x) = \arg \min_{i=1,\dots,M} \sum_j C_{ij} n_j = \arg \min_{i=1,\dots,M} C_{iq}. \tag{4}$$

Since we have assumed that $C_{ii} = 0$ for all i , eq. (4) means that the minimum risk rule is also the nearest neighbor rule: we will assign to x the class of the nearest point in the training set.

2.2 Interval-Valued Measurements

Nevertheless, the nearest neighbor rule does not hold if we do not accurately perceive the points in the sample, but balls containing the measurements. For example, consider the situation in Figure 1: the smallest volume V we are certain that contains one element of the sample, also intersects two other objects, but it does not completely contain them. In the same example, the application of eq. (2) requires deciding whether

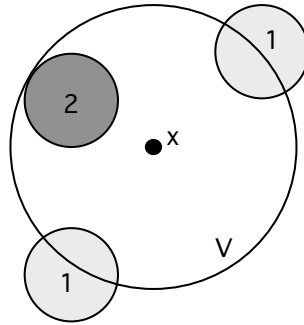


Fig. 1. Interval-valued data: The smallest ball centered in x that contains for sure one object has non-null intersection with two other objects. Therefore, $P(1|x) \in [0, 2/3]$ and $P(2|x) \in [1/3, 1]$. Depending on the relative values of C_{12} and C_{21} , we may assign either the label 2 or the set of labels $\{1, 2\}$ to x .

$C_{21}[0, 2/3] \geq C_{12}[1/3, 1]$. If $C_{21} \geq 0.5 \cdot C_{12}$, then we do not have information enough to know the response, thus we will label the example x with the whole set $\{1, 2\}$.

When the measurements taken on the objects ω_i of the training set are described by the sets $\Gamma(\omega_i)$, we propose to define the minimum risk classifier as follows:

$$c_\Gamma(x) = \left\{ i \mid \sum_j C_{ij}n_j \leq \sum_j C_{qj}n_j, \quad \forall q \in \{1, \dots, M\}, n_{j^*} \leq n_j \leq n_j^* \right\} \quad (5)$$

where $n_j^* = \#\{i \mid \text{class}(\omega_i) = j \text{ and } \Gamma(\omega_i) \cap V \neq \emptyset\}$, $n_{j^*} = \#\{i \mid \text{class}(\omega_i) = j \text{ and } \Gamma(\omega_i) \subseteq V\}$ and, in turn, V is the smallest sphere, centered in x , for which $\exists i \in \{1, \dots, N\} \mid \Gamma(\omega_i) \subseteq V$.

2.3 Fuzzy-Valued Measurements

The fuzzy case is an extension of the interval case. Let us consider, for instance, the data displayed in Figure 2. We have a one-dimensional problem, and we want to label the point $x = 7$, according to the minimum risk rule. We have three imprecisely measured objects surrounding x : two of them, the triangular fuzzy numbers $(1; 3; 5)$ and $(9; 11; 13)$, belong to class 1. A third one, $(4; 6; 8)$, belong to class 2. The smallest volume, centered in 7, that contains one element of the sample, is the interval $[4, 10]$. Now observe that each α -cut of these three sets forms an interval-valued classification problem. For levels greater than 0.5, the only object in V is that of class 2. However, for $\alpha \leq 0.5$, V intersects with the three objects. For these cuts, applying eq. (5) we obtain

$$c(X) = \begin{cases} \{1, 2\} & C_{21} \geq 0.5 \cdot C_{12} \\ \{2\} & \text{otherwise.} \end{cases} \quad (6)$$

Therefore, our knowledge about the class of the point x is given by the fuzzy set

$$c(X) = \begin{cases} 0.5/1 + 1/2 & C_{21} \geq 0.5 \cdot C_{12} \\ 1/2 & \text{otherwise.} \end{cases} \quad (7)$$

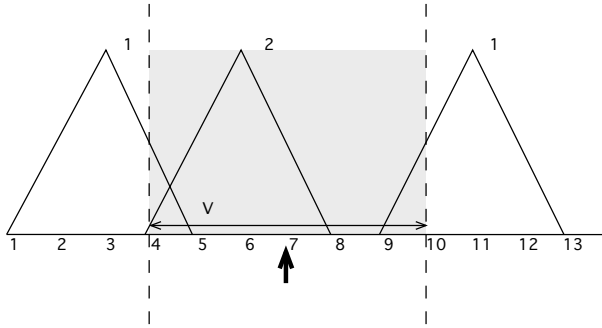


Fig. 2. Fuzzy data: The smallest volume centered in 7 that completely contains one object is the interval $V = [4, 10]$. Let $C_{12} = 1$, $C_{21} = 0.75$, $C_{11} = C_{22} = 0$. For α -cuts lower than 0.5, the risks $\sum_i C_{i1}P(1|x)$ and $\sum_i C_{i2}P(2|x)$ are $[0, \frac{2}{3} \cdot 0.75]$ and $[\frac{1}{3} \cdot 1, 1 \cdot 1]$ respectively, whose intersection is not null. For $\alpha > 0.5$, $\sum_i C_{i1}P(1|x) < \sum_i C_{i2}P(2|x)$. Therefore, the class of x is the fuzzy set $0.5/1 + 1/2$.

If the perception of the measurements taken on the object ω is given by the fuzzy set $\tilde{X}(\omega)$, we propose to define the minimum risk classifier as follows:

$$c_{\tilde{X}}(x)(t) = \sup\{\alpha \mid t \in c_{\tilde{X}_\alpha}(x)\} \tag{8}$$

where $c_{\tilde{X}_\alpha}(x)$ is the set of classes assigned to the point x by eq. (5) when the information about the measurements of the objects in the sample is given by the cuts $[\tilde{X}(\omega_i)]_\alpha$.

It is emphasized that, for classifying either a crisp or a fuzzy set instead of a point, the volume V must completely contain k elements of the sample and also the whole area being classified.

2.4 Measurement of the Risk of a Classifier with Imprecise Data

For computing the risk of the classifier over the training set, we add the costs of all the classifications:

$$R = \sum_i C_{c(x_i)\text{class}(\omega_i)}. \tag{9}$$

In case the output of the classifier is the set of classes induced by the interval measurements $\Gamma(\omega_i)$, the risk is, in turn, a set of values. Let

$$r_{i\Gamma^*} = \min\{C_{t,\text{class}(\omega_i)} \mid t \in c_\Gamma(x_i)\} \tag{10}$$

$$r_{i\Gamma}^* = \max\{C_{t,\text{class}(\omega_i)} \mid t \in c_\Gamma(x_i)\}. \tag{11}$$

We know that the risk is bounded as follows:

$$\sum_i r_{i\Gamma^*} \leq R_\Gamma \leq \sum_i r_{i\Gamma}^*. \tag{12}$$

Lastly, if the output of the classifier is a fuzzy set of classes induced by the fuzzy measurements $\tilde{X}(\omega_i)$, the risk of the i -th example is the fuzzy set

$$r_{i\tilde{X}}(\alpha) = \sup\{\alpha \mid u \in \text{conv}\{C_{t,\text{class}(\omega_i)} \mid t \in c_{\tilde{X}_\alpha}(x_i)\}\} \tag{13}$$

Category	Test	Description
Verbal compr.	BAPAE	Vocabulary
	BADIG	Verbal orders
	BOEHM	Basic concepts
Logic reasoning	RAVEN	Color
	BADIG	Visual memory
Sensory motor skills	BENDER	Visual-motor coordination
	BADIG	Perception of shapes
	BAPAE	Spatial relations, Shapes
	STAMBACK	Audition, Rhythm
	HARRIS/HPL	Laterality, Pronunciation
	GOODENOUGHT	Spatial orientation
Reading	TALE	Analysis of reading and wr.

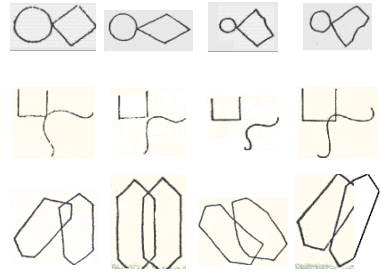


Fig. 3. Left: Categories of the tests currently applied in Spanish schools for detecting dyslexia in children between 5 and 8 years. Right: Example of some of Bender’s tests for detecting dyslexia. Upper part: The angles of the shape in the right are qualified by a list of adjectives that can contain the words “right,” “incoherent,” “acceptable,” “regular” and “extra.” Middle and lower part: The relative position between the figures can be “right and separated,” “right and touching,” “intersecting”, etc.

where $\text{conv}(\cdot)$ means “convex hull”, and the global risk is the fuzzy set

$$R_{\tilde{X}} = \bigoplus_i r_{i\tilde{X}}. \tag{14}$$

3 Practical Application: Symptoms and Detection of Dyslexia

In the left part of Figure 3, we have listed the categories of the tests in our experimentation. In the right part of the same figure, we have included an example of one of the tasks that the children have to solve in these tests: copying some geometric drawings. When a child is being evaluated, the expert has to decide whether the angles, relative position and other geometrical properties have been accurately copied or not, choosing between a given set of adjectives. Other tests produce numbers, or linguistic labels as “low”, or “very high”. Lastly, we allow the use to express indifference between different responses by means of intervals, as in “lower than 3” or “between 2 and 4”. There are 13 categories of tests, that expand to a total of 413 numerical, categorical and interval-valued variables.

We have selected a sample of 65 infants between 5 and 8 years old, in urban schools of Asturias (Spain), and collected their responses to the 413 tests mentioned before. Afterwards, the same children were examined by a psychologist, who assigned each one of them a subjective score, which is an interval of values between 0 (normal child) and 4 (high degree of dyslexia). In Figure 4 we display the matrix of costs suggested by the expert. The cost of an interval classification has been computed as the centerpoint of the set of costs: $C(A, B) = \text{CoG}\{t \mid t = C(a, b), a \in A, b \in B\}$ (for example, $C(1, [2, 3]) = \text{CoG}([0.75, 1]) = 0.875$).

The objective of this study is to relate these measured variables with the expert judgement of the professional, and highlight those factors that are involved in the early

	0	1	2	3	4
0	0	0.75	1	1	1
1	0.75	0	0.75	1	1
2	1	0.75	0	0.75	1
3	1	1	0.75	0	1
4	1	1	1	1	0

Fig. 4. Costs of the misclassification, according to the human expert

development of the dyslexia during the preschool age. With this aim, we have designed a Fuzzy Rule Based System (FRBS), with modified fuzzification and defuzzification interfaces, and an extended reasoning method [5], capable of handling interval and fuzzy data. The inputs of this FRBS are the responses to a selected set of tests, and the output is a set of classes, modeling the dyslexia degree and our confidence in the estimation. The Knowledge base of the FRBS was obtained by a Pittsburgh class GFS. The multi-criteria GA mentioned in the introduction is defined in [6].

For selecting the most representative set of tests, we have used six different methods: the weighted Fuzzy SSGA proposed in this paper (WFSSGA), the Fuzzy SSGA method defined in [8], the FMIFS algorithm defined in [7] and three crisp methods: WSSGA, SSGA and MIFS. These three last methods are the counterparts of the fuzzy algorithms when each imprecise feature is replaced by its centerpoint. A maximum of 6 features were allowed.

The mean values of the risks of the corresponding FRBSs are displayed in Figure 5. The dispersion of the results are shown in Figure 6. Being intervals, we have plotted the boxes describing the lower and upper bounds of each method. 10 repetitions of each experiment were made, with a 10-cv experimental design.

Given this preliminary data, we can draw two conclusions: First, the use of fuzzy fitness based techniques obtains more information about the sample than standard

	WFSSGA	FSSGA	FMIFS	WSSGA	SSGA	MIFS
Train risk	[0.637,0.661]	[0.729,0.764]	[0.684,0.711]	[0.992,0.992]	[1.000,1.000]	[0.955,0.963]
Test risk	[0.743,0.764]	[0.856,0.921]	[0.772,0.814]	[1.000,1.000]	[1.000,1.000]	[1.000,1.000]

Fig. 5. Mean values of risk in training and test sets, for the six methods studied in this paper. The values shown are means of 10 repetitions of the experiments, 10-cv cross validation.

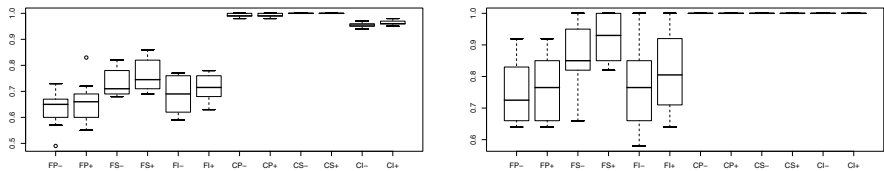


Fig. 6. Boxplots of the bounds of the estimated risks of the FRBSs learned from the selection of features obtained by the six methods studied. The columns are: WFSSGA: [FP-, FP+]. FSSGA: [FS-, FS+]. FMIFS: [FI-,FI+]. CP, CS, CI are the crisp versions of the same algorithms.

techniques. None of the crisp algorithms was able to find a representative set of 6 tests. Second: a wrapper algorithm that minimizes the Bayesian risk is more efficient than both a wrapper that minimizes the total error and information theory based methods.

4 Concluding Remarks

We have proposed an extension of the k-NN rule that approximates the minimum Bayesian risk classifier from low quality data. This classifier has been included in a wrapper-type evolutionary feature selection algorithm. This algorithm has improved other feature selection algorithms based on mutual information and minimum Bayesian error, k-NN-based wrappers, and has been used to find a meaningful set of tests for the early diagnosis of dyslexia.

Acknowledgements

This work was funded by Spanish M. of Science and Technology, grants TIN2005-08036-C05-05 and TIN2007-67418-C03-03, and by Principado de Asturias, PCTI 2006-2009.

References

1. Ajuriaguerra, J.: Manual de psiquiatría infantil (in spanish). Toray-Masson (1976)
2. Cantú-Paz, E.: Feature Subset Selection, Class Separability, and Genetic Algorithms. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 959–970. Springer, Heidelberg (2004)
3. Casillas, J., Cordon, O., del Jesus, M.J., Herrera, F.: Genetic Feature Selection in a Fuzzy Rule-Based Classification System Learning Process for high-dimensional problems. *Information Sciences* 136, 135–157 (2001)
4. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k -nearest neighbor algorithm. *IEEE Trans. Systems Man Cybernet.* 15(4), 580–585 (1985)
5. Sánchez, L., Couso, I., Casillas, J.: A Multiobjective Genetic Fuzzy System with Imprecise Probability Fitness for Vague Data. In: Proc. Int. Symp. on Evolving Fuzzy Systems, pp. 131–136 (2006)
6. Sánchez, L., Couso, I., Casillas, J.: Modelling vague data with genetic fuzzy systems under a combination of crisp and imprecise criteria. In: Proc. 2007 IEEE MCDM, Honolulu, USA (2007)
7. Sánchez, L., Suárez, M.R., Villar, J.R., Couso, I.: Mutual Information-based Feature Selection and Partition Design in Fuzzy Rule-based Classifiers from Vague Data. In: *Int. J. Approximate Reasoning* (in press, 2009)
8. Sánchez, V.J.R., Couso, I.: Genetic Feature Selection for Fuzzy Discretized Data. In: IPMU 2008, Málaga (in press, 2008)
9. Sarkar, M.: Fuzzy-rough nearest neighbor algorithms in classification. *Fuzzy Sets and Systems* 158, 2134–2152 (2007)

An Approach to Flocking of Robots Using Minimal Local Sensing and Common Orientation

Iñaki Navarro¹, Álvaro Gutiérrez², Fernando Matía¹, and Félix Monasterio-Huelin²

¹ Intelligent Control Group, Universidad Politécnica de Madrid,
José Gutiérrez Abascal 2, E-28006 Madrid, Spain
{inaki.navarro, fernando.matia}@upm.es

² Departamento de Tecnologías Especiales Aplicadas a la Telecomunicación,
Universidad Politécnica de Madrid,
Avd. Ciudad Universitaria s/n, E-28040 Madrid, Spain
aguti@etsit.upm.es, felix.monasteriohuelin@upm.es

Abstract. A new algorithm for the control of robot flocking is presented. Flocks of mobile robots are created by the use of local control rules in a fully distributed way, using just local information from simple infra-red sensors and global heading information on each robot. Experiments were done to test the algorithm, yielding results in which robots behaved as expected, moving at a reasonable velocity and in a cohesive way. Up to seven robots were used in real experiments and up to fifty in simulation.

Keywords: Distributed Robot Systems, Robot Flocking, Mobile Robot.

1 Introduction

There are many applications in which a multi-robot approach is an advantage compared to single robot systems. Groups of robots moving together can act as sensor arrays, allowing them to locate a desired source in a more effective way. Thus, formations of robots can be very useful in search tasks, especially when spatial pattern of the source is complex, like in odor [1] or sound [2] cases. They are also useful in mapping tasks [3], since measurement redundancy allows robots to build more accurate maps.

One of the basic problems in multi-robot systems is how to make mobile robots move together as a group, behaving as a single entity. This problem is solved by flocking and formations of robots. In flocking problem robots move as a group but the shape and relative positions between the robots are not fixed, allowing robots to move within the group. The first work about artificial flocking was a computer graphic animation of a group of birds [4]. Some characteristic examples of robot flocking in which a theoretical effort is made are [5, 6], while [7] is focused on experiments with real robots. On the other hand, a robot formation can be defined as a group of robots that is able to maintain pre-determined positions and orientations between its members at the same time that it moves as a whole [8, 9].

A desirable characteristic that flocking of mobile robots may have is scalability in the number of robots. In order to have this scalability, local sensing and communications and a decentralized controller are necessary [10].

In this paper we propose a distributed and scalable algorithm for the control of mobile robots flocking that uses very simple proximity sensors and information about their own absolute headings. The main advantages are the simplicity of the sensors required for the flocking and the scalability of the algorithm. The platform used to test the designed algorithm is described in Sect. 2. In Sect. 3, the proposed algorithm is explained. Experiments performed to test the algorithm and their results are described and discussed in Sect. 4. Finally the conclusions and future work can be found in Sect. 5.

2 Experimental Test Platform

The proposed algorithm uses local sensing to detect the distance and angle between nearby robots, and information about its own global heading. Thus, the *e-puck* robots, used to test the algorithm, need to have these capabilities. They are small wheeled cylindrical robots, 7cm of diameter, equipped with 8 infra-red proximity sensors distributed around their bodies, 3 infra-red ground sensors and a 3-axis accelerometer. Their mobility is ensured by a differential drive system.

The infra-red sensors are used to estimate the distance to the neighboring robots, while the angle to the nearby robot is approximated by the direction of the infra-red sensor. Every obstacle seen by each sensor is considered as a neighboring robot, so no real obstacles are placed or considered in the experiments. The range of the infra-red sensors is about 12cm. A picture of the *e-puck* robot and the distribution of its infra-red sensors can be seen in Fig. 1.

In order to know its heading in a global coordinate system, robots move in a vertical plane, using accelerometer sensors to obtain a global orientation. A magnetic cubic extension is added to the bottom of the robots, permitting the robots to move attached to a metallic wall. A picture of the robot with the magnets is shown in Fig. 1c. This modification allows us to design a virtual compass using the 3-axis

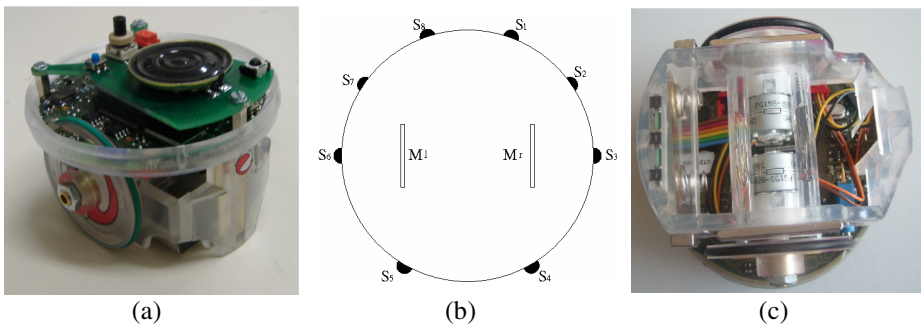


Fig. 1. (a) The *e-puck* robot. (b) Distribution of the infra-red sensors. (c) Magnets placed at the bottom of the robot.

accelerometer thanks to the gravity force. Robots sense the gravity on x and y accelerometer's axis, giving $\tan(y/x)$ the global heading. A preliminary calibration is needed on every robot to overcome with the accelerometer bias.

Communication between robots is necessary in some parts of the algorithm. It is implemented through bluetooth and a central computer. *E-pucks* transmit information to a computer that is redirected to the robots

Webots simulator [11] was used initially to test the algorithm, using a realistic model of the robots. It allowed us to perform the experiments in a fast way, tuning the different parameters easily, and using up to 50 robots. A compass was added to the *e-puck* model in order to know the own heading in global coordinates.

3 Algorithm

The developed algorithm is fully distributed among the robots, allowing the robots to move as a group in a common pre-defined direction, by just using local information from the infra-red proximity sensors and global orientation.

Each robot reacts to every object detected by its infra-red sensors, being attracted or repelled depending on the measured distance. This makes that the robots try to maintain a desired distance between them. Each robot generates a virtual velocity $V_{aggregation}$ as the sum of the reactions to nearby robots:

$$V_{aggregation} = \sum V_i \quad (1)$$

The magnitude and angle of V_i are defined as follows:

$$|V_i| = \begin{cases} K_1(desiredDist - dist_i), & \text{if } dist_i \leq desiredDist \\ K_2(dist_i - desiredDist), & \text{if } desiredDist < dist_i \leq \max Dist \\ 0, & \text{if } \max Dist < distSensor_i \end{cases} \quad (2)$$

$$\arg(V_i) = \begin{cases} angle_i, & \text{if } dist_i \leq desiredDist \\ angle_i + \pi, & \text{if } desiredDist < dist_i \leq \max Dist \\ 0, & \text{if } \max Dist < distSensor_i \end{cases} \quad (3)$$

where *desiredDist* is the desired distance that robots are supposed to maintain between them; *dist_i* is the measured distance by a sensor; *maxDist* is a threshold indicating that every measure above should not be considered as a robot; and *angle_i* is the position in radians of the sensor, that represents an approximation of the direction of the detected nearby robot. K_1 and K_2 are the adjusting parameters of the proportional controller.

In order to move in the pre-defined desired direction, each robot reacts generating another desired virtual velocity $V_{desiredDirection}$, defined by its magnitude and angle as follows:

$$|V_{desiredDirection}| = K_3 \quad (4)$$

$$\arg(V_{desiredDirection}) = desiredDirection - myHeading \tag{5}$$

where *desiredDirection* is the desired common direction of movement, and *myHeading* is the robot heading expressed in the same coordinate system as *desiredDirection*.

As a result from this virtual velocity $V_{desiredDirection}$, robots would move in the same direction and approximately at the same speed. In order to make the robots to move together as a group in the same direction and maintaining the desired distance between them, both virtual velocities are added resulting in the final total virtual velocity:

$$V_{total} = V_{aggregation} + V_{desiredDirection} \tag{6}$$

Since robots used in the experiments are not purely holonomic, the total virtual velocity must be translated in the appropriate motor speeds, approximating the desired velocity by making use of the *Low Level Controller* described in next the section.

3.1 Low Level Controller

The *Low Level Controller* (LLC) is designed to translate the virtual velocity in mobile robots with differential drive configuration. The controller is inspired by a similar one described by Hsu [12], but allowing backwards movement. The angle (θ) and magnitude ($|V|$) of the virtual velocity are the inputs for getting the angular and linear velocities:

$$V_{linear} = K_4 |V| \cos(\theta) \tag{7}$$

$$V_{angular} = \begin{cases} K_5(\theta + \pi), & \text{if } \theta < -\pi/2 \\ K_5\theta, & \text{if } \pi/2 > \theta > -\pi/2 \\ K_5(\theta - \pi), & \text{if } \theta > \pi/2 \end{cases} \tag{8}$$

From (7) it can be seen that the linear velocity (V_{linear}) is proportional to the magnitude of the virtual velocity, and it is multiplied by $\cos(\theta)$. This makes velocity maximum when $\theta = 0$ or $\theta = -\pi/2$, and minimum ($V_{linear} = 0$) when $\theta = \pi/2$ or $\theta = -\pi/2$. This is natural since a desired movement towards $\theta = \pi/2$ or $\theta = -\pi/2$ is not possible because of the kinematics of the robot. The robot will move forwards when $-\pi/2 > \theta > \pi/2$ and backwards otherwise. The angular velocity ($V_{angular}$) is proportional to θ for $-\pi/2 > \theta > \pi/2$, when moving forwards, and proportional to $\theta + \pi$ when moving backwards. Thus, the robot will reach the desired heading that depends on if it is moving forward or backwards.

The sum of the linear and angular velocities is translated to motor speeds taking into account the kinematics of differential drive robot as follows:

$$s_{motor-right} = V_{linear} + B * V_{angular} \tag{9}$$

$$s_{motor-left} = V_{linear} - B * V_{angular} \tag{10}$$

where B is half the distance between the two wheels, $s_{motor-right}$ the speed of the right motor and $s_{motor-left}$ the speed of the left motor.

By applying the LLC to V_{total} in all the robots, it results in a flocking of the robots towards the pre-defined direction.

3.2 Search for Lost Flock Algorithm

Since robots' sensor range is just $12cm$, eventually a robot may stop detecting any neighboring robot and might not follow the flock. In order to overcome with that problem, a simple algorithm to look for the group of robots has been designed and implemented.

When a robot loses the flock it first orientates in the direction of the last seen robot. After that it moves during few seconds in that direction. If the flock is still not found, the lost robot moves in the direction that the flock is moving, this is, *desiredDirection*. If after a certain time the flock is not found the robot consider itself as completely lost and stops. When the robot finds the flock it quits the searching algorithm and continues with the general control algorithm.

This recovery algorithm works quite well, usually during the first seconds the robot finds the flock, partially because it moves 50% faster than the flock.

4 Experimental Results

In order to test the quality and scalability of the algorithm, experiments have been done both in simulation and with real robots. In simulation, experiments have been repeated with 7 and 50 robots, while in the real environment just 7 of them were used.

Three types of experiments were done in simulation. In *Type 1* experiments, robots move in an unbounded arena without borders always in the same direction. In *Type 2* experiments, the arena has borders marked on the floor that robots are able to detect. When one robot detects that it has arrived to the border it communicates that it has reached the limit to the rest of the flock and the direction of movement is inverted. In *Type 3* experiments, robots move in an unbounded arena but they change their desired direction of movement progressively with time, making a complete turn in $50s$. The aim of this experiment is to prove that the algorithm might be used in combination with a higher level algorithm to make more complicate tasks and not only move in a linear direction. With real robots just *Type 2* experiments were done.

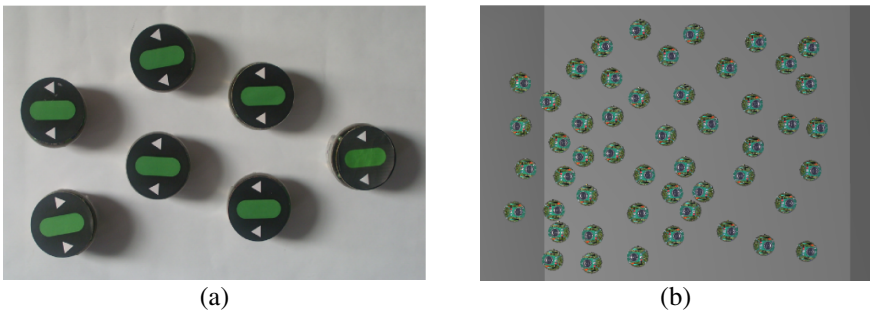


Fig. 2. (a) Flock of 7 real *e-puck* robots. (b) Flock of 50 *e-puck* robots in simulation.

Thirty experiments of 180s were done for each type of experiments and number of robots in simulation and reality. In the experiments with real robots, positions and headings of every robot were stored every 100ms using a tracking software tool [13].

Three parameters were measured to analyze the performance of the algorithm: *i)* group velocity, this is, the velocity of the center of mass of the group; *ii)* the area given by the convex hull [14], this is, the area of the minimum convex polygon that contains all the robots; and *iii)* the polarization that is a measure of how well aligned are the headings of the robots.

The area measurement is used to identify if the area of the group grows too much, that will indicate that the flock is being split in small groups. Polarization $P(G)$ of a group of robots G is defined as the mean vector angle deviation between the group heading and each individual heading [15]. If all robots are aligned, then $P(G)=0$. Conversely, if headings are evenly distributed, $P(G)=2\pi$. Lastly, if headings are random, i.e. drawn from a uniform distribution, then $P(G)=\pi$ in average. Like robots in the experiment are able to move backwards and forwards, the polarization measure is $P(G)=\pi$ for robots evenly distributed and $P(G)=\pi/2$ for random uniform distribution.

All the systematic experiments done worked well since robots were able to move as a whole in the pre-defined direction, and at the expected group velocities. In Fig. 2, two pictures of flocks in simulation and of real robots can be seen.

4.1 Results in Simulation

Figure 3a shows how the area of a group of 7 simulated robots evolves on average for the three different types of experiments. As it is observed in the graph, robots start increasing the area and after a period of time ($t=120s.$) they reach a stable plateau of about $0.1m^2$. The group responds as a unique individual, moving on the environment at constant velocity of $0.05m/s$, after the first seconds in which the flock is created. Small oscillations are observed on its steady-state as shown in Fig. 3b.

Results with 50 robots are similar to the ones with 7 robots as it can be seen in Fig. 4. Robots reach its steady-state area at $t=140s$, which is maintained during the rest of the experiment. There is no difference in the group velocity between experiments with 7 and 50 robots.

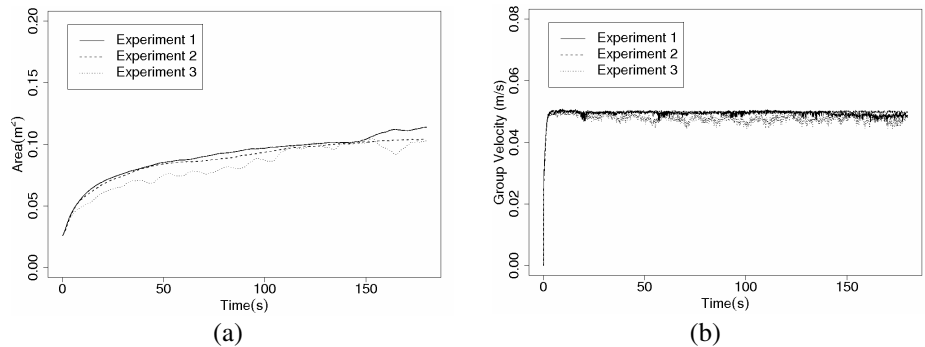


Fig. 3. Results for a group of 7 simulated robots for the three different experiments: (a) area on average (30 experiments), (b) group velocity on average (30 experiments)

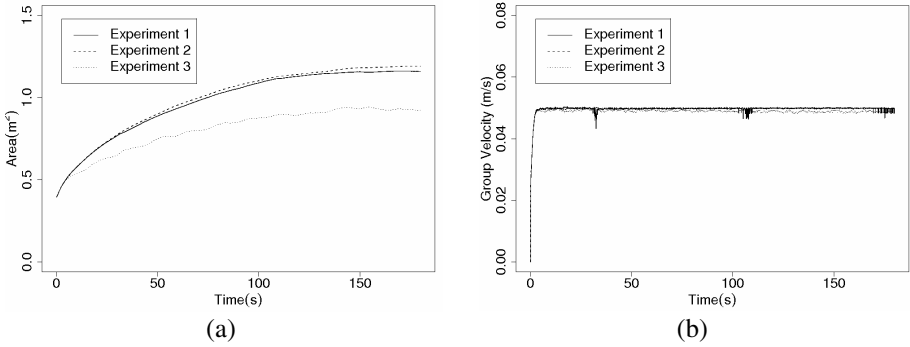


Fig. 4. Results for a group of 50 simulated robots for the three different experiments: (a) area on average (30 experiments), (b) group velocity on average (30 experiments)

In both group sizes, polarization starts at a approximate value of $\pi/2rad$ converging to a value of $0.05rad$ in $t=10s$. Polarization reaches its minimum at the same time as the group velocity is maximum.

4.2 Results with Physical Robots

Experiments with real robots were all of them of *Type 2*. As it can be seen in Fig. 5, their performance is similar to the one in the simulation case, with very small differences in the velocity and area values. There is a first phase when robots try to adopt similar headings, that lasts to $t=15s$, when the group reduces its polarization value down to $0.07rad$. After this phase, robots keep moving at constant velocity while maintaining the orientation. Group velocity is reduced from what expected from simulation to $0.04m/s$. This decrement might be due to the magnets which may be introducing friction in the movement, but also to the dynamics derived of moving in the vertical plane. Area increases in similar way as the one in simulation, up to about $0.13m^2$.

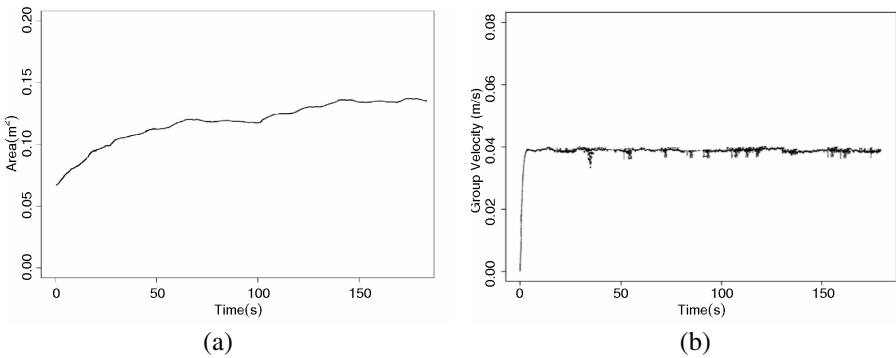


Fig. 5. Results for a group of 7 real robots for the three different experiments: (a) area on average (30 experiments), (b) group velocity on average (30 experiments)

5 Conclusions and Future Work

The presented algorithm works well according to the carried out experiments. A flock of mobile robot results from the local interactions between the robots, moving at the desired velocity and in a cohesive way. It was proved that the algorithm works with real robots, using simple real infra-red sensors and global heading provided by the *on-board compass*, while in [7] the sensors need to be emulated. Experiments in simulation with 50 robots show the scalability on the number of robots of the algorithm, which represent an advantage compared to other implementations like [9]. Both group velocity and polarization have reasonable values with real robots and in simulation. In addition, the absence of any robot leader and the use of many robots make the flock tolerant to the failure of any of its robots. The good performance of *Type 3* experiments shows that the algorithm could be used for tasks that would need of turns of the flock.

The use of a real compass on each robot, instead of the virtual one using accelerometers, would make the experiments easier since the use of magnets and vertical movement was a limiting factor in the velocity and smoothness of the movements. In addition, if obstacles need to be avoided, a system like the one presented in [16] to detect nearby robots and differentiate from other objects, will be necessary.

Acknowledgments. Authors want to acknowledge the ASL, LIS and SWIS laboratories of the Ecole Polytechnique Fédérale de Lausanne (EPFL) for the design of the *e-puck* robot, and Michele Leidi and Tarek Baaboura of EPFL for their help and ideas about the magnets on the *e-puck*. I. Navarro and F. Matía are partially sponsored by Spanish Ministry of Science (ROBONAUTA project DPI2007-66846-C02-01). I. Navarro is sponsored by Madrid Region with a Ph.D. grant (FPI-2005).

References

1. Hayes, A.T., Martinoli, A., Goodman, R.M.: Distributed Odor Source Localization. *IEEE Sensors Journal* 2(3), 260–271 (2002)
2. Pugh, J., Martinoli, A.: Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In: *IEEE Swarm Intelligence Symposium*, Piscataway, NJ, pp. 332–339. IEEE Press, Los Alamitos (2007)
3. Howard, A.: Multi-robot mapping using manifold representations. In: *IEEE International Conference on Robotics and Automation*, Piscataway, NJ, vol. 4, pp. 4198–4203. IEEE Press, Los Alamitos (2004)
4. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21(4), 25–34 (1987)
5. Tanner, H., Jadbabaie, A., Pappas, G.: Stable flocking of mobile agents, part i: fixed topology. In: *Proceedings. 42nd IEEE Conference on Decision and Control*, Piscataway, NJ, vol. 2, pp. 2010–2015. IEEE Press, Los Alamitos (2003)
6. Canepa, D., Potop-Butucaru, M.G.: Stabilizing Flocking Via Leader Election in Robot Networks. In: *Stabilization, Safety, and Security of Distributed Systems*, pp. 52–66. Springer, Heidelberg (2007)
7. Hayes, A.T., Dormiani-Tabatabaei, P.: Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots. In: *IEEE International Conference on Robotics Automation*, Piscataway, NJ, pp. 3900–3905. IEEE Press, Los Alamitos (2002)

8. Fredslund, J., Mataric, M.J.: A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation*, Special Issue on Multi Robot Systems 18, 837–846 (2002)
9. Balch, T., Hybinette, M.: Social potentials for scalable multi-robot formations. In: *IEEE International Conference on Robotics and Automation*, Piscataway, NJ, vol. 1, pp. 73–80. IEEE Press, Los Alamitos (2000)
10. Sahin, E.: Swarm robotics: From sources of inspiration to domains of application. In: Sahin, E., Spears, W. (eds.) *Swarm Robotics 2004*. LNCS, vol. 3342, pp. 10–20. Springer, Heidelberg (2005)
11. Michel, O.: Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems* 1(1), 39–42 (2004)
12. Hsu, H., Liu, A.: Multiagent-based multi-team formation control for mobile robots. *Journal of Intelligent and Robotic Systems* 42, 337–360 (2005)
13. Correll, N., Sempo, G., de Lopez, M.Y., Halloy, J., Deneubourg, J.-L., Martinoli, A.: SwisTrack: A Tracking Tool for Multi-Unit Robotic and Biological Systems. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Piscataway, NJ, pp. 2185–2191. IEEE Press, Los Alamitos (2006)
14. Graham, R.L.: An efficient algorithm for determining the convex hull of a finite planar set. *J.-Info.-Proc.-Lett.* 1(4), 132–133 (1972)
15. Viscido, S.V., Parrish, J.K., Grunbaum, D.: The effect of population size and number of influential neighbors on the emergent properties of fish schools. *Ecological Modelling* 183, 347–363 (2005)
16. Pugh, J., Martinoli, A.: Relative Localization and Communication Module for Small-Scale Multi-Robot Systems. In: *IEEE International Conference on Robotics and Automation*, Piscataway, NJ, pp. 188–193. IEEE Press, Los Alamitos (2006)

Applying Reinforcement Learning to Multi-robot Team Coordination*

Yolanda Sanz, Javier de Lope¹, and José Antonio Martín H.²

¹ Perception for Computers and Robots, Universidad Politécnica de Madrid
yolanda.sanz.sanchez@gmail.com, javier.delope@upm.es

² Dep. Sistemas Informáticos y Computación, Universidad Complutense de Madrid
jamartinh@fdi.ucm.es

Abstract. Multi-robot systems are one of the most challenging problems in autonomous robots. Teams of homogeneous or heterogeneous robots must be able to solve complex tasks. Sometimes the tasks have a cooperative basis in which the global objective is shared by all the robots. In other situations, the robots can be different and even contradictory goals, defining a kind of competitive problems. The multi-robot systems domain is a perfect example in which the uncertainty and vagueness in sensor readings and robot odometry must be handled by using techniques which can deal with this kind of imprecise data. In this paper we introduce the use of Reinforcement Learning techniques for solving cooperative problems in teams of homogeneous robots. As an example, the problem of maintaining a mobile robots formation is studied.

Keywords: Multi-robot Systems, Reinforcement Learning, Cooperative Behaviors, Coordination.

1 Introduction

Learning is the process by means of which knowledge, skills, understanding, values and wisdom are acquired. This acquisition is made through study, experience or teaching. There is a consensus about learning being one of the fundamental characteristics of the Intelligence, being human, animal or artificial.

We tend to use solutions that include learning when the problem to be solved is not completely described and the knowledge about the environment is not available during the designing stage. Learning introduces autonomy to the systems that we are creating. Machine Learning discipline studies how these concepts can be included in artificial systems, for instance robotic systems.

Several questions are arisen when we try to apply the learning to Robotics [1]. The first one could be what can robots learn? Robots can learn about themselves. Sensors introduce uncertainty and errors in the control system. Usually we try to define complex sensor models to estimate these errors but the control system can also learn them and adapt itself to be consistent. This can be applied to the actuators, too. Actuators

* This work has been partially funded by the Spanish Ministry of Science and Technology, project: DPI2006-15346-C03-02.

are the second contact between the robot and the world and are also a source of errors. The control system can also learn how actuators work, from simple direct-current electric motors to complex articulated locomotion mechanisms. And also it could be interesting that robots learn about the behaviors that they tend to follow and how the behaviors are activated according to the stimuli present in the environment.

Robots can also learn about the environment. Robots usually make metric or topological environment models which are used for path-planning, localizing goals and landmarks, etc. Sometimes robots learn paths directly rather than a map of their world. For instance, the paths can be learned as sequences of basic movements or behaviors activation in structured environments. Also they can learn where the hazardous zones are and try to avoid them while they are planning the route.

In multi-robot systems, robots can also learn from other robots in the environment. They can acquire simple knowledge as for instance how many robots are in the room but they also can learn the robot type—in heterogeneous multi-robot systems—studying some of their features. Thus, a robot could determine if a specific task can be started in the room or if it should go to other room because there is already another robot with its features.

We could also formulate a second question: Why must robots learn? Learning allows the robot to improve the performance while it carries out its task. This is an important issue due to the robot controllers are not perfect. As we previously stated neither sensors nor actuators are perfects and they generate errors and uncertainty. By means of learning, robot controllers can incorporate that uncertainty into the system and improve its response.

Learning can also help robots to adapt their behaviors to the changes that are produced in the environment or in the task they are performing. For developing controllers that could manage tasks or changes in the environment, the designer must know them and to anticipate. This is not an easy issue in the general case. Furthermore, if the changes are known in prior, we can dispute if the learning is really needed because the changes can be directly included.

Finally, learning can simplify the robot programming. Some tasks are hard to be accomplished and require a considerable effort to be programmed by hand. In such cases, the robot can learn task on its own. Consider the case of fine tuning of thresholds or another parameters in the controller. Under a classical perspective, we must get a model, which usually it is not easy to achieve, to solve some kind of complex, maybe differential, equation system and to obtain the values for the parameters. Once we have adjusted the parameters, the system must be tested to verify if the model was defined correctly and iterate in the process until the desired behavior is achieved.

The last question we are going to formulate is what method can be used for robot learning. The Artificial Intelligence classical literature defines four main subfields in the Machine Learning discipline [2]. Learning by examples that includes topics as inductive learning can be used in robotics. The main difficulty in this case is to obtain the examples to feed the learning system. When we consider autonomous robots that can solve all kind of situations, this type of learning, usually symbolic, is not probably well suited.

Artificial neural networks are also a way to acquire and arrange knowledge. Supervised methods also need examples but, by using unsupervised methods, the robot can create models with the data it is recovering during the mission.

The third subfield is integrated by several statistics methods as for instance the Bayesian reasoning. Several flavors of evolutionary strategies such as the classical genetic algorithms or the evolutionary programming are also considered in this group. They are usually off-line methods which can be used during the designing stage. Evolutionary strategies are incredible optimization methods that can help the designer to adjust the parameters of a robot controller. During the evolving stage these controllers are organized in populations composed by a large number of individuals. These evolved controllers are simulated under restricted conditions but finally they must be tested in the real robot under real conditions. Used in addition to other soft-computing technique, as for instance the artificial neural networks, they can be applied to a lot of different domains.

The reinforcement learning is the last subfield in which the machine learning can be divided. Reinforcement learning is a type of learning that allows the robots to learn from the feedback they receive from the environment. Basically, the reinforcement learning is based on to try several alternatives and check what happened. If it was well, that alternative will be selected again in the future when the same environmental conditions are produced. If it was wrong, that alternative will be avoided. We can use these ideas for building robot controllers that describe very complex behaviors.

2 Reinforcement Learning

Reinforcement Learning (RL) [3] is a paradigm of Machine Learning (ML) in which rewards and punishments guide the learning process. One of the main ideas of RL is learning by a “direct-online” interaction with the environment. In RL there is an agent (learner) which acts autonomously and receives a scalar reward signal which is used to evaluate the consequences of its actions. The framework of RL is designed to guide the learner in maximizing the average reward that it gathers from its environment in the long run.

Let us start with a classical temporal difference (TD) learning rule [4]:

$$Q(s, a)_{t+1} = Q(s, a)_t + \alpha[x_t - Q(s, a)_t], \quad (1)$$

where $x_t = r + \gamma \max_a Q(s_{t+1}, a)_t$.

This basic update rule of TD-learning method can be derived directly from the formula of the expected value of a discrete variable. We know that the expected value (μ) of a discrete random variable with possible values x_1, x_2, \dots, x_n and probabilities represented by the function $p(x_i)$ can be calculated as:

$$\mu = \sum_{i=1}^n x_i p(x_i), \quad (2)$$

thus, for a discrete variable with only two possible values the formula becomes:

$$(1 - \alpha) a + \alpha b, \quad (3)$$

where α is the probability of the second value (b). Then taking a as a previously stored expected value μ and b as a new observation x_t , we have:

$$\mu = (1 - \alpha)\mu + \alpha x_t, \quad (4)$$

and doing some operations we get:

$$\mu = (1 - \alpha)\mu + \alpha x_t = \mu - \alpha\mu + \alpha x_t = \mu + \alpha[-\mu + x_t] = \mu + \alpha[x_t - \mu]. \quad (5)$$

Then we can see that the parameter α refers to the probability that the variable μ get the value of the observation x_t . Thus by replacing μ by $Q(s,a)_t$ and replacing x_t by the received reward plus a fraction of the best expected future reward $r + \gamma \max_a Q(s_{t+1},a)_t$ we finally get again the TD-learning update rule (1).

One of the major breakthroughs in RL was the introduction of an off-policy RL algorithm called Q-learning [4]. The development of an off-policy RL algorithm arose very important consequences from a theoretical point of view due to it was the first time that a formal convergence proof was achieved for a RL algorithm, determining that the Q-learning algorithm reaches an optimal control policy with probability 1 under certain strong assumptions [3, p.148].

Fig. 1 shows the agent-environment interaction cycle in RL. This interaction cycle could be described by four basic steps:

1. Perceive the state s of the environment.
2. Emit an action a to the environment based on s .
3. Observe the new state s' of the environment and the reward r received.
4. Learn from the experience based on s , a , s' and r .

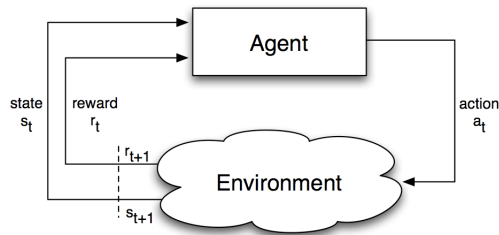


Fig. 1. Agent-Environment interaction cycle in RL

3 Coordinating Multi-robots

Currently, one of the most challenging problems in the autonomous robots field is the control of multi-robots teams for accomplishing a common task [5,6]. Some tasks can be performed more efficiently by a team of robots than a single robot. A well-know example of this improvement is the case of the localization of multiple robots: they can localize themselves faster and more accurately if they exchange their positions when a robot perceives each other [7].

We can discuss some of the basic characteristics of multi-robot systems. These systems are usually composed by a relatively large number of robots. Each robotic unit is a low-cost implementation that solves just a part of the whole problem. Since we are using low-cost robots the system can include redundancy because the adding of more elements is inexpensive. Thus, we get more fault-tolerant than having only one

powerful and expensive robot and also an improved robustness. Obviously these systems present an inherent parallelism, which makes that several tasks can be performed concurrently.

One of the main problems in multi-robot systems is how the agents must be coordinated to solve the assigned task [8]. There exist several alternatives but basically we can consider two complementary cases: a centralized control in which a central system coordinates the actions finally performed by the autonomous agents, and a distributed approach in which each robot is able to solve some questions locally while it try to coordinate itself —implicitly or explicitly— with the other robots.

There are also several alternatives to apply Reinforcement Learning to multi-robots systems [9]. Traditional RL methods or specific RL for multi-agent domains can be used [10,11,12]. As we will describe in the sequel we follow a centralized control approach that is based on a traditional RL method, in this case, Q-learning [4].

4 Learning to Line Up

We use a simulated environment in which the robots are placed in a parallel plane, with an initial situation established as pointed out in Fig. 2 (left). Note that the robots are not aligned, and their orientation is set up by the direction of the front part pointing at the positive x-axis.

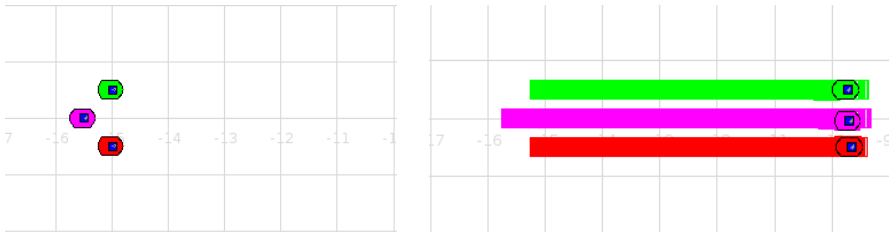


Fig. 2. The robot team in an initial position (right) and the final desired configuration (left)

Taking into account the values obtained by the left side ultrasonic sensors, the robot receives the information needed to know which is its relative position regarding the nearest robot at its left side. Distinguish which is that nearest robot is a problem to consider when using sonars as system of perception, as long as the same patterns are generated from a certain distance. This is the main reason why we are discretizing the sonar readings in order to set which one is the nearest robot to the left.

The above-mentioned reference system shows if the robot is ahead, it is even or it is behind. The robot situated in the lowest position of the y-axis is going to take care of align itself only with the robot located right to its left side. All the robots will apply the corresponding actions and every one of them properly aligned among them as shown in Fig. 2 (right). Nevertheless the robot situated in the upper position of the y-axis is taking no references since, as long as the robot located at its right side is duly aligned with the rest, applying the transitivity property this first robot will be aligned with all the others.

The learning task is divided into a sequence of episodes. Each episode consists of two hundred steps. After those two hundred steps the environment and the robots are reset, being then setting up as in the beginning of the episode. The episode breaks if one of the robots does not fit one of the states to be defined further on.

State Space Definition. States in this problem are not determined taking each robot individually, but rather we consider as an only entity the whole set of positions of the multi-robots system. The state is determined by $n - 1$ digits according to the number of robots, being each one of this digits the relative position of each one of the robots—less one— inside the system.

To specify each robot's relative position in this experiment we use just the four sonars on the left side (remarked in dark grey in Fig. 3), the two laterals, the first upper oblique and the first lower oblique. These sonars will be the ones in charge of designate whether the robots are ahead, even or behind the adjacent robot on its left as stated above. The lateral sonars will activate up to a predefined threshold when the robot is aligned with the nearest one on its left. We do not set a threshold for each lateral one because they act as just a sonar because of its nearness. For its part, the upper oblique will activate when the robot is behind and the lower one when it is ahead.

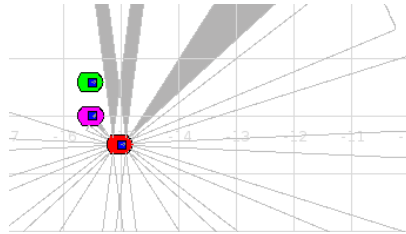


Fig. 3. The sonars used for the nearest robot detection are shown in dark grey in the figure

The set of all robots positions shapes the current state of the multi-robot system. Thus, it can be told that there are three sub-states per robot, which translates into nine states concerning this very particular multi-robot system. However we should add one more state, one not enclosed in those aforementioned, which corresponds with the robot being so ahead or so behind regarding the reference robot. This happens when the values of the robot's sensors exceed the threshold previously established for each situation. When this state is reached the episode breaks, the robots positions are reset and a new episode begins.

Actions. A specific action represents a set of three values to apply to each one of the three robots which made up the multi-robots system. Those values symbolize the speed of each robot. In a more general approach, each system action divides into as many sub-actions as robots has that very system, being these actions applied to each robot separately.

The robots, which as stated above move in a parallel way, move through the environment picking one of the following sub-actions: *move ahead* or *stop*. So we have got two sub-actions for each robot, making it eight possible actions within the multi-robots system. At first we thought about making it three sub-actions per robot, adding to those

stated before *move backwards*, but we noticed that it does not contribute with new information and the global system is up to twenty-seven actions. As a result of this we only get redundant information and the convergence time increases dramatically.

Rewards. We have employed several types of reward functions, though the most efficient was the one described on this paper. Thus, when all the robots reach the objective state once the all are aligned, they get positive reward $+3$. For the rest of the states the reward is negative -1 , except when the robots separate too much of its reference robot, in which case they are punished with a remarkably negative reward.

5 Experimental Results

The behavior of the learning system is described in Fig. 4. Fig. 4(left) shows the cumulative reward obtained per episode. As one can see there is a first stage of learning and exploration where the system is improving its behavior while exploring new strategies in order to line-up the robots. When the system reaches an optimal control policy it can be seen that the behavior is stabilized in the optimal policy obtaining thus the maximal cumulative reward per episode. This result shows that the addressed task is solved in a very short time and that an optimal solution is found.

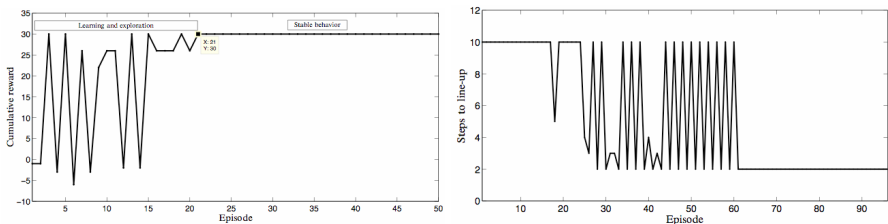


Fig. 4. Reward per episode (left) and Steps to line-up per episode (right)

Furthermore, Fig. 4(right) shows the behavior of another run of the same experiment from the optimal policy point of view. In this figure we can appreciate that the robots learn to line-up in only two steps, which is the minimal possible solution to the problem under our experimental framework. Again it can be seen that there is an initial stage for learning and exploration policy and a final stage when the system reaches a stable configuration and the optimal policy is found.

6 Conclusions and Further Work

The multi-robot systems domain is a perfect example in which the uncertainty and vagueness in sensor readings and robot odometry must be handled by using techniques which can deal with this kind of imprecise data. In this paper we introduce the use of Reinforcement Learning techniques for solving cooperative problems in teams of homogeneous robots. As an example, the problem of maintaining a mobile robots formation is studied. Here we have formally presented the Line-up problem where a

team of robots has to reach a line-up configuration. The experimental results show the viability of the proposed strategy, design and implementation of the Line-up problem.

Further research is guided by the idea of experimenting with physical, real robots in a set of more general and harder coordination problems such as coordinated maneuvers, robot interaction and competition. Currently we are starting to apply artificial neural networks in order to avoid the state space discretization for improving the perception and obtaining thus better controllers.

References

1. Mataric, M.: *The Robotics Primer*. MIT Press, Cambridge (2007)
2. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 2nd edn. Prentice Hall, New Jersey (2002)
3. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press, Cambridge (1998)
4. Watkins, C.J., Dayan, P.: Technical Note Q-learning. *Machine Learning* 8, 279 (1992)
5. Liu, J., Wu, J.: *Multi-agent Robotic Systems*. CRC Press Int., Boca Raton (2001)
6. Resenfeld, A., Kamika, G.A., Kraus, S., Shehory, O.: A study of mechanisms for improving robotic group performance. *Artificial Intelligence* 172(6–7), 633–655 (2008)
7. Fox, D., Burgard, W., Kruppa, H., Thrun, S.: A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots* 8(3), 325–344 (2000)
8. Farinelli, A., Iocchi, L., Nardi, D.: Multi-robot systems: A classification focused on coordination. *IEEE Trans. on Systems, Man, and Cybernetics, Part B* 34(5), 2015–2028 (2004)
9. Yang, E., Gu, D.: *Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey*. Technical Report CSM-404, University of Essex, Department of Computer Science (2004)
10. Mataric, M.: Reinforcement Learning in the Multi-Robot Domain. *Autonomous Robots* 4(1), 73–83 (1997)
11. Zheng, Z., et al.: Multiagent Reinforcement Learning for a Planetary Exploration Multirobot System. In: Shi, Z.-Z., Sadananda, R. (eds.) *PRIMA 2006*. LNCS (LNAI), vol. 4088, pp. 339–350. Springer, Heidelberg (2006)
12. Wang, Y., De Silva, C.W.: Multi-robot Box-pushing: Single-Agent Q-Learning vs. Team Q-Learning. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 3694–3699 (2006)

A Complex Systems Based Tool for Collective Robot Behavior Emergence and Analysis*

Abraham Prieto, Francisco Bellas, Pilar Caamaño, and Richard J. Duro

Integrated Group for Engineering Research,
Universidade da Coruña, Spain
{abprieto, fran, pcsobrino, richard}@udc.es
<http://www.gii.udc.es>

Abstract. This paper presents a Complex Systems Theory based methodology and tool for the automatic design of multiagent or multirobot collective behaviors for the optimized execution of a given task. The main goal of this methodology is the representation of a generic task to be optimally performed in a Complex Systems simulator called WASPBED and the subsequent analysis of the emergent states thus obtained. This way, by tweaking environmental parameters in the system, the behaviors of the different collective behaviors obtained can be studied. The example used to test the methodology deals with collective behaviors for optimized routing in unknown environments.

Keywords: complex systems, behavior emergence, simulation tools.

1 Introduction

Currently Multirobot systems (MRS) are an area of growing importance within the fields of robotics and artificial intelligence. Several multirobot systems for tasks of varying complexity have been presented in the literature (see [1][2] for general reviews on MRS). It is easy to see that the ability of the robots to cooperate and collectively perform tasks is an element of fundamental importance in this type of systems, in particular as the environments and tasks become more complex. Consequently, design methodologies and tools that allow for the study and the optimal design of behavior controllers for MRS are a necessity.

Typical MRS systems are designed by hand, that is, the engineers program ad hoc algorithms for particular hardware platforms. Recently, some authors have started to look into other approaches such as learning systems [3] or evolution [4], basically exporting techniques and methods from single robot systems to the multirobot domain. In general these techniques have allowed a certain amount of research to be carried out and have even provided some successes and ideas on future lines of research. However, one of the main ideas that have resulted from this collective work is that directly exporting single robot approaches to multiple robot systems does not address many of the problems found in this area. For instance, determining the size of

* This work was supported by the MEC of Spain through project CIT-370300-2007-18 DPI2006-15346-C03-01 and DEP2006-56158-C03-02.

the MRS or if it should be homogeneous or heterogeneous in its control. As indicated by Farinelli et al. [1], the proposed approaches need to be precisely characterized in terms of assumptions about the environment and in terms of the internal system organization [5].

In this paper, we present the application of a methodology that is directly designed for obtaining collective multirobot and multiagent behavioral strategies that lead to the optimization of the performance of a given task. This problem is similar to other engineering problems requiring complex optimization procedures and which are suitable for the application of population-based search algorithms to deal with the high dimensionality of the resulting solution spaces. General search techniques usually imply having to spend a lot of time in their adaptation to the particular practical engineering problem at hand in order to reduce the high computational cost derived from the large number of evaluations that are typically necessary and avoiding problems such as getting stuck in suboptimal solutions.

To address this kind of problems in a more efficient manner, we have started to develop techniques that allow us to obtain more efficient algorithms to solve particular engineering problems instead of adapting general algorithms that were not created to deal with this kind of limitations. In this line, inspiration was sought from complex systems theory, in particular from the field of Artificial Life, where different approaches can be found related to the application of complex systems to non-biological practical problems [6]. Particularizing to function optimization, Yang et al [7], have considered distributed algorithms which can be successfully applied to parallel optimization and design. The underlying idea is the same one we want to exploit in this work: complex tasks may be performed by distributed activities over massively parallel systems composed of computationally and/or physically simple elements.

All of these approaches use complex systems in the optimization of a target function. Here, instead of optimizing a particular function, we are more interested in the emergence of a distributed optimized way of performing a task, basically of a distributed algorithm, where the distribution is performed over the behaviour controllers of a set of robots. This idea was originally presented in Langton's Artificial Life book [8] with a chapter devoted to the design of artificial problem solvers.

2 Methodology

Obtaining emergent distributed optimization algorithms for engineering problems requires, on one hand, deciding how this problem can be represented as a complex system, that is, through a set of simple interacting agents. As a second step, it is highly recommended to establish a simulator for its study that must be very flexible in the definition of the capabilities of the agents in order to be able to represent different algorithm topologies. Finally, it is necessary to be able to analyze the main features of the resulting agent population, which may be very complex with large populations.

For the study of virtual complex systems and any other dynamic interactions between the components of simulated environments, a simulation interface called WASPBED (World-Agent Simulation Platform for BEhavior Design) was developed. It provides the capability of changing the environments, the definition of the participating elements or the constraints with the minimum effort and it permits developing

a creation template useful for all the different configurations of the different problem environments. The use of this template guides and simplifies the simulation definition process. In addition, WASPBED is provided with different and very configurable analysis tools. Finally, it permits having full control and freedom to introduce any components, rules, interactions, control systems and so on.

The WASPBED core is composed of four main blocks:

1. **Main manager:** responsible for initialization and linkage of all the modules, control of external inputs, file access, the graphic interface, etc. In order to increase the generalization and simplify defining the environments and their interactions, each "world" is composed of just two types of structures: Elements and Events.
2. **Elements:** they represent anything inside the world that contains a set of characterization parameters. These parameters can be associated to the element or to the type of element and can change or not throughout the simulation run (element state or descriptive parameters).
3. **Events:** they are the representation of the different interactions that guide the simulation of evolution, that is, any action that implies variations in the value of the state parameters is an event.
4. **Configuration templates:** the set of xml files that define the environment. There are three types: creators (initialization parameters of the elements), type definitions (parameter definitions and their default initialization values) and the compatibility board (definition of the associations between elements and events).

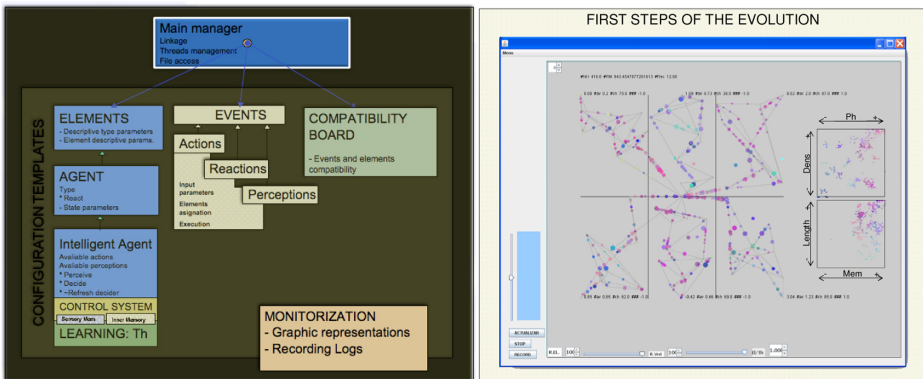


Fig. 1. WASPBED structure and during execution

The WASPBED tool has been programmed in JAVA due to its platform independence capabilities. The computation of the events algorithms is distributed amongst the elements that run as independent process threads. The simulation interface also allows associating a control system to each agent, managed by one or more decision threads. This control system is provided with a sensory apparatus and an internal memory. In addition it contains a learning block that provides means to adapt the control system.

In terms of analysis, the environment is basically endowed with what are called "monitoring agents" and several representation panels within the graphic interface. These agents collect information of single element parameters or complex combinations

of them changing in real time, and represent them in the different panels while the simulation is running. These combinations of parameters, which may include statistics, trends, local trends, etc. are defined by the user in the same way as the events, and represented through bars, graphs, or numerical values. Thus, the analysis tools allow for easy reconfiguration and adaptation to each specific case.

Finally, through the analysis of the control structures obtained for all the elements in the population it will be possible to extract the distributed algorithm for the problem to be solved. In particular, here we will be seeking algorithms having to do with problems that may be represented as graphs (routing problems).

3 Application Example

A practical application example will be used to show the main features of the methodology developed here. As an initial application test, we have chosen a routing problem that consists on sets of robots finding an algorithm that provides the best route to go from the origin to the target on a random graph. Consequently, we have defined an environment made up of the following elements (see Fig. 1):

- **The graphs:** they contain nodes and edges linking nodes. There are two special nodes, the origin and the target. We have situated 6 random graphs with 12 nodes and 25 edges each. Each edge has a modifiable state parameter (pheromone density). The graph has a state parameter representing the amount of resource present in the target node that is decreased by the robots when they arrive there.
- **The robotic agents:** that travel around the edges of the graphs. Their relevant state parameters are their position, age, activation level and an itinerary memory. Their descriptive parameters (inheritable) are four coefficients of the control system and another one controlling pheromone production.

The interaction rules (events) in this case are:

- **Advance:** each agent goes through the current edge towards the next node.
- **Choosing the next edge:** this represents the control system. It decides the next edge using an evaluation function. It is activated when the agent is on a node.
- **Reproduction:** when an agent reaches the origin after having reached the target, that is, after a complete cycle, it performs a crossover process of its inheritable parameters with any other agent in the same situation in order to create a new agent.
- **Graph regeneration:** when one graph's resources run out, new random nodes and edges are generated.
- **Pheromone release:** when an agent finds the target node, it may increase the pheromone density of the edges it goes through on its way back to the origin node, the release rate is regulated by the pheromone production parameter.
- **Itinerary memory update:** Every time an agent leaves an edge, its memory is updated adding this new edge.
- **Agent death:** when the agent's age exceeds its longevity value it is removed from the population.

With these rules we have obtained an environment where the agents go through the different graphs leaving different pheromone trails, choosing their paths and

reproducing depending on their controller parameters. As time progresses, they improve their ability to find better routes between the origin and target nodes due to the association between reproduction and finding the origin and target nodes.

The control system decides the next edge based on an evaluation function that consists on the sum of four product terms: the pheromone level of the evaluated edge times the pheromone density coefficient, the logarithm of the number of agents on the edge times the agent density coefficient, the square root of the position of the edge in the itinerary memory times the memory coefficient and the edge length times the edge length coefficient. All of the terms are normalized between 0 and 1. The best evaluated edge will be followed by that individual. Thus, different coefficients determine different behaviors for the different individuals. An additional parameter is used, the pheromone production rate, which represents the amount of pheromone an ant produces when it leave an edge on its way back to the origin.

As soon as we started with the simulations, we noticed that the initial definition of the rules described above presented a shortcoming: when the population improves, the agents select shorter paths and the resource consumption increases. As a consequence, the individuals tend to perform better but the system can't assure that the resources are available when the target point is reached. As a consequence the system degrades its performance.

This happens mainly because the agents do not have any sensor providing information on the resource levels in a graph and therefore they do not have information on the existence of resources until they reach the target point. In addition, eventually the graphs become too easy to solve, then the extinction of a graph occurs frequently. The main implication of this is that the success of an individual becomes more and more random and less focused on improving efficiency. To solve this problem, we have introduced in the system a predator/prey model based rule that uses a new parameter in the graphs: the advance velocity.

In real models, the average path required to find a resource point is directly related with the prey/resource density and, consequently, when the resources decrease, this density decreases too and the average path increases. As a consequence the complexity to find resources increases accordingly and thus the agent's performance must improve for its survival.

To include such rule in the system, we start by assuming an apparent fixed surface of area (S) (representing the area of the territory in which the system evolves), that together with the available resources level (R_{lev}) will lead us to an apparent resource density (D_{ap})

$$D_{ap} = \frac{R_{lev}}{S}$$

The average path length (L_{av}) to reach a resource will be proportional to a characteristic length and, thus, inversely proportional to the squared root of the apparent resource density:

$$L_{av} = \frac{c_0}{\sqrt{D_{ap}}} = \frac{C}{\sqrt{R_{lev}}}$$

Instead of modifying the graph length as a function of the resource density, we modify the advance velocity so that the average time to cover a graph stretch is the

same as by a modification of edge length. Thus, starting from an initial length (L_0) and an initial advance velocity (v_0), the instantaneous velocity (v_{adv}) is:

$$T_{elapsed} = \frac{L_{av}}{V_0} = \frac{L_0}{v_{adv}} \Rightarrow v_{adv} = \frac{L_0 v_0}{L_{av}} = C' \sqrt{R_{lev}}$$

3.1 Resulting Algorithms

We have allowed the system to evolve with several different configurations changing different parameters of the environment rules. Those we consider the most interesting to analyze are the following:

1. We let the system evolve with the previously described rules and, after a few minutes, the population reaches a stable state. Its genetic parameters are shown in Fig. 2. As shown, the most relevant parameters are the memory coefficient and then the pheromone density, which means that, initially, the agents select, out of the available edges, those that are more separated in its memory. That is, they select those in which they have not been before or where they have been a long time ago. Secondly, the agents select the edges that have the highest pheromone density. The agent density is used in third place with a low and negative value, but it is also of interest since, despite the fact that it is not decisive in most of the edge selections, it helps to spread population about into several graphs at the starting point or after the deactivation of a graph (remember that all the graphs are linked and when an individual is choosing a new graph, each graph is treated as an edge with an equivalent pheromone level and agent density). Finally the edge distance coefficient has a value near zero, nevertheless this parameter is not very important because each of the randomly generated graphs could have an optimum composed by short or long edges indistinctly, in fact, on several evolutions the first three coefficients preserve the same values and this one varies.

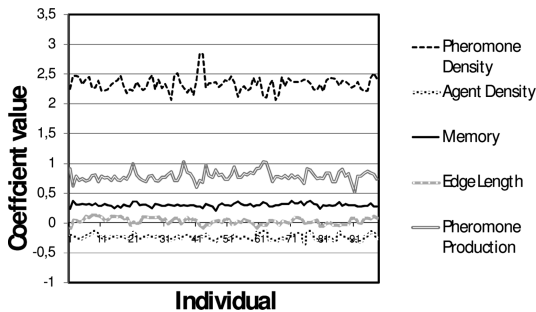


Fig. 2. Resulting algorithm parameters for the whole population using the original rules

2. In a second case we let the system evolve without updating the pheromone values of the edges, that is, the communication through the environment (stigmergy) the agents were using is no longer possible. With this modification, we obtain the population parameters shown in Fig. 3. The behavior is now guided by both the memory and the agent density. The individuals in this case behave in a similar

way as in evolutions with pheromone updating because they are choosing the most populated edges and thus most of them are following the same paths. They are in fact using another way of indirectly communicating information and taking advantage of it. The other coefficients are not relevant: the pheromone coefficient has no effect, the pheromone value of the edges is now always zero, and the edge length is not related with the quality of the solution, as explained before.

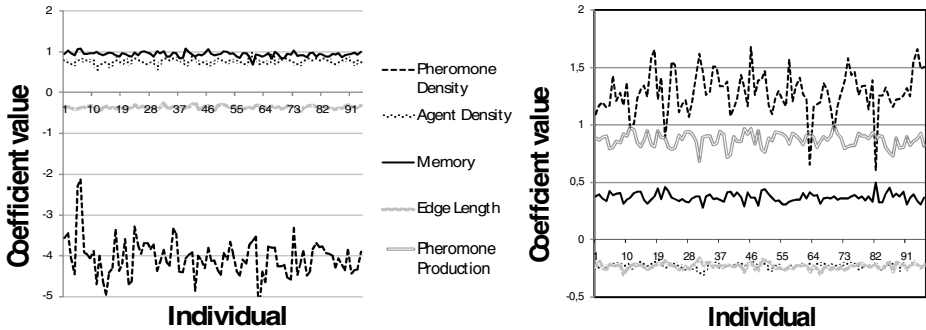


Fig. 3. Resulting algorithm parameters for the whole population without updating the pheromone values of the edges (left) and with a maximum lifetime for the graphs (right)

3. In the last configuration we have tried to make the problem a little bit harder. A maximum lifetime has been defined for the graphs, and when consumed the graphs are deactivated. So now graphs are deactivated when they run out of resources or when they reach their time limit. Pheromone updating is allowed in this case. The first consequence of this rule is that there is almost no time to exploit a graph after finding the good paths. Thus, basically, the agents are most of the time exploring new graphs and finding solutions as fast as possible. The coefficients obtained are shown in Fig. 4 and correspond to a behavior that is not very different from the first one we analyzed except for one aspect: the pheromone coefficient is now twice as high as in the first case. The memory coefficient and the pheromone production levels are the same. This means that the agents assign a higher relevance to the pheromone trail. This can be considered normal as now there is no time for the pheromones level to increase too much.

Thus, to conclude this experiment, we must point out that after enough iterations, the simulations converge to a dominant type of agents with very similar coefficients (similar controller and behavior), although there are still agents that have different behaviors from the general population. It is important to note that memory is fundamental, as we have seen in all the simulations with several configurations, otherwise the individuals are easily trapped in loops around the same nodes or edges. In fact, after the first time steps all those individuals that do not use their memory disappear.

Our efficiency measure is the average number of steps needed to travel between the origin and target which, despite fluctuations due to changes in the environmental conditions, improves with evolution.

4 Conclusions

Trying to obtain algorithmic solutions from the analysis of populations that have reached emergent stable behaviors in adapted environments seems to be a promising avenue of work, especially for very complex problems with dynamic interactions which is the case of Multirobot Systems. For dealing with this kind of simulations, it is necessary to adjust a huge number of parameters which can influence the stability and efficiency of evolution. Consequently, developing and studying analysis tools to, on one hand understand and be able to guide our population and on the other hand to extract solutions from the results, becomes very important and profitable.

We are now working on improving the control tools, finding more relevant parameters to characterize and guide the evolution, making the methodology more general to be able to adapt any problem to an evolutionary dynamic complex system.

References

1. Farinelli, A., Iocchi, L., Nardi, D.: Multirobot systems: A classification focused on coordination. *IEEE Trans. Syst., Man, Cybern. B* 34(5), 2015 (2004)
2. Gerkey, B.P., Mataric, M.J.: A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research* 23(9), 939–954 (2004)
3. Yang, E., Gu, D.: A Survey on Multiagent Reinforcement Learning Towards Multi-Robot Systems. In: *Proceedings of IEEE Symposium on Computational Intelligence and Games* (2005)
4. Baldassarre, G., Nolfi, S., Parisi, D.: Evolving mobile robots able to display collective behaviour. *Artificial Life* 9, 255–267 (2003)
5. Mataric, M.J.: Interaction and intelligent behavior. Ph.D. dissertation, Dept. Elec. Eng. Comput. Sci., Mass. Inst. Technol., Cambridge (1994)
6. Sung-BaeCho, K.-J.K.: A Comprehensive Overview of the Applications of Artificial Life. *Artificial Life* 12, 153–182 (2006)
7. Yang, B.S., Lee, Y.H.: Artificial Life Algorithm for Function Optimization. In: *Proceedings 2000 ASME Design Engineering Technical Conferences and Computer and Information in Engineering* (2000)
8. Langton, C.G.: *Artificial Life: an overview*. MIT Press, Cambridge (1995)

On the Need of Hybrid Intelligent Systems in Modular and Multi Robotics^{*}

R.J. Duro¹, M. Graña², and J. de Lope³

¹ Grupo Integrado de Ingeniería, Universidade da Coruña, Spain,

² Universidad del País Vasco, Spain

³ Percepción Computacional y Robótica, Universidad Politécnica de Madrid, Spain
richard@udc.es, ccpgrrrom@si.ehu.es, javier.delope@upm.es

Abstract. The area of cognitive or intelligent robotics is moving from the single robot control and behavior problem to that of controlling multiple robots operating together and even collaborating in dynamic and unstructured environments. This paper introduces the topic and provides a general overview of the current state of the field of modular and multi robotics taking both of these subareas as different representations of the same problem: how to coordinate multiple elements in order to perform useful tasks. The review shows where Hybrid Intelligent Systems could provide key contributions to the advancement of the field.

Keywords: Intelligent robotics, multi-robot systems, modular robotics.

1 Introduction

The classical concept of general purpose industrial robot, both in the case of manipulators and mobile robots, makes sense when the task to be carried out takes place in static or controlled completely structured settings. However, when the environments are highly dynamic and unstructured and when the tasks to be performed are seldom carried out in the same exact way, it is necessary to make use of robotic systems that require additional properties depending on the task. Examples of these environments are shipyards, plants for constructing unique or very large structures, etc. In these environments, work is not generally carried out as in traditional automated plants, but rather, a series of individuals or groups of specialists perform the tasks over the structure itself in an ad hoc manner. Consequently, it is necessary to seek new approaches that permit automating processes in this type of environments based on design specifications such as modularity, scalability, fault tolerance, ease of reconfiguration, low fabrication and maintenance costs and adaptation capabilities.

Thus, structures that can adapt their hardware and capabilities in a simple manner to the task in hand are sought. At the same time these structures, as they are designed for operating in dynamic environments, must be endowed with capabilities that allow them to adapt to their environment in real time. Obviously, they must continue to operate even when failures occur in some of their components, that is, they must degrade in a

^{*} This work was funded by the Spanish MEC through project DPI2006-15346-C03.

non catastrophic way. All of these requirements imply the construction of a modular architecture, an standardization of the interfaces between modules and an appropriate organization of perception, processing and control for these types of structures that implies the reconfiguration of the system in an intelligent manner for the completion of the mission.

Hybrid intelligent systems are characterized by the composition of the different available computational tools (Bayesian reasoning, neural networks, fuzzy systems, statistical classifiers, evolutionary algorithms, etc.) in a way that is adapted to the particular problem to be solved. They are aimed at achieving the highest degrees of flexibility and adaptation. Until now, most multirobot or modular robotic systems are characterized by the simplicity of their control systems, which are often handcrafted for the particular task that must be demonstrated. We believe there is an open wide application field for hybrid approaches to this type of systems.

2 Approaches to Modular Robotics and Multi-robot Systems

Three different approaches to modularity and multirobotics may be found:

1. Modularity of a continuous robot. It deals with the creation of modules, usually homogeneous, that through their connection into different configurations permit the creation of a single and physically continuous robot. This is the idea of the so called polybots.
2. Modularity of a distributed robot. In this case, the designers seek modules, generally inhomogeneous, and where each one of them is specifically designed for a given part of the global task. They may or may not be physically connected.
3. Distributed cooperative robotic systems. This concept arises from traditional robotics where the cooperation of independent robotic units is sought for performing joint tasks. Their differential characteristic is that the robotic unit is independent and does not require the others in order to perform the task. Cooperation allows them to carry out tasks of a larger scope or complexity.

From now on, and for the sake of simplicity, we will only consider two categories. The first one are systems that present modular solutions, whether continuous or distributed. The second one are systems that encompass several independent robots.

2.1 Modular Robotics

In general, modular robots can be taken as robotic structures that are made up of multiple, generally identical, modules. The underlying idea of modular robotics takes inspiration mainly from cellular automata and social insect theories. Through cooperation, social insects such as ants, bees or termites, perform tasks that would be impossible for a single individual. This way, modular robots use the emergence and holistic principles that are quite popular in current science. This principles state that there are behaviors or properties of the systems that are intrinsic to the whole system and cannot be found in any of its constituents, that is, they emerge from the interaction of the system parts. Using similar principles, modular robots autonomously self-organize and change their

shape in order to adapt to different tasks or classes of terrain. For instance, some modular robots may transform into snakes in order to follow a tunnel and then may transform into quadrupeds to go up stairs. Another important feature of modular robots is their potential for self repair. As the modules making a unit up are usually identical, it is possible to eliminate the damaged module and substitute it using another one, if available. Notwithstanding these comments, most developments in modular robotics are not classified as self organizing systems. This is due to the fact that in most cases, one module is used as the general coordinator and the rest as slaves and these roles can be interchanged.

Even though this research area has progressed very fast since its beginnings in the early nineties, modular robots have achieved their objectives only in very controlled laboratory environments. In reality, the control systems within the modules are created ad hoc for the task to be carried out. There is hardly any instance of this type of systems that presents the capability of autonomously deciding the configuration change or the necessary adaptation parameters. The introduction of advanced (hybrid) intelligent systems may be the key for achieving these properties.

Modular robots present structural degrees of freedom in order to adapt to particular tasks. One of the first implementations is a sewer inspection robot [1] although the idea of reconfigurable modular robots starts with the designs by Yim [2-6] of a polypod robot that is capable of adapting its structure in order to produce different gaits for moving over different terrains. In [7] this philosophy is applied to the design of flexible fabrication cells. A robot for operating in vertical surfaces is presented in [8]. The work in [9] discusses the limitations of metamorphic robots based on cubic modules. Different modular configurations are being proposed even nowadays, examples are [10-14]. New classes of robots are introduced in [15] where Campbell et al. present robots that are configured as power buses while performing the assigned task. In [16] Carrino and col. present modules for the construction of feed deposition heads in the generation of composite materials.

The idea that robots should be able to self-configure is introduced in [17-20] over one type of modular robot called M-TRAN [21,22]. In its current state the transitions between configurations are carried out manually and a few configurations for particular tasks are obtained through genetic algorithms and other global random search methods. The idea of self-configuration over Yim's polybots is proposed in [23]. Liping et al. [24] propose a coupling method that is based on position sensors and in [25] Patterson and col. propose another one based on magnetic couplings that may lead to new proposals of self-configurable robots. An area of active interest is that of the application of intelligent (hybrid) systems for the autonomous on line reconfiguration of this type of robots. It would really be necessary to reformulate the problem as a distributed optimization problem with partial information and combine estimation methods (Bayesian or neuronal) with robust optimization methods (evolutionary or graduated convexity).

Papers [26-29] study kinematic calibration methods and ways for obtaining the inverse kinematics and the dynamics of modular and reconfigurable robots in order to solve the problems introduced by tolerances in the fabrication of the modules. On the other hand, [30] presents a methodology for the dynamic modeling of multirobot systems that facilitates the construction of simulators that may be used in order to accelerate the development of intelligent control systems through virtual experiments.

2.2 Automatic Design

Regarding the automated design of modular robots, some work has been carried out in the application of evolutionary algorithms that seek the minimization of a criterion based on the variety of the modules employed for a given task that is cinematically characterized [31] or on the mass, ability and workspace [32]. In [33] Zhang and col. provide a representation of the robot and the environment that permits the application of case based reasoning techniques to the design of a modular robot. For the automation of the design of the configurations of modular robots, including self-reconfigurable robots, [34] proposes a representation of the potential connection topologies among the modules. Saidani [35] discusses the use of graph theory and cellular automata as a base for the development of design and reconfiguration algorithms. However, this type of systems will not be really autonomous until these design and analysis tasks are carried out in an autonomous and distributed manner over the robot modules themselves. Again, robust estimation and modeling methods that are still not in general use are required.

An aspect we are interested in is that of the need to organize the sensing of the robot so that the different sensors are integrated in order to obtain the desired information. A primitive example is the application of Bayesian decision theory for door detection as presented in [36]. A decentralized Bayesian decision algorithm that may be used for the fusion of sensorial information in sensor networks is introduced in [37]. This algorithm indicates the path to follow for the application of hybrid intelligent techniques to this problem.

2.3 Multi-robot Systems

The second category of interest are robot swarms [38,39]: groups of robots that collaborate to achieve an objective, for example, rescue tasks [40], material handling in flexible fabrication cells [41]. Possibly, the RoboCup robotic football championship is the most important concept testing ground in this field.

The biological foundations of the idea of robot swarms are reviewed in [42], including a prospective of their application in [43]. Different studies of complexity have been carried out over these types of systems. They include the characterization of chaotic behaviors [44]. Dynamic studies have also performed out over simple models such as foraging [45, 46]. The negotiation method [47] proposed for self-reconfigurable robots could also be applied to swarms. On another tack [48] shows how a swarm may be converted into a self-reconfigurable robot. In [49] Fukuda and col. discuss the advantages and disadvantages of multiagent robotic systems as compared to single robots. The individuals considered are in general very simple in their internal dynamics and, consequently, the introduction of sophisticated approximate reasoning systems would permit an extension of the range of behaviors and their robustness to changing situations.

One of the critical aspects of this type of systems is the communication between the members of the swarm [50]. It is usually carried out using radio-links. In [51] Dumbar and Esposito study the problem of maintaining communications among the robots performing tasks. Dongtang et al. [52] study the need of optical communications and evaluate a system based on photo sensors and laser. In [42, 53-55] different authors use the pheromone metaphor as a means of communications in, among other, applications

for the detection of damaged components. Another line of research where hybrid intelligent systems are becoming necessary is that of stigmergy based communication, that is, communication through the environment [56].

The production of consistent information on the environment in a distributed manner is another challenge for this type of system. In [57] Kumar and Sahin consider this problem in the realm of detecting mines. Pack and Mullings [58] introduce metrics so as to measure the success of a joint search performed by a swarm as well as a universal search algorithm. More elaborate representation methods that include training algorithms adapting the agents to their environment and tasks are needed.

Obtaining decentralized control that provides interesting collective behaviors is a central problem [39, 47, 59-65]. In [62] Peleg presents a universal architecture for the decentralized control of groups of robots. A review of the state of the art of decentralized control is given in [63]. Wessnitzer and Melhuish [66] integrate behavior based control strategies with swarm control systems in a task having to do with the elimination of underwater mines. In general, the formulation of decentralized control implies the need to work with incomplete or temporally inconsistent information. Hybrid intelligent systems should help to improve the robustness of these control systems.

Marco Dorigo's group has been very active in this field, developing the idea of swarm-bot: In [67] they discuss a transportation behavior that is similar to that of ants. In [68] they present a hole avoiding behavior. A Review of their work can be found in [64]. Finally, in [69] they study the application of evolutionary techniques for obtaining distributed control methods. Again, these evolutionary techniques are run based on the global information available about the system and are not implemented on the agents. Thus, these systems are still far from being autonomous

An interesting application is that of positioning and map generation through robot swarms [70]. A precedent may be found in [71], and in [72] Di Marco et al. consider the simultaneous localization and map generation problem (SLAM) for a robot team. On the other hand, Stroupe and Balch [73] try to estimate the best next move of a group of robots in order to obtain the map. The techniques are based on variations of Kalman filters, which can clearly be improved through the application of techniques from the hybrid intelligent systems tool box and thus things such as inverting the observation prediction functions could be avoided.

3 Conclusions

The two most important approaches to the construction of multi or modular robots have been reviewed. As a general comment it must be pointed out that this type of systems still lack the desirable level of autonomy. The application of hybrid intelligent systems for the construction of truly autonomous control systems or for the interpretation of the sensing data are open fields of great potential. A systematic need of working with imprecise and incomplete information that may be temporally inconsistent is detected when contemplating distributed implementations of control and sensing. Sensor fusion, whether from several sensors from the same robot or from different robots, requires robust and efficient modeling techniques. It is with the hybridization of different approaches that this may be achieved.

References

1. Cordes, S., et al.: Autonomous sewer inspection with a wheeled, multiarticulated robot. *Robotics and Autonomous Systems* 21(1), 123–135 (1997)
2. Yim, M.: Locomotion With A Unit-Modular Reconfigurable Robot in Department of Computer Science. Stanford University (1995)
3. Yim, M., Duff, D.G., Roufas, K.D.: PolyBot: a modular reconfigurable robot (2000)
4. Yim, M., Duff, D.G., Roufas, K.D.: Walk on the wild side [modular robot motion]. *Robotics & Automation Magazine, IEEE* 9(4), 49–53 (2002)
5. Yim, M., Ying, Z., Duff, D.: Modular robots. *Spectrum, IEEE* 39(2), 30–34 (2002)
6. Ying, Z., et al.: Scalable and reconfigurable configurations and locomotion gaits for chain-type modular reconfigurable robots (2003)
7. Chen, I.M.: Rapid response manufacturing through a rapidly reconfigurable robotic work-cell. *Robotics and Computer-Integrated Manufacturing* 17(3), 199–213 (2001)
8. Sack, M., et al.: Intelligent control of modular kinematics - the robot platform STRIUS (2002)
9. Vassilvitskii, S., et al.: On the general reconfiguration problem for expanding cube style modular robots (2002)
10. Castano, A., Behar, A., Will, P.M.: The Conro modules for reconfigurable robots. *IEEE/ASME Transactions on Mechatronics* 7(4), 403–409 (2002)
11. Golovinsky, A., et al.: PolyBot and PolyKinetic/spl trade/ System: a modular robotic platform for education (2004)
12. Hafez, M., Lichter, M.D., Dubowsky, S.: Optimized binary modular reconfigurable robotic devices. *IEEE/ASME Transactions on Mechatronics* 8(1), 18–25 (2003)
13. Karbasi, H., Huissoon, J.P., Khajepour, A.: Uni-drive modular robots: theory, design, and experiments. *Mechanism and Machine Theory* 39(2), 183–200 (2004)
14. Tokashiki, H., et al.: Development of a transformable mobile robot composed of homogeneous gear-type units (2003)
15. Campbell, J., Pillai, P., Goldstein, S.C.: The Robot is the Tether: Active, Adaptive Power Routing for Modular Robots With Unary Inter-robot Connectors (2005)
16. Carrino, L., Polini, W., Sorrentino, L.: Modular structure of a new feed-deposition head for a robotized filament winding cell. *Composites Science and Technology* 63(15), 2255–2263 (2003)
17. Murata, S., et al.: Concept of self-reconfigurable modular robotic system. *Artificial Intelligence in Engineering* 15(4), 383–387 (2001)
18. Kurokawa, H., et al.: Self-reconfigurable M-TRAN structures and walker generation. *Robotics and Autonomous Systems* (2005)
19. Yoshida, E., et al.: Self-reconfigurable modular robots -hardware and software development in AIST (2003)
20. Terada, Y., Murata, S.: Automatic assembly system for a large-scale modular structure - hardware design of module and assembler robot (2004)
21. Kurokawa, H., et al.: Self-reconfigurable modular robot (M-TRAN) and its motion design (2002)
22. Murata, S., et al.: M-TRAN: self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics* 7(4), 431–441 (2002)
23. Yim, M., et al.: Connecting and disconnecting for chain self-reconfiguration with PolyBot. *IEEE/ASME Transactions on Mechatronics* 7(4), 442–451 (2002)
24. Liping, Z., et al.: Position-sensing based a new docking system of RPRS (2004)
25. Patterson, S.A., Knowles Jr., K.A., Bishop, B.E.: Toward magnetically-coupled reconfigurable modular robots (2004)
26. Yang, G., Chen, I.M.: A novel kinematic calibration algorithm for reconfigurable robotic systems (1997)

27. Chen, I.M., Guilin, Y.: Inverse kinematics for modular reconfigurable robots (1998)
28. Yangiong, F., Xifang, Z., Xu, W.L.: Kinematics and dynamics of reconfigurable modular robots (1998)
29. Seong-Ho, K., Pryor, M.W., Tesar, D.: Kinematic model and metrology system for modular robot calibration (2004)
30. Bonaventura, C.S., Jablokow, K.W.: A modular approach to the dynamics of complex multirobot systems. *IEEE Transactions on Robotics and Automation* 21(1), 26–37 (2005)
31. Yang, G., Chen, I.M.: Task-based optimization of modular robot configurations: minimized degree-of-freedom approach. *Mechanism and Machine Theory* 35(4), 517–540 (2000)
32. Lemay, J., Notash, L.: Configuration engine for architecture planning of modular parallel robots. *Mechanism and Machine Theory* 39(1), 101–117 (2004)
33. Zhang, W.J., Liu, S.N., Li, Q.: Data/knowledge representation of modular robot and its working environment. *Robotics and Comp.-Integrated Manuf.* 16(2-3), 143–159 (2000)
34. Ko, A., Lau, T.L., Lau, H.Y.K.: Topological representation and analysis method for multiport and multi-orientation docking modular robots (2004)
35. Saidani, S.: Self-reconfigurable robots topodynamic (2004)
36. Kristensen, S.: Sensor planning with Bayesian decision theory. *Robotics and Autonomous Systems* 19(3-4), 273–286 (1997)
37. Makarenko, A., Durrant-Whyte, H.: Decentralized Bayesian algorithms for active sensor networks. *Information Fusion* (in press, 2005) (Corrected Proof)
38. Dudek, G., et al.: A taxonomy for swarm robots. In: *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems 1993, IROS 1993* (1993)
39. Sugawara, K., Watanabe, T.: Swarming robots - collective behavior of interacting robots. In: *SICE 2002. Proceedings of the 41st SICE Annual Conference* (2002)
40. Stormont, D.P., et al.: Building better swarms through competition: lessons learned from the AAAI/RoboCup Rescue Robot competition (2003)
41. Doty, K.L., Van Aken, R.E.: Swarm robot materials handling paradigm for a manufacturing workcell. In: *1993 IEEE International Conference on in Robotics and Automation, 1993. Proceedings* (1993)
42. Szu, H., et al.: Collective and distributive swarm intelligence: evolutionary biological survey. *International Congress Series*, vol. 1269, pp. 46–49 (2004)
43. Sahin, E.: *Swarm Robotics: From Sources of Inspiration to Domains of Application. Lecture Notes in Computer Science*, pp. 10–20 (2005)
44. Johnson, J., Sugisaka, M.: Complexity science for the design of swarm robot control systems. In: *26th Annual Conf. Industrial Electronics Society, IECON 2000. IEEE, Los Alamitos* (2000)
45. Sugawara, K., Watanabe, T.: Swarming robots-foraging behavior of simple multirobot system. In: *IEEE/RSJ Int. Conference on Intelligent Robots and System, 2002* (2002)
46. Sugawara, K., Sano, M.: Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica D: Nonlinear Phenomena* 100(3-4), 343–354 (1997)
47. Salemi, B., Will, P., Shen, W.M.: Distributed task negotiation in self-reconfigurable robots. In: *Intelligent Robots and Systems, 2003 (IROS 2003). Proceedings* (2003)
48. Sahin, E., et al.: SWARM-BOT: pattern formation in a swarm of self-assembling mobile robots. In: *IEEE International Conference on Systems, Man and Cybernetics, 2002* (2002)
49. Fukuda, T., Takagawa, I., Hasegawa, Y.: From intelligent robot to multi-agent robotic system. In: *Integration of Knowledge Intensive Multi-Agent Systems* (2003)
50. Stilwell, D.J., Bishop, B.E.: A framework for decentralized control of autonomous vehicles. In: *Robotics and Automation, 2000. Proceedings. ICRA 2000* (2000)
51. Dunbar, T.W., Esposito, J.M.: Artificial potential field controllers for robust communications in a network of swarm robots. In: *Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory, 2005, SSST 2005* (2005)

52. Dongtang, M., Jibo, W., Zhaowen, Z.: A novel optical signal detecting and processing method for swarm robot vision system. In: 2003 Proceedings of IEEE International Conference on Robotics, Intelligent Systems and Signal Processing (2003)
53. Purnamadaja, A.H., Russell, R.A.: Pheromone communication: implementation of necrophoric bee behaviour in a robot swarm. In: 2004 IEEE Conference on Robotics, Automation and Mechatronics (2004)
54. Payton, D., Estkowsky, R., Howard, M.: Compound behaviors in pheromone robotics. *Robotics and Autonomous Systems* 44(3-4), 229–240 (2003)
55. Payton, D., Estkowsky, R., Howard, M.: Pheromone Robotics and the Logic of Virtual Pheromones. *Lecture Notes in Computer Science*, pp. 45–57 (2005)
56. Caamaño, P., Becerra, J.A., Duro, R.J., Bellas, F.: Incremental Evolution of Stigmergy-Based Multi Robot Controllers Through Utility Functions. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) *KES 2007, Part II. LNCS (LNAI)*, vol. 4693, pp. 1187–1195. Springer, Heidelberg (2007)
57. Kumar, V., Sahin, F.: Cognitive maps in swarm robots for the mine detection application. In: *IEEE Int. Conference on Systems, Man and Cybernetics* (2003)
58. Pack, D.J., Mullins, B.E.: Toward finding an universal search algorithm for swarm robots. In: *Intelligent Robots and Systems (IROS 2003)* (2003)
59. Cassinis, R., et al.: Strategies for navigation of robot swarms to be used in landmines detection. In: *Third European Workshop on Advanced Mobile Robots, 1999 (Eurobot 1999)* (1999)
60. Pack, D.J., Mullins, B.E.: Toward finding an universal search algorithm for swarm robots. In: *2003 Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (2003)
61. Dahm, I., et al.: Decentral control of a robot-swarm. In: *Proceedings of Autonomous Decentralized Systems, ISADS 2005* (2005)
62. Peleg, D.: *Distributed Coordination Algorithms for Mobile Robot Swarms: New Directions and Challenges*. *Lecture Notes in Computer Science*, pp. 1-12 (2005)
63. Shen, W.-M., et al.: Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms. *Autonomous Robots* 17(1), 93–105 (2004)
64. Mondada, F., et al.: Swarm-Bot: A New Distributed Robotic Concept. *Autonomous Robots* 17(2 - 3), 193–221 (2004)
65. Helwig, S., Haubelt, C., Teich, J.: Modeling and Analysis of Indirect Communication in Particle Swarm Optimization. In: *The 2005 IEEE Congress on Evolutionary Computation* (2005)
66. Wessnitzer, J., Melhuish, C.: Collective Decision-Making and Behaviour Transitions. In: *Distributed Ad Hoc Wireless Networks of Mobile Robots: Target-Hunting. LNCS*, pp. 893–902 (2003)
67. Trianni, V., Nolfi, S., Dorigo, M.: Cooperative hole avoidance in a swarm-bot. *Robotics and Autonomous Systems* (in press) (corrected proof)
68. Trianni, V., Labella, T.H., Dorigo, M.: Evolution of Direct Communication for a Swarm-bot Performing Hole Avoidance. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) *ANTS 2004. LNCS*, vol. 3172, pp. 130–141. Springer, Heidelberg (2004)
69. Dorigo, M., et al.: Evolving Self-Organizing Behaviors for a Swarm-Bot. *Autonomous Robots* 17(2 - 3), 223–245 (2004)
70. Rothermich, J.A., Ecemis, I., Gaudiano, P.: Distributed Localization and Mapping with a Robotic Swarm. *LNCS*, pp. 58–69 (2005)
71. Cohen, W.W.: Adaptive mapping and navigation by teams of simple robots. *Robotics and Autonomous Systems* 18(4), 411–434 (1996)
72. Di Marco, M., et al.: Simultaneous localization and map building for a team of cooperating robots: a set membership approach. *IEEE Trans. Rob. and Aut.* 19(2), 238–249 (2003)
73. Stroupe, A.W., Balch, T.: Value-based action selection for observation with robot teams using probabilistic techniques. *Robotics and Autonomous Systems* 50(2-3), 85–97 (2005)

Modelling of Modular Robot Configurations Using Graph Theory

José Baca, Ariadna Yerpes, Manuel Ferre, Juan A. Escalera, and Rafael Aracil

Universidad Politécnica de Madrid,
Jose Gutierrez Abascal, 2, 28006 Madrid, Spain
{jbbaca,ayerpes,jescalera,aracil}@etsii.upm.es, m.ferre@upm.es

Abstract. Modular robots are systems that can change its geometry or configuration when connecting more modules or when rearranging them in a different manner to perform a variety of tasks. Graph theory can be used to describe modular robots configurations, hence the possibility to determine the flexibility of the robot to move from one point to another. When the robot's configurations are represented in a mathematical way, forward kinematics can be obtained.

Keywords: Modular Robots, Graph Theory, Configurations for Displacement.

1 Introduction

Now in days, different modular robots designs have been proposed to give a solution to varied fields like versatility, adaptability, robustness, costs, etc. It is possible to find module robot designs from 1 degree of freedom (DOF) [1], 2 DOF [2], [3] and 3 DOF [4], [5]. Each system shows diverse relevant advantages between them, a few of them are reconfigurable and others are autoconfigurable systems, but something in common is that they can form new robot structures when combining modules. When attempting to obtain the model of a defined robot is just a matter of following a systematic procedure. With a defined robot, the number of degrees of freedom, length's links, masses and geometry are normally well defined and constant, facilitating its modeling. On the contrary, modular robots are more complex to model, due to its capability of increasing the DOF, and links when adding modules to the structure. The kinematics of the robot will be defined by the number of modules and the way they are connected. Therefore it is necessary to use a method to easily represent the structure's configuration.

2 RobMAT Architecture

The desire of any modular system is that the whole is greater than the sum of its parts. The main advantage of using modular robots lies in the possibility of constructing different kinematics chains by combining modules.

RobMAT's system architecture is divided into modules, molecules and colonies. A module (figure 1 (a)) is the base component of the system and it has movement and

communication capacity. A **molecule** (figure 1 (b)) is an autonomous entity made out of n-modules connected together ($n > 1$). The molecule holds higher capabilities of movement, communication and manipulation than the module. Various molecules cooperating in the area to fulfill a task is defined as **colony**.

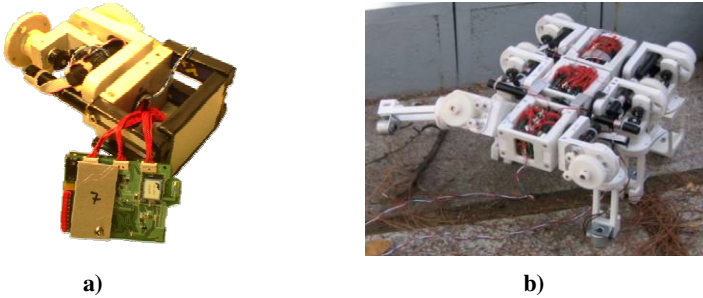


Fig. 1. a) Base component of the RobMAT system. b) Six module configuration.

The system has three communication channels in order to properly transmit commands to the whole system and to be able to work in a cooperative mode. These communication channels are the Intra-Module Communication, the Inter-Modules Communication and the Video Communication.

The first type is in charge of the communication between modules belonging to the same molecule and it uses a CAN bus [6]. The Inter-Modules Communication is in charge of communication between molecules or between a molecule and the control station and it is based on Bluetooth. Finally, the Video Communication is used to send radio frequency video signals from the two mini cameras that can be attached to one of the molecules and that would be used to control the system in a remote working environment using teleoperation.

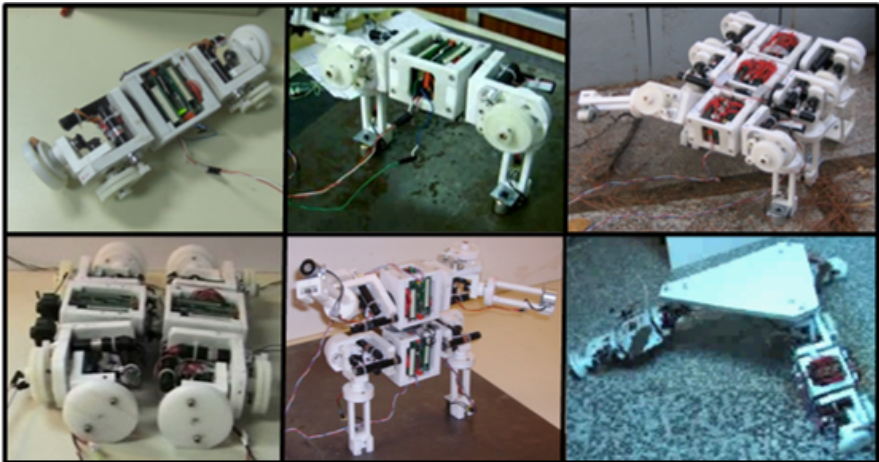


Fig. 2. By combining modules, distinct modular robot configurations can be formed, creating many ways of displacement and manipulation

The used module had an actuator with 3 rotational degrees of freedom with its axes intersecting at one point creating a wrist-like joint. The design of the module allows accessories to be attached on one end of the module and the connector used to enable reconfigurations is made out of an electromagnet (male connector) and a metallic plate (female connector), each part embedded in one of the molecules making possible to form different modular robot configurations as is shown in figure 2.

3 Describing Modular Robot Configurations

The flexibility of modular robots provides enormous adaptation capacity and the ability to perform a number of tasks. However, it complicates the robot's kinematics and dynamic modeling [7]. The changing configuration of molecules means that, unlike other robots, modeling in advance is not possible. An algorithm is therefore required to automatically generate the model for any molecule configuration during the execution of each step of a task.

3.1 Graph Theory

Kinematics graphs are a widely-used tool for the theoretical analysis of mechanisms [8]. These graphs entail vertices and edges (or lines), which represent robot links and joints respectively. All elements of a mechanism belong to one of these categories, so any mechanism can be represented by a graph. Graph Theory [9] offers many theoretical tools to analyze graphs, and applied to mechanisms, specifically talking about the graphs of mechanisms; it provides the opportunity to analyze them in a more abstract manner. A graph is a mathematical object that captures the notion of connection.

3.2 Graph Theory into Modular Robots

A modular system can be represented by its corresponding graph, showing the relationship existing among different modules. The application of Graph Theory to modular robots requires the use of specific graph elements. This degree of specialization is achieved by labeling graph vertices and edges in such a way that there are as many labels as different types of robot links and joints. If we consider RobMAT as a homogenous robot, i.e. without considering the tools it can handle, it can be seen in general one type of **link** (each prism (p) next to the spherical joint), two types of **joint** (connector (c) and spherical (s) joint), and another element to be taken into account when representing module chains is the linking point between modules, which is called the **port**. Basically, a connector can have more than one place or port to join with other connector. Generally, a number codifies each port and is located in the graph close to the corresponding element where the linking is produced. The figure 3 (a) shows representative graph elements over each robot module. Each module is composed of two links, one joint and its ports (e.g. module one is composed of L_1 , L_2 , and J_1 and module two is composed of L_3 , L_4 and J_3), but when combining modules a joint between them is created due to the port at the link section (e.g. J_2 is created when both modules are connected at port 4 in both cases). The figure 3 (b) shows a four module configuration graph and it can be noticed the joints created at each module union (i.e. J_2 , J_4 , J_6 , and J_7) and its ports are showed.

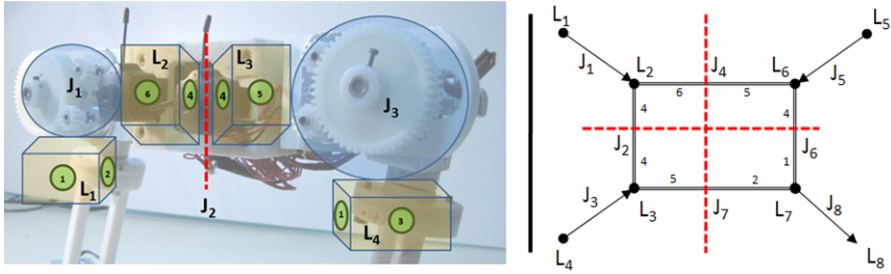


Fig. 3. a) Representative graph elements over each RobMAT module. b) 4-module configuration graph.

All this information is set forth in the Assembly Incidence Matrix AIM [10], so that it can be easily included in algorithms. The AIM is a $(N+1) \times (M+1)$ matrix with N vertices (v) and M edges (e). This matrix is formed by giving to each entry a_{ij} the number of the port that joins v_i and e_j , or 0 when no linking appears. The extra column $(M+1)$ indicates the link type, while the extra row $(N+1)$ shows the joint type. Figure 3 (a) shows the RobMAT module with its ports and their assigned numbers. It can be seen that module has six possible connection ports with other modules. To clear up some final concepts, Figure 3 (b) shows the graph of the 4-module RobMAT molecule, where each link is named L_i with $i=1,2,\dots,8$ and each joint is named J_i with $i=1,2,\dots,8$. To distinguish between the two kinds of joints, spherical joints are represented as a single-line edge and connectors as a double-line edge. Also, the spherical joint edge is directed (from -1 to 1) to show the order of its three axes. The arrow indicates where the end-effector link is located. Also, the number at both ends sides of each connector indicates the port of connection between the links. The graph therefore contains enough information to describe the modular robot assembly.

The AIM for the Figure 3 (b) graph is:

$$AIM = \begin{pmatrix} & J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 & J_8 & | & \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & | & P \\ 1 & 4 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & | & P \\ 0 & 4 & 1 & 0 & 0 & 0 & 5 & 0 & 0 & | & P \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & | & P \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & | & P \\ 0 & 0 & 0 & 5 & 1 & 4 & 0 & 0 & 0 & | & P \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & -1 & 0 & | & P \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & | & P \\ s & c & s & c & s & c & c & s & 0 & | & \end{pmatrix} \begin{matrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \end{matrix}$$

3.3 Modular Robot Configurations for Displacement Determined by the Complexity of the Graphs

Many configurations can be built for displacement, but just a few of them are efficient for the task. By analyzing the graph (links and joints structure), it can be determined the complexity for the robot to move from one point to another. An example of this situation can be noticed when comparing the six modules configuration from figure 4 with the six modules configuration from figure 5. Although the same number of

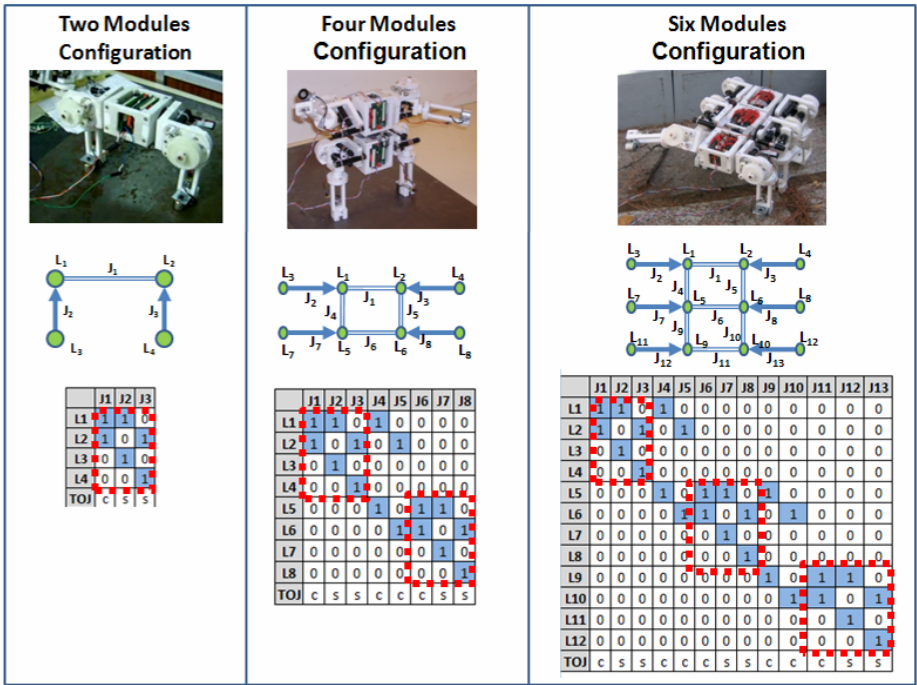


Fig. 4. Each configuration can be represented by graphs and with this a mathematical way to describe the structure generated

modules are used, and thanks to an additional platform (accessory) as common connection between each pair of modules in figure 5, the flexibility of the robot to execute displacement is higher when less links and joints appears at the graph.

Therefore, during real applications, it is important to seek out for equilibrium between the complexity of the task and the configuration of the modular robot, For example. the use of two molecules which have simple graphs due to its configuration (e.g. four modules configuration at figure 4) to carry on with the task, instead of using one molecule compound of many modules with high pull-push torque as a result, but forming a complex graph full of several links and joints (e.g. six modules configuration at figure 4).

When analyzing modular robot configurations in a simple way, it can be seen repeatable assembly patterns. This repeatability will depend undoubtedly when using the same configuration of a pair of modules to form chains, as is shown in figure 4. By this observation, it is achievable to describe the actual configuration in a simplified manner. The figure 4 shows different configurations formed by two, four, and six modules chains. It can be detected that the matrix (pattern) obtained by the two modules graph is repeated at four and six module matrix due to the use of a second and third same pair of modules configuration, the only difference is the connection between each set.

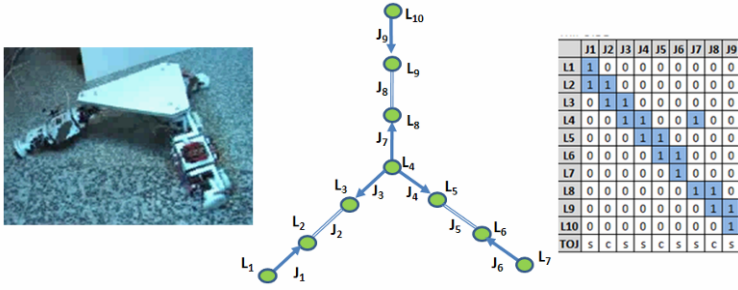


Fig. 5. While smaller the complexity of the graph, the highest flexibility of the robot to move

4 Forward Kinematics

Once a configuration can be described in a mathematical formulation the corresponding kinematical model can be obtained. All joints in the RobMAT are rotational; nevertheless modules can have any kind of joint. For this reason, it is helpful to apply Screw Theory [11]. This allows treating prismatic and rotational joints in the same expressions without specific changes. Also, almost any joint can be decomposed as a combination of those basic ones. A forward kinematics equation of an open chain robot can be uniformly expressed as a product of exponentials (POE) [12]. Using POE and Graph Theory, forward kinematics of tree-type modular robots can be obtained. The following equation is an example that solves RobMAT module direct kinematics (Figure 3). Let q_1, q_2 and q_3 be the generalized coordinates of the 3-DoF RobMAT module. The corresponding POE formula is given as follows:

$$H_e^0(q_1, q_2, q_3) = e^{\hat{S}_1 q_1} e^{\hat{S}_2 q_2} e^{\hat{S}_3 q_3} H_e^0(0)$$

where:

$$\begin{aligned} \hat{S}_1 &= \begin{pmatrix} w_1 = (0,0,-1) \\ v_1 = (0,0,0) \end{pmatrix} \\ \hat{S}_2 &= \begin{pmatrix} w_2 = (0,-1,0) \\ v_2 = \left(-\frac{L}{2}, 0,0\right) \end{pmatrix} \\ \hat{S}_3 &= \begin{pmatrix} w_3 = (1,0,0) \\ v_3 = \left(0,-\frac{L}{2},0\right) \end{pmatrix} \end{aligned} \quad H_e^0 = \begin{pmatrix} 0 & 0 & 1 & \frac{L}{2} \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & \frac{L}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(\hat{S}_i) is a screw and is expressed in the base frame (0). Furthermore, all coordinates are expressed in the base frame, and also in the zero position which is a fixed position. $H_e^0(0)$ is the well-known homogenous transformation between the end-effector frame (e) and the base frame at zero position. Once the module kinematics model has been determined, the kinematics of molecules can be obtained by compounding several module kinematics. Hence, an arbitrary module is designated as root, and position/orientation is propagated from this root to every end-side module through the modules in the molecule. In this way, the forward kinematical model is obtained.

To automate this process, it is important to know how modules are connected to each other. Depending on where the module port is attached, orientation and/or position changes will or will not be required. According to the port number in both links, a fixed homogeneous transformation can be obtained that provides the reference frames of neighbour modules.

For instance, the following described forward kinematics model of link L_8 (figure 3 (b)). The root link is L_1 and one possible path from L_1 to L_8 , which is obtained from the graph or AIM, would be $\{L_1, J_1, L_2, J_2, L_3, J_7, L_7, J_8, L_8\}$. Then, for each vertex on the path, a homogeneous transformation is added as shown in the next expression.

$$H = H_e^0 \cdot H_4^e \cdot D \cdot (H_4^e)^{-1} \cdot H_5^e \cdot D \cdot (H_2^0)^{-1} \cdot H_e^0$$

where:

$$H_1^0 = H_4^e = I_4$$

$$H_2^0 = H_6^e = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -w/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{array}{l} \text{Transformations} \\ \text{between} \\ \text{reference frames} \\ \text{and connectors} \end{array}$$

$$H_3^0 = H_5^e = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & w/2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Docking matrix}$$

Thus, by evaluating all paths from the root to the pending modules of each molecule, the full forward kinematics model is solved. This model is very important in order to obtain molecule workspace and to determine its functionality.

5 Conclusions

The possibility of modelling modular robot configurations can be achieved by graph theory. Its study generates a mathematical object so it can be manipulated to perform complex analyzes, like forward kinematics. Analyzing the graph of each corresponding robot configuration and comparing them with tested robot displacements a conclusion show up. The more complex the graph is (i.e. high number of links and joints), the robot flexibility to move from one point to another decreases, due to the reduced workspace of the DOF when having many modules so close. Although it is possible to represent any modular robot structure with graphs, just some configurations will be useful for real applications. Therefore, the search for equilibrium between complexity of the task and modular robot configuration must be achieved.

References

1. Yim, M., Duff, D., Roufas, K.D.: Walk on the Wild Side. *IEEE Robotics & Automation Magazine* 9(4), 49–53 (2002)
2. Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K.Y., Kokaji, S.: M-TRAN: Self-Reconfigurable Modular Robot System. *IEEE/ASME Trans. Mechatronics* 7(4), 431–441 (2002)
3. Castano, A., Behar, A., Will, P.M.: The CONRO Modules for Reconfigurable Robots. *IEEE/ASME Trans. Mechatronics* 7, 403–409 (2002)
4. Shen, W.-M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., Venkatesh, J.: Multi-mode Locomotion for Reconfigurable Robots. *Autonomous Robots* 20(2), 165–177 (2006)
5. Escalera, J.A., Ferre, M., Aracil, R., Baca, J.: ROBMAT: Teleoperation of a Modular Robot for Collaborative Manipulation. In: *Proc. 11th International Conference, Knowledge-Based Intelligent Information and Engineering Systems, Part 2*, pp. 1204–1213 (September 2007)
6. Crespo, A., Baca, J., Yerpes, A., Ferre, M., Aracil, R., Escalera, J.A.: CAN Application in Modular Systems. In: *12th International CAN Conference, Barcelona, Spain*, pp. 01-1 – 01-7 (2008)
7. Chen, I.-M., Yang, G.: Automatic generation of dynamics for modular robots with hybrid geometry. In: *Proceedings of 1997 IEEE International Conference on Robotics and Automation, April 20-25, vol. 3*, pp. 2288–2293 (1997)
8. Freudenstein, F., Dobrjanskyj, L.: On a Theory for the Type Synthesis of Mechanisms. In: *Proc. 11th Int. Congress of Applied Mechanics, Munich*, pp. 420–428 (1964)
9. Gross, J.L., Yellen, J.: *Graph Theory and Its applications*. Chapman & Hall/CRC (2006)
10. Chen, I.-M., Yang, G., Kang, I.-G.: Numerical inverse kinematics for modular reconfigurable robots. *Robotic Systems J.* 14, 213–225 (1999)
11. Ball, R.: *A Treatise on the Theory of Screws*. Cambridge University Press, Cambridge (1900)
12. Brockett, R.: Robotic Manipulators and the Product of Exponential Formula. In: *Mathematical Theory of Networks and Systems*, pp. 120–129 (1984)

A Hybrid Intelligent System for Robot Ego Motion Estimation with a 3D Camera

Ivan Villaverde and Manuel Graña*

Computational Intelligence Group
University of the Basque Country
<http://www.ehu.es/ccwintco>

Abstract. A Hybrid Intelligent System (HIS) for self-localization working on the readings of innovative 3D cameras is presented in this paper. The system includes a preprocessing step for cleaning the 3D camera readings. The HIS consist of two main modules. First the Self-Organizing Map (SOM) is used to provide models of the preprocessed 3D readings of the camera. The 2D grid of the SOM units is assumed as a surface modeling the 3D data obtained from each snapshot of the 3D camera. The second module is an Evolution Strategy, which is used to perform the estimation of the motion of the robot between frames. The fitness function of the Evolution Strategy (ES) is given by the distance computed as the matching of the SOM unit grids.

Keywords: Self Organizing Maps, evolutive computation, robot navigation.

1 Introduction

Until now, and despite the high potential of being able to get depth information from the environment, the use of 3D cameras on mobile robotics was extremely restricted by their size, power consumption and price. However, recently a new generation of affordable, small and lightweight cameras has been developed, opening a new and interesting research area in robotics. Our current line of research is directed to perform Simultaneous Localization and Mapping (SLAM) [1,6] using those new 3D cameras. In this paper, we report our first steps towards this objective, which consists in the estimation of the ego motion from the camera readings. The results obtained show that we can estimate the path followed by the robot as precisely as the odometry readings.

The system is composed of two main modules plus a preprocessing step. The first module is a SOM [2] that performs the approximation of the cleaned 3D readings with a 2D grid of units. The SOM robustness allows us to postulate the smooth variation of the fitted grids from one camera frame to the next, provided that the motion is slow enough. The second module is an ES [3] which performs the task of estimating the motion parameters by searching on the space of potential transformations of the SOM grids of units. The fitness function is the matching distance computed between the candidate transformed grid and the one to be predicted.

* This work was supported in part by the Spanish Ministerio de Educación y Ciencia under grant DPI2006-15346-C03-03.

In section 2 we will explain briefly the sensor characteristics in order to give a proper context to the data and its necessary preprocessing. In section 3 we present the HIS that performs the motion parameter estimation. In section 4 we present some experimental results on a real life wander of the Pioneer robot incorporating the camera. Finally, section 5 provides some conclusions and further research efforts.

2 Sensor and Data Pre-processing

In our work, we use a Swiss Ranger SR-3000 time-of-flight range camera [4]. Their full specs and capabilities are not in the scope of this paper, but some of its characteristics are relevant to how data is processed. The SR-3000 is based on phase-measuring Time of Flight principle [5]. The SR-3000 camera uses an array of near-infrared leds in order to illuminate the scene. Knowing the amplitude of the wavelength of the light emitted, it can determine the distance to objects measuring the phase delay of the reflected light, within a range of non-ambiguity determined by the wavelength. Figure 1 shows the optical image obtained from a specific robot position and orientation. This position will be the same for all illustrations in the paper. Figure 2 shows the raw data provided by the 3D camera, which consists of two matrices storing at each pixel the measured distance and the intensity of the reflected infrared light. Since each pixel corresponds to fixed azimuth and zenith angles in spherical coordinates, we can translate the distance image to more appropriate Cartesian coordinates (shown in figure 3) and obtain a cloud of 3D points, representing the camera measures.

This 3D point cloud is extremely noisy and it's necessary a strong filtering in order to retain only relevant points. As can be seen in (fig 2, 3) most of the noise comes from measurements beyond the non-ambiguity range: far away measurements (from space that is through the door) appear to be very close to the camera, forming a dense cone near the vertex of the cloud of points. Also, pixels without any object in their line of sight present measurements spread along all distances, caused by specular reflections. As those points are supposed to be far away, the reflection intensity should be very low. Taking this into account, we define a confidence value C for each pixel i , calculated proportionally to its measured distance and intensity:

$$C_i = I_i \times D_i \tag{1}$$

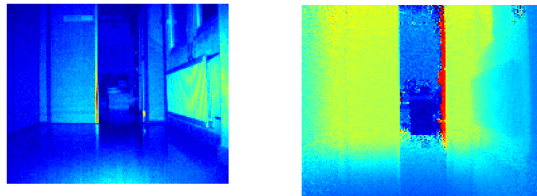


Fig. 1. Reference optical camera view

Fig. 2. Intensity (*left*) and distance (*right*) images from the 3D camera. Note that the distance measured beyond the door is very low (dark blue) due ambiguity range.

This confidence value allows filtering noisy points, while keeping relevant distant points with low intensity that could be discarded if filtering is done only by intensity.

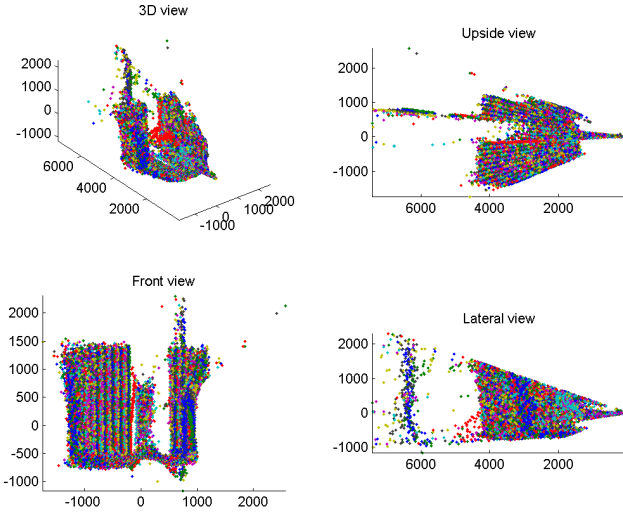


Fig. 3. Cloud of points in Cartesian coordinates extracted from the distance image

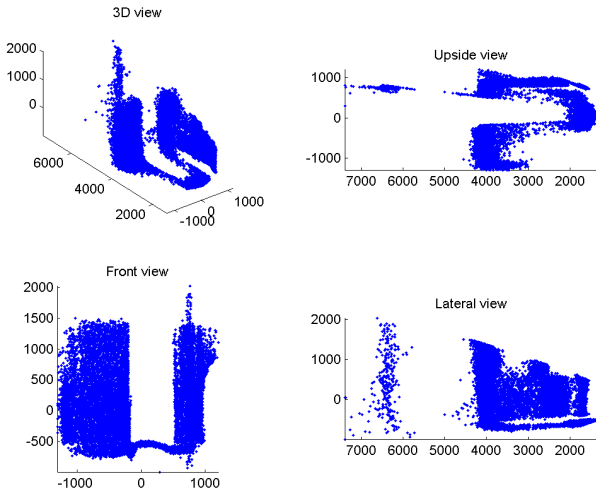


Fig. 4. Filtered points from figure 3. Relevant surfaces can be appreciated

3 Hybrid Intelligent System

The filtered cloud of points (figure 4) still presents two big problems, namely, the big size of the data set (usually more than 15.000 points) and the uncertainty in the

measurements of the surfaces. In order to cope with both problems, a Self Organizing Map [2] is trained to fit the cloud of points. In this way, we obtain from the SOM a grid that keeps the spatial shape of the points cloud and, hopefully, of the objects in the environment, and at the same time reduces dramatically the size of the data set Figure 5 shows the result of the trained SOM for the points in figure 4

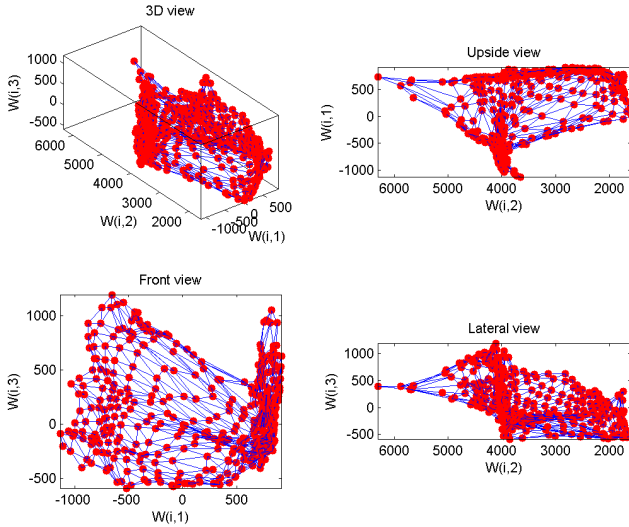


Fig. 5. SOM grid obtained after fitting the cloud of points

The data we are going to work with for the ego motion estimation consists of a series of grids G that approximate the shape of the environment in front of the robot at each time instant corresponding to a 3D camera frame. The robot is described by its position at each time instant t given by $P_t = (x_t, y_t, \theta_t)$, and the grid G_t fitted over the observed data. At the next time instant $t+1$ we obtain G_{t+1} from the camera. Our objective is to estimate the position P_{t+1} from the knowledge of the grid G_{t+1} . and the previous estimation of the position at time t . We apply an Evolution Strategy (ES) to search the new position of the robot. The ES individuals are hypothesis $h_i = (x_i, y_i, \theta_i)$ of position P_{t+1} . Mutation is performed by adding Gaussian perturbations with zero mean and standard deviation set heuristically (we are introducing now adaptive mutation parameters). We perform standard selection.

To compute the fitness function, we use G_{t+1} and G_t . Supposing the hypothesis h_i , G_{t+1} should be equivalent to G_t translated and rotated from P_t to h_i . The transformation matrix is:

$$T_i = \begin{bmatrix} \cos(\theta_i - \theta_t) & -\sin(\theta_i - \theta_t) & x_i - x_t \\ \sin(\theta_i - \theta_t) & \cos(\theta_i - \theta_t) & y_i - y_t \\ 0 & 0 & 1 \end{bmatrix}$$

Note that although the SOM's grid is three-dimensional, the displacement is only along the plane of the floor, so is not necessary to transform the vertical dimension. For each hypothesis h_i we have a prediction of the observed grid:

$$(\hat{G}_{t+1})_i = T_i \times G_t \tag{2}$$

which is used to calculate the fitness function as a matching distance between grids, computed as the sum of the individual differences in position of the grid nodes.

$$e(h_i) = \left\| (\hat{G}_{t+1})_i - G_{t+1} \right\|^2 \tag{3}$$

The full process of the algorithm, as outlined in Fig. 6, can be described by the following pseudo code:

```

For each new position
  Take measurements from the camera
  Filter cloud of 3D points
  Fit SOM to the cloud of filtered 3D points
  Generate an initial population
  Evolve until stop
  For each  $h_i$ 
    Transform  $G_t$  to  $(\hat{G}_{t+1})_i$  with  $h_i$ 
    Match  $(\hat{G}_{t+1})_i$  to  $G_{t+1}$ 
  End for
  Get the best  $h$ 
  Stop if not improvement
  Form new population
End evolve
Update position
End for
    
```

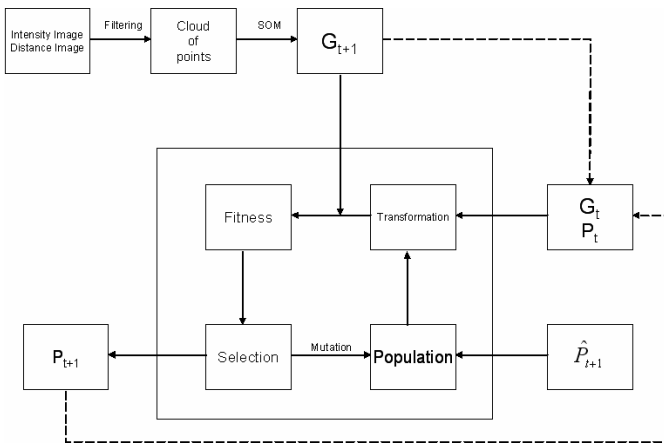


Fig. 6. Path estimation algorithm flow diagram

4 Experimental Settings and Preliminary Results

To test the approach, we have collected data from a series of pre-recorded walks across the corridors of our building. Being a first attempt, the robot motion configuration and its relation to the 3D camera setting were not well tuned, providing blurry images caused by the combination of slow shutter speeds and the robot movement. New experimental data will be of improved quality. A 20x20 SOM was used to fit the cloud of points provided by the 3D camera in each position.

Each odometry reading was put in relation to the previous one. This provides us with a reading of the actual displacement from point to point, instead of the global position obtained initially from the robot. This has also the double advantage of easing the transformation matrix calculation (being in fact a shift to the origin of the last, and supposedly correct, position) and the actualization of measurements after corrections.

The ES populations are composed of 20 individuals. Each generation, the best fitted one third is passed to the next generation and is used as parents for the rest two thirds of the generation, mutating two times each one to obtain two descendants.

The grid matching function used as fitness function (4) is the simplest one. We assume that the basic structure of the fitted SOM will be similar between two consecutive 3D frames, and that the same node in two consecutive SOM corresponds approximately to the same point in the surface. So, $e(h_i)$ comes from the node to node Euclidean distance between G_{t+1} and $(\hat{G}_{t+1})_i$.

Fast operation is a critical issue in mobile robotics. Taking this into account, waiting for the ES to fully converge to an optimal result could be too expensive in time cost for

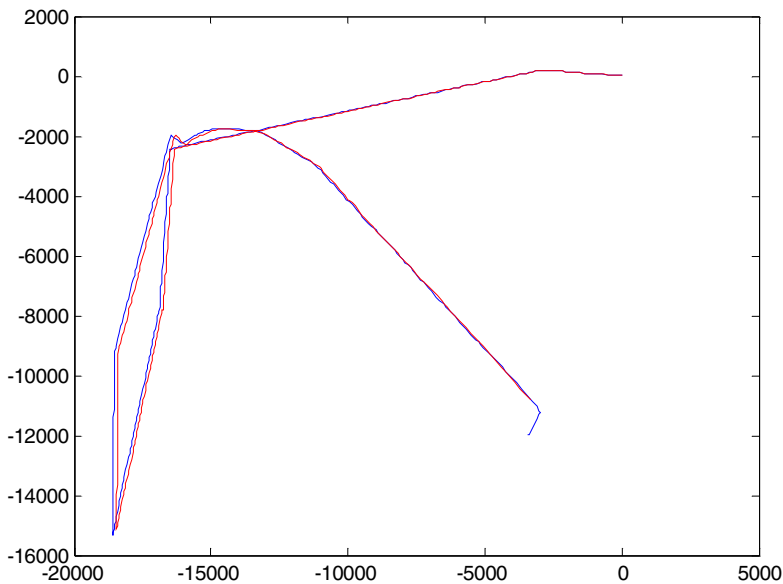


Fig. 7. Path estimated by odometry (blue line) versus corrected path (red line)

the benefit of getting a few millimeters more precise localization. Supposing that the initially estimated position is noisy but approximately correct locally, we assume as the stopping condition the lack of improvement in one generation. In this way, the ES will be very fast when there is little error in the initial population, but will take longer when we have an initial very noisy estimation with more error. As a handicap, this condition can cause that the algorithm stops easily in local minima, but we expect that this error will be compensated over time, corrected by following estimations.

Figure 7 shows the results of the ego motion estimation for the experimental data recorded. It shows the path as estimated by our approach (red line) versus the odometry readings (blue). It can be appreciated that our approach follows the odometry path very closely.

5 Conclusions and Further Work

In this paper, we have presented a new mobile robot ego motion estimation algorithm, based on information obtained from a 3D camera and a hybrid intelligent system composed of a SOM and an ES. The 3D camera provides information about the distances to the objects in the robot environment. This information is translated to a cloud of points, which is filtered and fitted with a SOM, in order to obtain a grid with the shape of the objects in the near environment of the robot. Those grids are used to estimate the next position, evolving a population of hypothesised positions using grid matching as fitness function. Some preliminary experimental results have also been presented.

Although the relevance of those preliminary results doesn't seem impressive, several aspects of the experimental settings must be taken in account. First, the experimental recorded set was of quite bad quality. Default 3D camera settings are set up for static operation and are not suitable for operation onboard a moving robot. Camera settings (e.g. shutter speed) must be tuned in order to face correctly that situation. Also other aspects of the camera, like the mount on the robot, should be adapted.

Most of the elements of the algorithm can also be tuned. We are planning to use Neural Gas networks to fit the cloud of points, in order to avoid the existence of nodes "stuck in the middle of nowhere" shown by the SOM grids, due to the topological preservation properties of SOM. Also, different grid matching techniques can be developed and tested. There are a number of improvements that can be tested on the ES parameters: population size, stopping condition, adaptive mutation.

We also plan to integrate this technique into a Kalman or particle filter SLAM architecture. We plan to fuse the information from the 3D camera with the optical camera and other sensors to enhance the robustness of SLAM systems on the Pioneer robot.

References

- [1] Dissanayake, G., et al.: A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans Robotics and Automation* 17(3), 229–241 (2001)
- [2] Kohonen, T.: *Self-Organizing Maps*, 3rd edn. Springer, Heidelberg (2000)
- [3] Randy, L.H., Sue, E.H.: *Practical Genetic algorithms*, 2nd edn. Wiley-Interscience, Chichester (2004)

- [4] Oggier, T., Lehmann, M., Kaufmann, R., Schweizer, M., Richter, M., Metzler, P., Lang, G., Lustenberger, F., Blanc, N.: An all-solid-state optical range camera for 3D-real-time imaging with sub-centimeter depth-resolution (SwissRanger). In: Proc. SPIE, vol. 5249, pp. 634–545 (2003)
- [5] Lange, R., Seitz, P.: Solid-State Time-of-Flight Range Camera. *IEEE J. Quantum Electronics* 37, 390–397 (2001)
- [6] Thrun, S.: Robotic mapping: A survey. In: Lakemeyer, G., Nebel, B. (eds.) *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, San Francisco (2002)

Evolutionary Parametric Approach for Specular Correction in the Dichromatic Reflection Model

Ramón Moreno, Alicia d'Anjou, and Manuel Graña*

Computational Intelligence Group
University of the Basque Country
<http://www.ehu.es/ccwintco>

Abstract. Assuming the dichromatic image model we propose a global reduction of specular effects by means of parametric illumination gradient images obtained by fitting 2D Legendre polynomials to the specular component of the images. Fitting is done applying a $(\lambda + \mu)$ Evolution Strategy. The method could be applied to static robotic monitoring in teams of robots, where the illumination gradient image could be computed once and applied to successive frames until the illumination conditions change drastically. The method could be useful for the detection of image regions with different chromatic properties.

Keywords: Dichromatic Reflection Model, specular, diffuse, luminosity correction, Legendre polynomials, region detection.

1 Introduction

The framework of this paper is the work on the design of multirobot systems for highly unstructured environments, such as shipyards. There a potentially critical role is that of observer or monitoring, that is a member of the team located in a position where it can monitor the entire environment and serve this information to the remaining members of the team. This robot will be static once it has reached the surveillance position, so that images will be relatively static also. That means that illumination conditions will vary slowly and it is possible to perform illumination correction under real time constraints. The point of view of the images used in the experiments reported below is that of the mouth of a ship hold. Naturally, this kind of images contains a lot of specular regions, due to the presence of water and metal surfaces. Moreover, the illumination will change continuously (slowly) due to changes in natural illumination.

The estimation of reflection components in images has been treated from several points of view [1,2,6,7]. Our starting point will be the dichromatic reflection model (DRM) proposed by Shafer [8]. It has been the source for works on illumination chromaticity estimation [3,4,5,10,12] and for the separation of diffuse and specular components [9,11,13,14,15,16,17,18]. These methods are not appropriate for robotic environments due to their high computational cost.

* This work was supported in part by the Spanish Ministerio de Educación y Ciencia under grant DPI2006-15346-C03-03. Thanks to Navantia and the GII from the UDC for providing the real life and synthetic images.

Legendre polynomials have applied successfully to intensity inhomogeneity correction in MRI [20], as a parametric model of the inhomogeneity field that can be estimated by an energy minimization method like the Evolution Strategies (ES) [22]. Our approach mimics that one, applying it to the dichromatic model. We obtain an illumination gradient bias that can be used to normalize the images easing further segmentation and detection processes. We call “specular field” or “bias” this illumination gradient image. Due to the slow change in natural illumination, we expect that the estimated bias would valid for several frames, reducing the time constraints for real life application.

Section 2 contains the description of the entire approach. Section 3 presents the dichromatic reflection model and how it can be used for reflectance estimation. Section 4 introduces the Legendre polynomials and the ES used for the parameter estimation. Section 5 gives some results and section 6 ends giving some conclusions and further work directions.

2 Description of the Approach

Figure 1 shows a flow diagram describing the process followed to obtain a specular field. The starting point is the captured image I that need to be normalized in chromaticity. We use the method of Inverse Intensity Chromaticity [12] to obtain the illuminant chromaticity estimation I^{est} and the normalized image is given by the ratio $I^{norm} = I / I^{est}$. It is also needed to obtain in parallel a Specular-Free image [11] (SF), we use the Specular Free Two Band method [16,21]. Then we obtain the derivatives of the logarithm of SF and I^{norm} , to obtain the diffuse pixels. From them we select the most representative k classes, $\{\mu_1, \dots, \mu_k\}$ corresponding to chromatic regions in the image.

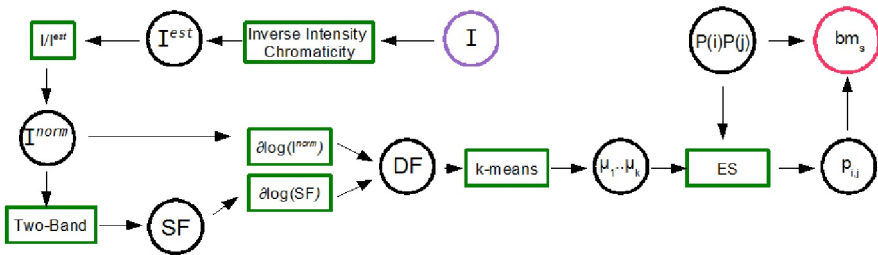


Fig. 1. Flow diagram showing the bm_s estimation process

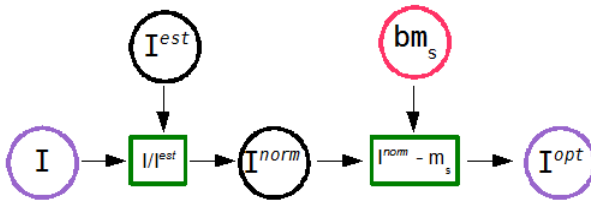


Fig 2. Illumination correction of an image using the specular field bm_s

Finally, the ES estimates the parameters of the 2D Legendre polynomials that give the estimation of the specular field denote “ bm_s ” in the figure. Figure 2 shows the flow diagram corresponding to the correction of the image removing the specular field.

3 Dichromatic Reflection Model

According to the DRM, the model of the image is given by the following expression:

$$I(x) = w_d \int_{\Omega} S(\lambda, x) E(\lambda, x) q(\lambda) d\lambda + w_s \int_{\Omega} S(\lambda, x) q(\lambda) d\lambda$$

Where $I = \{I_R, I_G, I_B\}$ is the color image intensity perceived on the camera sensor. The spatial parameter x denotes the two dimensional space coordinates. $q = \{q_R, q_G, q_B\}$ is the three-element vector of sensor sensitivity. $w_d(x)$ and $w_s(x)$ are the weighting factors for diffuse and specular components, they depend of the geometric structure at location x . $S(\lambda, x)$ is the diffuse spectral reflectance, $E(\lambda, x)$ is the spectral power distribution function of illumination, it is independent of the spatial location x because we assume a uniform illumination color. The integration is done over the visible spectrum Ω . For the sake of simplicity, the last equation can be written as: $I(X) = W_d B(x) + W_s G$, where the G component does no depend on x because we assume that the reflected light is identical to the projected light, and it is the same over the entire surface.

3.1 Chromaticity

The diffuse chromaticity is defined as $\Lambda(x) = B(x) / [B_R(x) + B_G(x) + B_B(x)]$ and the specular chromaticity is defined as $\Gamma = G / [G_R + G_G + G_B]$ then, an image expressed in the dichromatic reflection model is $I(x) = m_d(x)\Lambda(x) + m_s(x)\Gamma$, where $m_d(x)$ and $m_s(x)$ are the weightings factors for the diffuse and specular components.

3.2 Normalization

In some methods before separating specular and diffuse components it is necessary to carry out a process of normalization, because they need that the specular component must be pure white colour. This process requires the value of I^{est} which can be obtained by some methods [3,4,5,10,12]. It is computed as $I^{norm} = I / I^{est}$. After normalization, the image losses its specular component Γ , that is

$$I^{norm}(x) = m'_d(x)\Lambda'(x) + m'_s(x)$$

3.3 Specular-Free Image

The specular free image is critical for the detection of the reflectance component [11,16,19], we used the Specular-Free Two-Band method proposed by [3]. The

process is simple: it subtract to each pixel its minimum band. The image is geometrically identical to the original:

$$I^{sf} = m'_d(x)\Lambda^{sf}$$

3.4 Intensity Logarithmic Differentiation

This technique allows the detection of pure diffuse pixels $m_s = 0$, and, as a consequence, the specular pixels. Pure diffuse pixels allows us estimate the chromaticity of the surface. Assuming uniform colour pixels (Λ becomes independent from x) applying the logarithm and spatial differentiation

$$\frac{\partial}{\partial x} \log(I^{norm}(x)) = \frac{\partial}{\partial x} \log(m'_d(x)\Lambda'(x) + m'_s(x))$$

For diffuse pixels we have that $m_s = 0$, then $\frac{\partial}{\partial x} \log(I^{norm}(x)) = \frac{\partial}{\partial x} \log(m'_d(x))$.

For the Specular-Free image we have that $\frac{\partial}{\partial x} \log(I^{sf}(x)) = \frac{\partial}{\partial x} \log(m'_d(x))$. Therefore, the test for a diffuse pixel is that $\Delta(x) = \frac{\partial}{\partial x} \log(I(x)) - \frac{\partial}{\partial x} \log(I^{sf}(x)) = 0$.

4 Evolution Strategy

We are adapting the ideas about intensity inhomogeneity correction in MRI [20] to our problem. The first step is to propose an energy function whose minimization would solve our problem. This energy function comes from the dichromatic model

$$E_{tot} = \sum_{x \in I^{norm}} (I^{norm}(x) - bm_s(x, p) - \mu_k)^2, \quad (1)$$

where the specular field is given by

$$bm_s(x, p) = \sum_{i=0}^l \sum_{j=0}^l p_{i,j} P(i)P(j),$$

and the $P(i)P(j)$ denote products of 1D Legendre polynomials. The parameters of the minimization are the Legendre polynomials linear coefficients. The energy includes some class representatives μ_k of the image reflectance given by the representatives obtained from the diffuse pixels by some clustering process (i.e. k-means). Image pixels must be classified according to the classes before performing the bias search. To search for the optimal parameters of the Legendre 2D field, we use an $(\lambda + \mu)$ -ES [22]. Each individual in the ES is a matrix of coefficients of the Legendre polynomials. We follow the standard ES process. Individuals are mutated by the addition of random Gaussian values (still we have not implemented any self-adaptive search method modifying the mutation variances). The starting point is population of 50 individuals, which are the seeds for the process. We select the best 20 in an elitist selection process. They are the parents for the next generation.

If we introduce the dichromatic model into the energy function of equation 1, we have

$$E(x) = (m'_d(x)\Lambda'(x) + m'_s(x) - bm_s(x, p) - \mu_k)^2,$$

so that

$$E(x) = (m'_d(x)\Lambda'(x) - \mu_k)^2$$

and the energy is proportional to the diffuse component of the image:

$$E(x) \cong (m'_d(x))^2$$

5 Experimental Results

Experiments have been performed using Scilab and the SIP toolbox. Figure 3 shows the original full image of the ship hold being watered for cleaning. It also shows the recovered image when the higher degree of the Legendre polynomials composing the bias is 2. Figure 4 shows the bias and the recovered image when the higher Legendre polynomial degree is 3. These figures illustrate how increasing the model order allowing higher degree polynomials the estimated specular field tends to fit also the variations in reflectance. Lower order models are desired to obtain more robust estimations of the specular field. Figures 5 and 6 show the effect of the algorithm on the region containing the images of the human operators.

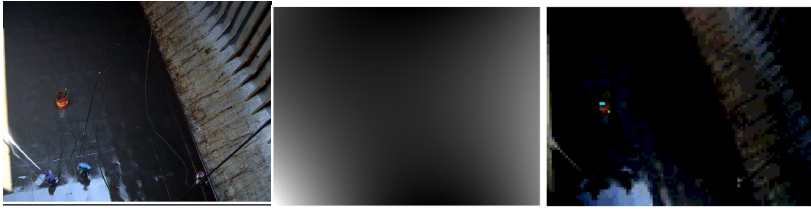


Fig. 3. From left to right: Original image, the estimated specular bias bm_s composed of polynomials of degree up to 2, and the corrected image obtained removing the specular bias

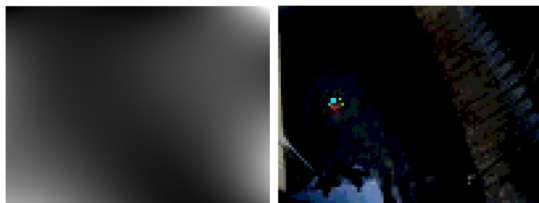


Fig. 4. From left to right: The estimated specular bias bm_s composed of polynomials of degree up to 3, and the corrected image obtained removing the specular bias



Fig 5. Left image: estimated specular field in the region of the workers. Right image: corrected removing the specular field.

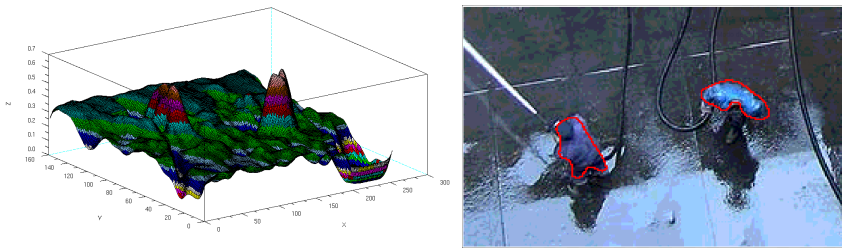


Fig. 6. Left image, map of the color saturation in HSV color space for image. Right image: identification of the workers performed thresholding the saturation image.

The resulting corrected image is, of course, more dark than the original image, but it retains all the geometric information, which can be observed computing the spatial gradient of the images, which we are not including for lack of space. The effect on the hold floor is that we get a constant intensity (color) image of it. The almost specular region on the lower left corner is greatly enhanced, making it more similar to the remaining floor surface. One of the goals of this work is to obtain robust segmentations of objects lying in the surface of the ship hold. It can be appreciated in figure 6 that the human operators are easily segmented despite the low color contrast between them and the hold floor. This segmentation is performed on the saturation values computed in the HSV space. We show in figure 6 the plot of the saturation values of the image, as a 3D surface (left). The contours for a saturation value of 0.5 are drawn in the figure (right). It can be appreciated that the works are fully and efficiently detected.

6 Conclusions and Further Work

We propose an image correction approach based on the modeling of the specular component of the image as bias field obtained as the linear composition of 2D Legendre polynomials. Model fitting is done by a $(\lambda + \mu)$ -ES on the space of the linear coefficients of the Legendre polynomials. The approach is based on the dichromatic reflectance model. The effect is that we can remove strong specular effects from

difficult scenes, such as the ship hold treated as example, allowing more robust segmentation of objects in the image. Further work must be addressed to the improved tuning of the ES and the study of the effect of the algorithm on the achromatic regions.

References

- [1] Feris, R., Raskar, R., Tan, K.-H., Turk, M.: Specular reduction with multish imaging. In: Proceedings of 17th Brazilian Symposium on Computer Graphics and Image Processing, 2004, October 17-20, pp. 316–321 (2004)
- [2] Hara, K., Nishino, K., Ikeuchi, K.: Light source position and reflectance estimation from a single view without the distant illumination assumption. *IEEE Trans. Pattern. Anal. Mach. Intell.* 27, 493–505 (2005)
- [3] Kuk-Jin Yoon, Y.-J.C., Kweon, I.S.: Illuminant chromaticity estimation using dichromatic slope and dichromatic line space. In: Korea-Japan Joint Workshop on Frontiers of Computer Vision. FCV, pp. 219–224 (2005)
- [4] Lehmann, T.M., Palm, C.: Color line search for illuminant estimation in real-world scenes. *Journal of the Optical Society of America A* 18, 2679–2691 (2001)
- [5] Marc Ebner, C.H.: On determining the color of the illuminant using the dichromatic reflection model. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 1–8. Springer, Heidelberg (2005)
- [6] Ping Tan, S.: Long Quan; Lin, Separation of highlight reflections on textured surfaces. *Computer Vision and Pattern Recognition* 2, 1855–1860 (2006)
- [7] Dror, E.A.R.O., Willsky, A.: Estimating surface reflectance properties from images under unknown illumination. In: Proceedings of the SPIE. Human Vision and Electronic Imaging IV, vol. 4299 (2001)
- [8] Shafer, S.A.: Using color to separate reflection components. *Color research and applications* 10, 43–51 (1984)
- [9] Tan, R., Ikeuchi, K.: Separating reflection components of textured surfaces using a single image. In: Proceedings of Ninth IEEE International Conference on Computer Vision, October 13-16, vol. 2, pp. 870–877 (2003)
- [10] Tan, T., Nishino, K., Ikeuchi, K.: Illumination chromaticity estimation using inverse-intensity chromaticity space. In: Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 18-20, vol. 1, pp. I673–I680 (2003)
- [11] Tan, R.T., Nishino, K., Ikeuchi, K.: Separating reflection components based on chromaticity and noise analysis. *IEEE Trans. Pattern. Anal. Mach. Intell.* 26, 1373–1379 (2004)
- [12] Tan, R.T., Nishino, K., Ikeuchi, K.: Color constancy through inverse-intensity chromaticity space. *J. Opt. Soc. Am. A Opt. Image. Sci. Vis.* 21, 321–334 (2004)
- [13] Tan, R., Ikeuchi, K.: Reflection components decomposition of textured surfaces using linear basis functions. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, June 20-25, vol. 1, pp. 125–131 (2005)
- [14] Umeyama, S., Godin, G.: Separation of diffuse and specular components of surface reflection by use of polarization and statistical analysis of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(5), 639–647 (2004)
- [15] Yoon, K.-J., Cho, Y.J., Kweon, I.-S.: Dichromatic-based color constancy using dichromatic slope and dichromatic line space. In: IEEE International Conference on Image Processing, 2005, ICIP 2005, September 11-14, vol. 3, p. III9603 (2005)

- [16] Yoon, K.-J., Choi, Y., Kweon, I.S.: Fast separation of reflection components using a specularly-invariant image representation. In: 2006 IEEE International Conference on Image Processing, October 8-11, pp. 973–976 (2006)
- [17] Yoon, K.-J., Kweon, I.-S.: Voting-based separation of diffuse and specular pixels. *Electronics Letters* 40, 1260–1261 (2004)
- [18] Yun-Chung Chung¹, S.C., Chang², S.-L., Chen¹, S.-W.: Dichromatic Reflection Separation from a Single Image, pp. 225–241. Springer, Heidelberg (2007)
- [19] Fu, Z., Tan, R., Caelli, T.: Specular free spectral imaging using orthogonal subspace projection. In: 18th International Conference on Pattern Recognition, 2006, ICPR 2006, vol. 1, pp. 812–815 (2006)
- [20] Styner, M.: Parametric Estimate of Intensity Inhomogeneities Applied to MRI. *IEEE Transactions On Medical Imaging* 19(3), 153–165 (2000)
- [21] Shen, H., Zhang, H., Shao, S., Xin, J.H.: Chromaticity-based separation of reflection components in a single image. *Pattern Recognition* (2008)
- [22] Randy, L.H., Sue, E.H.: *Practical Genetic algorithms*, 2nd edn. Wiley-Interscience, Chichester (2004)

On Distributed Cooperative Control for the Manipulation of a Hose by a Multirobot System*

José Manuel López-Guede, Manuel Graña, and Ekaitz Zulueta

Grupo de Inteligencia Computacional UPV/EHU
{jm.lopez,manuel.grana,ekaitz.zulueta}@ehu.es
www.ehu.es/ccwintco

Abstract. The manipulation of a hose can be formulated as the motion of a collection of autonomous robots linked by non-rigid elements so that the distance between robot units is not constant. The link may be a random component on the individual robot environment because under some circumstances it will be not known beforehand how the hose will influence the behaviour of the individual robot in response to control commands. We try to state the problem of distributed control of this kind of systems, under a gradation of conditions. We try to highlight the role of Hybrid Intelligent Systems in the development of solutions to this control problem. We have taken the works of Beard on UAV as the framework for the formulation of the system.

Keywords: Multirobot, cooperative control, distributed control, hose transport system, consensus.

1 Introduction

We are concerned with very unstructured environments for robot deployment, specially those where the resources and the workers are required to move over the product being built, such as ships in shipyards, or buildings. In these cases, hoses play a very important role, because they allow the transportation of power and other resources (water, air and other fluids) to the precise spatial point where they are required. Hose control and manipulation can be formulated as a multirobot system task. This paper is devoted to develop this idea and to define design elements and constraints that may allow a deep analysis and search for solutions to this problem. We will follow the methodological reasoning elements developed in [1-7] for the distributed control of unmanned air vehicles (UAV) applied to diverse tasks. Figure 1 provides some illustration of the kind of environment that we are considering. The left image shows a synthetic 3D model of a ship's hold where works can be placed. The right image corresponds to actual workmen doing some cleaning works inside a ship hold.

Section 2 gives a general introduction to the problem. Section 3 discusses modeling elements and alternatives. Section 4 gives some conclusions and expectations.

* This work was supported in part by the Spanish Ministerio de Educación y Ciencia under grant DPI2006-15346-C03-03. Navantia and the GII from the UDC have provided the images shown in figure 1.

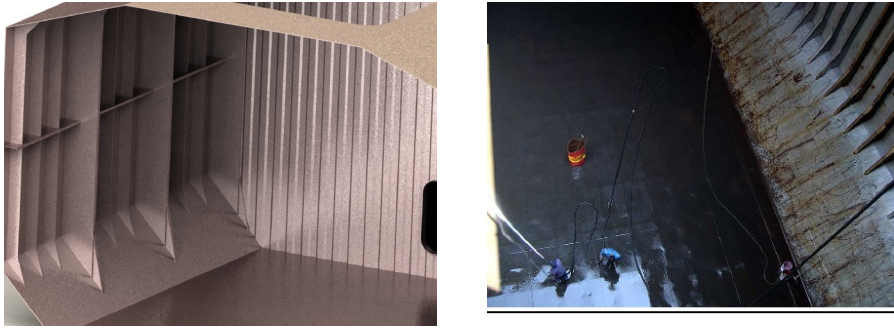


Fig. 1. Illustration of a shiyard environment, left a synthetic model of a ship hold, right: cleaning works with a water hose

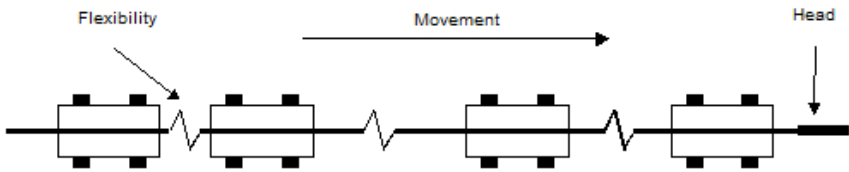


Fig. 2. Multirobot system attached to a flexible hose trying to move it in a linear motion

2 Description of the General System

Figure 2 shows an abstract illustration of the kind of multirobot systems we are trying to deal with. Individual robots, with motion (wheels or other) capabilities are attached to a hose. They try to bring the hose head to a certain point in space. The hose is not a rigid link between the robots, so that it can be twisted in some places and can have limited properties to transmit a force from one robot to another. When the hose is not tight between two robots, the effort of the leading one is devoted to tighten this hose segment. When it is tight, the effort may be wasted in pulling the rear robot, even stopping the lead robot. The possibility that the distance between robots varies according to their relative dynamics is a key feature of the system that differentiates it from other multirobot systems, such as snakes. In snakes the link between elements of the system is rigid and there is efficient transmission of forces between elements, such as rotation moments and traction forces. In the hose/robots system, the hose is not a reliable link and it is a source of nonlinearities in the dynamics of the system.

3 Discussion of System Elements

We will follow the methodology developed in [2,6,7] for the formulation of decentralized control systems, because it helps to bring some structure to the problem.

3.1 Notation

- $z_i(t)$: location of the i -th robot at time t . $\hat{z}_i(t)$ estimation based on incomplete information.
- $x_i(t)$: state of the i -th robot. $\hat{x}_i(t)$ estimation based on incomplete information. In this paper, $x_i(t) = z_i(t)$, nevertheless we retain this distinction between state and position.
- $X(t) = \{x_1, \dots, x_N\}$: the global state given by the collection of the local states of the robots. When only a guess or estimation is available we denote it $\hat{X}(t) = \{\hat{x}_1, \dots, \hat{x}_N\}$.
- z^* : goal position that the robots must go through.
- $v_i(t)$: speed vector of the i -th robot at time t . Sometimes we denote it $v_i(u_i)$ to highlight its dependence on the control commands.
- $u_i(\theta^*, X, t)$: Control command applied as a function of the current (global) state and the goal. In its simplest formulation the command control is the instantaneous velocity of the robot $v_i(t) = u_i(\theta^*, X, t)$.
- $U(\theta^*, X, t) = \{u_1(\theta^*, X, t), \dots, u_N(\theta^*, X, t)\}$: the global set of commands at time t .
- θ^* : Coordination variable, defined [2,7] as the minimum information that must be shared by the robots to obtain a cooperative behaviour. Here it is the goal position arrival time of the first robot at the hose head.

3.2 Cooperation Constraint

The cooperation constraint is in fact a kind of formal statement of the task to be accomplished by the robot team. For the system sketched in the previous section this task would be the transport of the hose through a predefined point. We will not take into account the boundary conditions imposed by the attachment of one end to the (power, water, etc.) source. There fore both ends of the hose will move freely. Under this assumption, we equate the transport of the hose to the motion of the robots attached to it. If the hose is well deployed the motion will be regular and all the robots will go through the specified point at regularly spaced time instants. This constraint is expressed by the following equation as function of the coordination variable θ^* :

$$J_{constraint}(\theta^*) = \sum_{i=1}^N \left\| z_i(\theta^* + (i-1)\Delta) - z^* \right\|^2 . \tag{1}$$

Where Δ is the desired interval of time between arrivals to the goal point. Depends on the number of robots in team and the length of the hose, assuming uniform distribution along of the robots along it. It is assumed that the robots are ordered according to their numbering.

The value of $J_{constraint}$ is zero when all the robots go through the goal \hat{z} according to their established physical ordering at uniform time intervals Δ .

3.3 Cooperation Objective

While the cooperation constraint states the task to be performed, the cooperation objective states some regularization property that enforces the cooperation between individual agents. Here this cooperation is enforced by trying to make the robots to follow the same motion vectors with a delay corresponding to their spatial distribution along the hose length. Taking into account this delay allows the team to follow complex motion patterns dictated by the hose head. The following equation describes this objective function.

$$J_{objective}(\theta^*, X, U, t) = \sum_{i=2}^N (v_i(X, U, t) - v_{i-1}(X, U, t - \Delta))^2 \quad (2)$$

Where the local velocity vector $v_i(X, U, t)$ depends on the conditions of the remaining robots: their states and control commands.

3.4 Coordination Function

The Coordination Function is the formal way to state the decoupling of the cooperation objective function into each agent's local representation. The coordination function states the local objective that the robot would pursue if working autonomously. Formally, the objective function is decoupled as follows:

$$J_{objective}(\theta^*, X, U, t) = \sum_{i=1}^N J_{cf,i}(\theta^*, x_i, u_i, X, U, t). \quad (3)$$

Each coordination function is given by the difference in velocities between the robot and its predecessor in the hose:

$$J_{cf,i}(\theta^*, x_i, u_i, X, U, t) = (v_i(X, U, t) - v_{i-1}(X, U, t - \Delta))^2 \quad (4)$$

3.5 Solving the Problem with Total Information

The distributed control problem can be stated as the minimization of the (decoupled) objective function, subject to the constraint expressed by the cooperation constraint function:

$$u_i = \arg \min \left\{ \sum_{i=1}^N J_{cf,i}(\theta^*, x_i, u_i, X, U, t) \right\} \quad (5)$$

subject to: $J_{constraint}(\theta^*) = 0.$

Assume that we have a mapping $f_i(X, U)$ that relates the robot state x_i and its control command u_i with the coordination variable θ^* and the stated goal position z^* . For instance, if the hose motion is along a line and the space between robots is set uniformly distributed, the time that will be in excess from its desired scheduled arrival time for the i -th robot is given by:

$$f_i(x_i, u_i) = (z^* - z_i(\theta^*)) / v_i(u_i) - \gamma_i \Delta . \quad (6)$$

If f_i is invertible, then we may have a function $u_i = f_i^{-1}(X, t)$ that will give us the command needed to apply to the robot to arrive at a desired time to the goal position relative to the hose head. In the very special circumstances said before, this function will have the following expression:

$$u_i = f_i^{-1}(x_i, t_0) = \frac{z^* - z_i(\theta^*)}{t_o + \gamma_i \Delta} . \quad (7)$$

If we have the above mapping and its inverse, the solution of the minimization problem in equation 5 could be computed searching first the control commands that give the minimum error approximation to the zero constraint function (i.e. using eq. 7), from the initial positions of the hose. Applying the mapping (i.e. eq. 6 in the simplest case) we would search for the optimal control commands to minimize the objective function. However, departing from the trivial case illustrated by equations 6 and 7, the minimization problem of equation 5 is difficult even in the case of complete information availability to each agent and the central control. Some factors that introduce complexity and nonlinearities are the following:

1. The hose elasticity imposes some constraints on the motion vectors, and conditions the control commands needed to obtain the desired motion vectors.
2. Hose weight and the distribution of the robot team members along it can introduce further non linear effects.
3. There is a distinction between the physical distance between the robots and their distance over the hose. If they are different then there will be interactions between the hose and the robots (i.e. blocking paths) that must be introduced in the prediction of the effect of control commands. They can require specific control strategies to obtain a good spatial distribution of hose and robots.
4. There are traction effects between robots where the hose plays the role of force transmission. Robots can contribute to the effort of robots ahead of them. Stuck robots may hinder the motion of robots ahead of them.
5. The misalignment of the motion vectors may cause chaotic behaviours.
6. Changes in hose content may produce changes in dynamic parameters (weight, elasticity, etc) that may affect the response to control commands in a dynamic way.

In [2] the dynamic response of the robots is quite simple, $v_i = u_i$, however in our case the dynamic response of a robot would be a (nonlinear) function of the hose elasticity and the position of neighbouring robots, that is $v_i = F(h, X, u_i)$ where h denotes the hose parameters (length, elasticity, etc.). When performing simulations, we could compute everything. But even in this very optimistic case, non-linear effects can produce non-invertible functions, so that the direct search methods (evolutionary algorithms) can be needed to search for the optimal control strategies. Also training

strategies (artificial neural networks, fuzzy systems) can be applied to build up approximations to the inverse control functions.

When dealing with real systems, even in the case of a central monitoring (robot) station, we need to predict the effect of the control commands on the system configuration. The precise model of the hose/robot dynamics, or an approximation, must be estimated from and validated against data in order to be useful. Even then the models must work on estimations of system state \hat{X} (the position of all robots and the hose) that must be obtained from sensors. For instance, trying to detect a hose in an image of a ship hold, such as the one in figure 1, can be tricky and need specific ways to remove illumination effects and algorithms for image segmentation. Of course there is room for a large collection of new and old algorithms.

3.6 Working with a Decentralized System

The case of the decentralized system as discussed in [2,3,4,7] is limited to obtain an estimation of the coordination variable θ^* on the basis of uncertain information. Each robot has its own estimation θ_i , it does communicate it to the remaining robots and improves it applying an averaging rule (called consensus schema), such as:

$$\theta_i[n+1] = \theta_i[n] + \sum_{j=1}^N g_{ij}[n] K_{ij} ((\theta_j + v_{ij}[n]) - \theta_i[n]), \quad (8)$$

where K_{ij} is a weighting matrix, v_{ij} denotes communication noise and g_{ij} models the existence of communication links between robots. These parameters are set so that the following condition always holds:

$$\sum_{j=1}^N g_{ij}[n] K_{ij} = 1 \quad (9)$$

It has been shown that this consensus schema converges to the true value of θ^* under some natural conditions [4]. In our case, the problem of decentralized control implies the estimation the local state and its relation to the whole system. The coordination variable will depend in a complex way of the whole system status. Each robot must autonomously estimate the condition of the hose and the robots in its neighbourhood. If we assume that there is no central sensing/controlling station, then the problem of determining the physical layout (self-perception) of the hose/robots system from the partial information given by each robot is a complex problem of sensing and fusing several information sources. We can identify some of the individual problems:

1. Computing the ego-motion parameters, to estimate the actual response of the robot to the control commands issued.
2. Estimating the position of neighbouring robots on the basis of communication of the ego-motion and/or the sensing via on board cameras or other sensors.
3. Estimating the layout of sections of the hose in the physical neighbourhood of the robot.
4. Detection of obstacles and environment conditions. The hose can be and obstacle by itself.

5 Conclusions

In this paper we have introduced the problem of decentralized control of a multirobot system for the manipulation of a hose in an unstructured environment. We have tried to pose it in the framework developed by Beard [1-7]. Our aim would be to set a hierarchy of control problems leading to the more autonomous decentralized case. The problem presents very significant departures from the assumptions in [1-7], because we have both strong and weak interactions between robots through the hose, depending on the spatial placement of the system elements (hose and robots). To study the stability of the system, we must define what are the desired stable behaviors and states. Then, the stability of the system must be analyzed in two circumstances. First in the case of total information and a centralized control, because even in this case there is no guarantee of convergence, due to the high number of degrees of freedom of the system. Second, we must study the case of decentralized control, where each local control has partial and noisy information. Usually, if the centralized system fulfills the stability conditions, the stability of the decentralized system depends of the convergence of the guessing process to the real value of the coordination variables. In this context, hybrid intelligent systems can play a number of roles from sensing to control design and system identification.

References

1. Beard, R.W., Lawton, J., Hadaegh, F.Y.: A Feedback Architecture for Formation Control. In: Proceedings of the 2000 American Control Conference, vol. 6, pp. 4087–4091 (2000)
2. Beard, R.W., McLain, T.W., Nelson, D.B.: Decentralized Cooperative Aerial Surveillance Using Fixed-Wing Miniature UAVs. Proceedings of the IEEE 94(7), 1306–1324 (2006)
3. Kingston, D., Beard, R., Casbeer, D.: Decentralized Perimeter Surveillance Using a Team of UAVs, <https://dspace.byu.edu/handle/1877/57>
4. Kingston, D., Wei, R., Beard, R.: Consensus Algorithms are Input-to-State Stable. In: 2005 American Control Conference, Portland, OR, USA, pp. 1686–1690 (2005)
5. McLain, T.W., Beard, R.W.: Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles. American Institute of Aeronautics and Astronautics, Inc. (2000)
6. McLain, T.W., Beard, R.W.: Cooperative Path Planning for Timing-Critical Missions. In: Proceedings of the American Control Conference, Denver, Colorado, USA (2003)
7. Ren, W., Beard, R.W.: Distributed Consensus in Multi-vehicle Cooperative Control. Theory and Applications. Springer, London (2008)

Multi-robot Route Following Using Omnidirectional Vision and Appearance-Based Representation of the Environment

Luis Payá, Oscar Reinoso, Arturo Gil, and Javier Sogorb

Departamento de Ingeniería de Sistemas Industriales,
Miguel Hernández University
Avda. de la Universidad, s/n.
03202 Elche (Alicante), Spain
{lpaya,o.reinoso}@umh.es

Abstract. In this work, a framework to carry out multi-robot route following is presented. With this approach, a team of mobile robots can follow a route that a leader robot has previously recorded or is recording at the moment. Each robot has a catadioptric system on it that provides omnidirectional images of the environment. To carry out the localization and control of each robot during route following, we have made use of an appearance-based philosophy in combination with compression techniques in the Fourier Domain and the Principal Components Analysis subspace. These techniques are especially interesting in the case of panoramic images due to their invariance to orientation and robustness to small changes in the environment. Several experiments with a team of mobile robots have been carried out to demonstrate the robustness of the approach in realistic office and laboratory environments.

Keywords: 1D Fourier Transform, Panoramic images, Principal Components Analysis, Probabilistic localization.

1 Introduction

Nowadays, the presence of robots in our houses and factories is continuously increasing. These robots have moved to such environments to fulfill our needs, usually, in repetitive or unpleasant tasks. However, it is desirable they show a higher degree of autonomy, fulfillment and robustness to expand its range of home applications. In order to achieve these aims, one of the most important skills a robot needs is the efficient environment representation and navigation, taking into account that in real working environments, the robot has to deal with some typical situations such as variation in illumination, occlusions and change in the position of some objects in the scene. The representation of the environment must cope with these features so that the robot is absolutely autonomous.

Previous research has shown how to face the problem of environment representation. In [1], a complete survey is presented with information about robot mapping and the different approaches that have been developed recently. Current works show how

a graph representation may be very efficient, such as [2], that constructs a weighted graph in which nodes are the images taken at certain positions and links denote similarity between images. SIFT features and the epipolar restriction are used to obtain a robust heading estimation between each pair of panoramic images. A similar approach is presented in [3], that is based on the use of hierarchical cognitive maps. The place cells represent the scenes through a PCA compression and the information provided by an electronic compass is used to compute the connectedness between nodes. A similar probabilistic representation is presented in [4]. Based on objects, they build a global topological representation of places with object graphs serving as local maps. The objects are recognized by means of the SIFT points extracted from the images captured by a stereo pair. Finally, the work in [5] presents a framework approach to robot mapping using omnidirectional vision and Fourier transformation that allows hierarchical localization with variable accuracy and self-organization of the visual memory of the environment.

In some applications, the use of a team of robots may help to make the achievement of the task faster and more reliable. In such situations, each robot works with incomplete and changing information that has, also, a high degree of uncertainty. This way, only a suitable choice of the representation and an effective communication between the members of the team can provide the robots with a complete knowledge of the environment where they move. As an example, [6] presents a probabilistic EKF algorithm where a team of robots builds a map online, while simultaneously they localize themselves. In [7], a map is build using visual appearance. From sequences of images, acquired by a team of robots, subsequences of visually similar images are detected and finally, the local maps are merged into a single map.

A typical problem in collaborative robotics implies following a path e.g., to perform a surveillance task in an office environment or an assembly or delivery task in an industrial environment. Also, the problem of formations, where a team of robots must navigate keeping a relative spatial distribution of the robot positions can be seen as a problem of path following, where one or several robots must follow the path the leader is recording with an offset either in space or in time. Some approaches suggest that this process can be achieved just comparing the general visual information of the scenes, without necessity of extracting any feature. These appearance-based approaches are especially useful for complicated scenes in unstructured environments where appropriate models for recognition are difficult to create. As an example, [8] addresses a method where several low-resolution images along the route to follow are stored. This approach is not able to cope with large and varying environments and is not robust. [9] uses PCA compression of the scenes in an incremental way, what allows multi-robot route following.

In this paper, we present an appearance-based method for multi-robot route following where 1D Fourier Transform on omnidirectional images is used to build the database, PCA compression is used to compute the control action of the robot and a probabilistic Markov process is implemented for robot localization. Section 2 presents the appearance-based process used to represent the environment and the previous work done in this field. Section 3 shows the implementation of the multi-robot route following application. Some experiments have been carried out with two mobile robots, whose results are exposed in section 4. To finish, the conclusions of the work are detailed.

2 Representation of the Environment

The main goal of the work is to create a framework to carry out multi-robot route following using just visual information and with an appearance-based approach. In this task, a leader robot goes through the desired route while a team of robots follows it. The first step to accomplish this task is the learning phase, where the leader robot stores some general visual information along the route to follow. In this step, it is important to define correctly the representation of the environment to allow that any robot can follow the route of the leader one with an offset either in space or/and in time in an efficient way. An appearance-based approach for robot navigation implies using the information of the whole images without extracting any kind of landmark. Previous works [10] have used PCA compression of the scenes captured with a forward looking camera to reduce the amount of memory needed and the computational cost during the navigation task. However, to allow multi-robot route following while the leader is still going through the route, an incremental model to build the database is needed. Previous works have made use of incremental PCA with this goal [9].

In the current work we use the information provided by a catadioptric system composed by a forward looking camera and a hyperbolic mirror. This system provides omnidirectional images of the environment that can easily be transformed into panoramic images as shown on fig. 1(a) and 1(b).

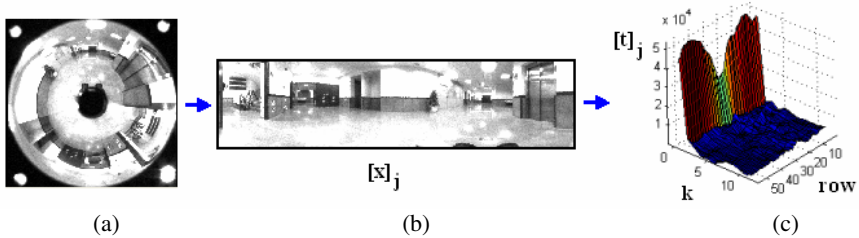


Fig. 1. (a) Original omni-directional image, (b) corresponding panoramic image and (c) power spectrum of the 1D Fourier Transform of the panoramic image computed row by row

This kind of images can be transformed to another compact representation into the Fourier domain [5]. The sequence of complex numbers $\{a_n\} = \{a_0, a_1, \dots, a_{N-1}\}$ can be transformed into the sequence of complex numbers $\{A_n\} = \{A_0, A_1, \dots, A_{N-1}\}$ using the Discrete Fourier Transform using the following expression:

$$A_k = \sum_{n=0}^{N-1} a_n \cdot e^{-j \frac{2\pi nk}{N}} ; \quad k = 0, \dots, N-1. \tag{1}$$

Using this expression, each row of the panoramic image can be transformed into the Fourier Domain. This transformation presents two interesting properties for robot mapping and localization when omnidirectional images are used. First, the most relevant information in the Fourier domain concentrates in the low frequency components. Furthermore, removing high frequency information can lead to an improvement in localization because these components are more affected by noise. The second

interesting property is the rotational invariance. If two images are acquired at the same point of the environment but having the robot different headings, then, the power spectrum of each row is the same in both cases. This is because these two rotated omnidirectional images lead to two panoramic images that are the same but shifted along the horizontal axis. If each row of the first image is represented with the sequence $\{a_n\}$ then and each row of the second image will be the sequence $\{a_{n-q}\}$, being q the amount of shift, that is proportional to the relative rotation between images. The rotational invariance can be deduced from the shift theorem, which can be expressed as:

$$\mathfrak{S}\{\{a_{n-q}\}\} = A_k \cdot e^{-j\frac{2\pi kq}{N}} \quad k = 0, \dots, N-1 \quad (2)$$

where $\mathfrak{S}\{\{a_{n-q}\}\}$ is the Fourier Transform of the shifted sequence and A_k are the components of the Fourier Transform of the non-shifted sequence. According to this expression, the Fourier Transform of a shifted sequence of numbers is equal to the Fourier Transform of the original signal multiplied by a complex number whose magnitude is the unit. This means that the amplitude of the Fourier Transform of the shifted image is the same as the original transform and there is only a phase change, proportional to the amount of shift q . This way, while the leader robot goes through the desired route, it captures N omnidirectional views, transforms them to the panoramic representation, resulting the images $[x]_j \in \mathfrak{R}^{R \times C}$, $j = 1, \dots, N$, and computes the 1D Fourier Transform row by row. After this process, the robot has created a database containing the Fourier matrixes $[t]_j \in \mathfrak{R}^{R \times F}$, $j = 1, \dots, N$, arranged by order of acquisition, being F the number of Fourier components retained.

3 Multi-robot Route Following

3.1 Localization of the Robot

Once the database is created or while it is being built, during the navigation phase, the second robot is situated on a point near the learned route. Then, it has to recognize which of the stored images is the nearest one to the image captured at the current position and drive to tend to the route, following it till the end. It must be carried out just comparing its current visual information with the information stored in the database. Two processes must run successively: auto-localization and control. During the auto-localization, the Fourier matrix of the current image is compared with the information in the database. Theoretically, the most similar point should correspond to the nearest position of the robot. However, in office environments which present a repetitive visual appearance, this simple localization method tends to fail often as a result of the aperture problem. This means that the visual information captured at two different locations that are far away can be very similar. To avoid these problems, a probabilistic approach based on a Markov process has been used. The current position of the robot can be estimated using the Bayes rule, using the next expression:

$$p(y_j | [t]_i; \theta) \propto p([t]_i | y_j; \theta) \cdot p(y_j) \quad (3)$$

where $p(y_j)$ denotes the probability that the robot is on the position y_j before observing the Fourier matrix $[t]_i$. This value is estimated using the previous information and the motion model. $p([t]_i | y_j)$ is the probability of observing $[t]_i$ if the position of the robot is y_j . This way, a method to estimate the observation model must be deduced. In this work, the distribution $p([t]_i | y_j)$ is modeled through a sum of Gaussian kernels, centered on the n most similar points of the route:

$$p([t]_i | y_j; \theta) = \frac{1}{n} \cdot \sum_{j=1}^n \left(\gamma_{ij} \cdot e^{-\left(\frac{y_i - y_j}{\sigma}\right)^2} \right); \quad j = 1, \dots, N. \quad (4)$$

Each kernel is weighted by the value of confidence γ_{ij} , that is the degree of similarity between the Fourier matrix of the currently captured image, $[t]_i$ and the j -th Fourier matrix of the database. Then, these kernels are convolved with a Gaussian function that models the motion of the robot, and that depends on the previous position and velocity of the robot. At last, the new position is considered at the point with highest contribution probability.

3.2 Control of the Robot

Once we have a new matching, in the control phase, the current visual information must be compared with the matched information of the database, and from this comparison, a control law must be deduced to lead the robot to the route. To do it, once the robot knows its position, it has to compute its orientation, comparing to the original heading that the first robot had when it captured the image at that position.

When working in the Fourier domain, eq. 2 allows computing the amount of shift q and so, the orientation of the robot. However, after some localization experiments, this expression has showed a very unstable behavior, and it has not been able to deal with small changes in illumination, noise and changes in the position of some objects in the environment. This way, an alternative procedure to retrieve the orientation has been developed, which is based on the use of sub-windows on the panoramic images. From each panoramic image stored in the database, $[x]_j$, a set of N' sub-windows is obtained from the whole image, where $\vec{w}_j^i \in \mathfrak{R}^{M \times 1}$ is each sub-window. These sub-windows are obtained scanning the original scene with a step in the horizontal axis (fig. 2). Carrying out a process of PCA compression, the PCA components $\vec{f}_j^i \in \mathfrak{R}^{K' \times 1}$ of each sub-window are calculated where $K' \leq N'$. Fig. 2 shows these projections as crosses in the case $K'=3$. On the other hand, during the autonomous navigation, another set of N' sub-windows is obtained from the current image, where $\vec{w}_N^i \in \mathfrak{R}^{M \times 1}$ is each sub-window. These windows are then projected onto the PCA subspace of the sub-images of the current image, giving as a result the dots on fig. 2. Then, the most similar projections to each one of them are extracted.

Arranging this information on a chart, we obtain the fig 3(a). To build this chart, the two most similar sub-images to each one are extracted. Extracting two sub-images instead of only one makes the method more robust to partial occlusions and small changes in the environment. The orientation of the robot is extracted from this chart using several least square fittings in an iterative process where data are shifted before

each new regression. After all the iterations, the best fitting results in a line whose ordinate in the origin is near to zero, whose slope is near to one and whose correlation coefficient is near to one. The orientation of the robot is proportional to the number of shifts until the best fitting is achieved. This method has demonstrated to be robust to recover the orientation of the robot.

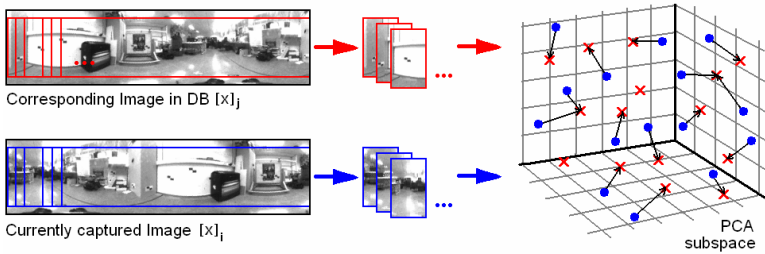


Fig. 2. From each image in the database, a set of sub-images is extracted and compressed using PCA methods. Then, a set of sub-images is extracted from the current images and projected on the same subspace. The most similar views to each of them are computed.

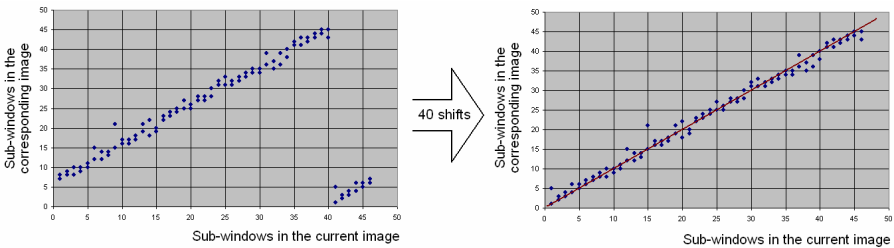


Fig. 3. (a) The sub-windows correspondences are arranged on a chart. The points of this chart are consecutively shifted until the best fitting is achieved (b). At each shift, all points are moved one position to the left and the first points are moved to the last position. The orientation of the robot is proportional to the number of shifts.

4 Results

Two Pioneer 3-AT robots with an omnidirectional vision system on each of them, and with processors P_A and P_B on-board, have been used to test the algorithms. R_A is the leader robot, which is recording the route and R_B is the follower one. Also, an independent processor P_C has been used. All the computers are connected in a CORBA-based architecture. To begin, the robot A is teleoperated through the desired route. After a while, R_B begins following this route. P_A is in charge of reading the new image, and storing it if is different enough to the previous ones. Then it computes the corresponding Fourier matrix and sends the panoramic image to P_C , which computes the PCA subspace of the sub-windows. When P_B captures a new image, it computes its Fourier matrix and calculates the nearest position of the database using the probabilistic approach. After this, P_C computes the control action to apply to R_B .

Several experiments have been carried out to validate the approach. Fig. 4 shows a typical route of around 50 meters, recorded in an office environment, and the route of the follower robot when it starts from two different points around it. Typically, the follower robot tends to the route and follows it, showing a great performance on the straight lines and a relatively bigger error in the turnings. However, with this approach, the robot is able to find the route and tend to it, showing a very stable behavior till the end. The control action implemented makes the robot move forward only if its orientation is close to the correct orientation. Otherwise, the robot will be applied only a steering velocity to correct the heading. In fact, in the second experiment, the robot is situated near to the route but with an orientation of about 180° with respect to the route. First, the robot turns until its orientation is correct and then it starts moving forward, with slight steering movements to correct its trajectory.

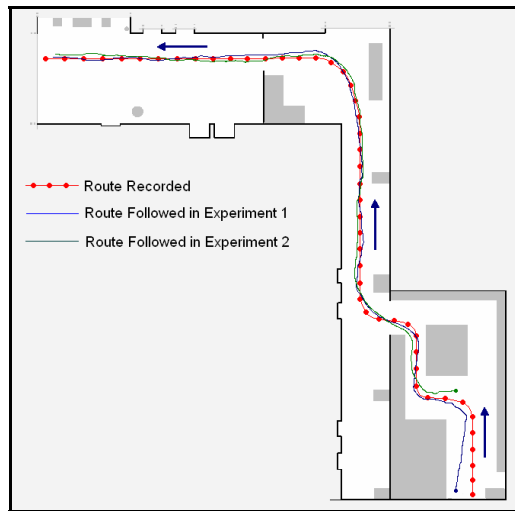


Fig. 4. Route recorded by the leader robot and routes followed by a follower robot in two experiments with different initial point

5 Conclusion

In this paper, an appearance-based multi-robot route-following scheme is presented. The proposed solution uses low resolution panoramic images and techniques for dimensionality reduction to extract the most relevant information along the environment. Thanks to its invariability to rotation, these techniques are especially interesting to be used in applications of map building and navigation. Also, an appearance-based method has been proposed to compute the orientation of the robot.

The final objective of the work is that other robots can follow a route from a distance (as in space or in time). To do it, a probabilistic algorithm has been implemented to calculate their current position among those that the leader has stored, and a controller has been implemented, also based on the appearance of the scenes, to calculate the linear and turning speeds of the robot.

Some experiments have been carried out with two Pioneer 3-AT robots using a CORBA-based architecture for communication. These experiments show how the process employed allows following a route in an accurate and robust way.

We are now working in other control methods to reduce the error during the navigation, studying the effects of illumination changes and scene changes more accurately. At last, more sophisticated ways of building a map are being evaluated so that the robot can find the route and follow it although its initial position is far from this route. These more complex maps should contain information of additional locations of the environment and they are expected to be useful in a number of applications in multi-robot navigation. Also, this application can be extended to a team of heterogeneous robots, equipped with different kinds of sensors. In this case, the hybrid systems approach would be useful to improve the performance of the task. Fuzzy logic and reinforced learning can be used both to build the database with information from several different sensors and to compare the information captured by the follower robot with the information in the database.

Acknowledgments. This work has been supported by MEC through the project DPI2007-61197: “Sistemas de percepción visual móvil y cooperativo como soporte para la realización de tareas con redes de robots”.

References

1. Thrun, S.: Robot Mapping: A Survey. Technical report, Exploring Artificial Intelligence in the New Milenium, pp. 1–35. Morgan Kaufmann, San Francisco (2003)
2. Booij, O., Terwijn, B., Zivkovic, Z., Kröse, B.: Navigation using an Appearance Based Topological Map. In: IEEE International Conference on Robotics and Automation, pp. 3297–3932. IEEE Press, New York (2007)
3. Vasudevan, S., Gächter, S., Nguyen, V., Siegwart, R.: Cognitive Maps for Mobile Robots – An Object Based Approach. *Robotics and Autonomous Systems* 55, 35–371 (2007)
4. Stimec, A., Jogan, M., Leonardis, A.: A Hierarchy of Cognitive Maps from Panoramic Images. In: 10th Computer Vision Winter Workshop, Zell an der Pram, Austria (2005)
5. Menegatti, E., Maeda, T., Ishiguro, H.: Vimage-based Memory for Robot Navigation using Properties of Ominidirectional Images. *Robotics and Autonomous Systems* 47(4), 251–276 (2004)
6. Thrun, S.: A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research* 20(5), 335–363 (2001)
7. Ho, K., Newman, P.: Multiple Map Intersection Detection using Visual Appearance. In: 3rd International Conference on Computational, Intelligence, Robotics and Autonomous Systems, Singapore (2005)
8. Matsumoto, Y., Ikeda, K., Inaba, M., Inoue, H.: Visual Navigation Using Omnidirectional View Sequence. In: IEEE International Conference on Intelligent Robots and Systems, pp. 317–322. IEEE Press, New York (1999)
9. Payá, L., Reinoso, O., Gil, A., Pedrero, J., Ballesta, M.: Appearance-Based Multi-Robot Following Routes Using Incremental PCA. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part III. LNCS (LNAI), vol. 4694, pp. 1170–1178. Springer, Heidelberg (2007)
10. Payá, L., Vicente, M.A., Navarro, L., Reinoso, O., Fernández, C.: Continuous Navigation of a Mobile Robot with an Appearance-Based Method. In: 2nd International Conference on Informatics in Control, Automation and Robotics, Barcelona, Spain, pp. 443–446 (2005)

Using CBR Systems for Leukemia Classification

Juan M. Corchado and Juan F. De Paz

Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, España
{corchado, fcofds}@usal.es

Abstract. The continuous advances in genomics, and specifically in the field of transcriptome, require novel computational solutions capable of dealing with great amounts of data. Each expression analysis needs different techniques to explore the data and extract knowledge which allow patients classification. This paper presents a hybrid systems based on Case-based reasoning (CBR) for automatic classification of leukemia patients from Exon array data. The system incorporates novel algorithms for data mining that allow to filter and classify. The system has been tested and the results obtained are presented in this paper.

Keywords: Case-based Reasoning, dendogram, leukemia classification, data mining.

1 Introduction

During recent years there have been great advances in the field of Biomedicine [1]. The incorporation of computational and artificial intelligence techniques to the field of medicine has yielded remarkable progress in predicting and detecting diseases [1]. One of the areas of medicine which is essential and requires the implementation of techniques that facilitate automatic data processing is genomics. Genomics deals with the study of genes, their documentation, their structure and how they interact [2]. We distinguish different fields of study within the genome. One is transcriptome, which deals with the study of ribonucleic acid (RNA), and can be studied through expression analysis [3]. This technique studies RNA chains thereby identifying the level of expression for each gene studied. It consists of hybridizing a sample for a patient and colouring the cellular material with a special dye. This offers different levels of luminescence that can be represented as a data array. Traditionally, methods and tools have been developed to work with expression arrays containing about 50.000 data points. The emergence of the Exon arrays [5], holds important potential for biomedicine. However, the Exon arrays require novel tools and methods to work with very large (5.500.000) amounts of data.

Microarray technology has been performed for the identification of acute leukemia prognosis. Microarray has become an essential tool in genomic research, making it possible to investigate global gene expression in all aspects of human disease [6]. Microarray technology is based on a database of gene fragments called expressed sequence tags (ESTs), which are used to measure target abundance using the scanned intensities of fluorescence from tagged molecules hybridized to ESTs [6]. The process

of studying a microarray is called expression analysis and consists of a series of phases: data collection, data pre-processing, statistical analysis, and biological interpretation. These phases analysis consists basically of three stages: normalization and filtering; clustering and classification. These stages can be automated and included in a CBR [17] system.

This paper presents a hybrid system system that facilitates the analysis and classification of data from Exon arrays corresponding to patients with leukemia. The system is based on a CBR that incorporates techniques of data mining in the different phases. Leukemia, or blood cancer, is a disease that has a significant potential for cure if detected early [4]. The system proposed in the context of this work focuses on the detection of carcinogenic patterns in the data from Exon arrays.

The paper is structured as follows: The next section presents the problem that motivates this research, i.e., the classification of leukemia patients from samples obtained through Exon arrays. Section 2 and Section 3 describe the proposed hybrid system and how it is adapted to the problem under consideration. Finally, Section 4 presents the results and conclusions obtained after testing the model.

2 CBR System for Classifying Exon Array Data

The CBR developed tool receives data from the analysis of chips and is responsible for classifying of individuals based on evidence and existing data. Case-based Reasoning is a type of reasoning based on the use of past experiences [8]. CBR systems solve new problems by adapting solutions that have been used to solve similar problems in the past, and learning from each new experience. The primary concept when working with CBRs is the concept of case. A case can be defined as a past experience, and is composed of three elements: A problem description, which delineates the initial problem; a solution, which provides the sequence of actions carried out in order to solve the problem; and the final stage, which describes the state achieved once the solution was applied. A CBR manages cases (past experiences) to solve new problems. The way cases are managed is known as the CBR cycle, and consists of four sequential phases: retrieve, reuse, revise and retain. The retrieve phase starts when a new problem description is received.. In our case study, it conducted a filtering of variables, recovering important variables of the cases to determine the most influential in the conduct classification as explained in section 0. Once the most important variables have been retrieved, the reuse phase begins, adapting the solutions for the retrieved cases to obtain the clustering. Once this grouping is accomplished, the next step is to determine the provenance of the new individual to be evaluated. The revise phase consists of an expert revision for the solution proposed, and finally, the retain phase allows the system to learn from the experiences obtained in the three previous phases, updating the cases memory.

2.1 Retrieve

Contrary to what usually happens in the CBR, our case study is unique in that the number of variables is much greater than the number of cases. This leads to a change in the way the CBR functions so that instead of recovering cases at this stage, important variables are retrieved. Traditionally, only the similar cases to the current

problem are recovered, often because of performance, and then adapted. In the case study, the number of cases is not the problem, rather the number of variables. For this reason variables are retrieved at this stage and then, depending on the identified variables, the other stages of the CBR are carried out. This phase will be broken down into 5 stages which are described below:

RMA: The RMA (Robust Multi-array Average) [9] algorithm is frequently used for pre-processing Affymetrix microarray data. RMA consists of three steps: (i) Background Correction; (ii) Quantile Normalization (the goal of which is to make the distribution of probe intensities the same for arrays); and (iii) Expression Calculation: performed separately for each probe set n . To obtain an expression measure we assume that for each probe set n , the background adjusted, normalized and log transformed intensities, follow a linear additive mode.

Control and error: During this phase, all probes used for testing hybridization are eliminated. These probes have no relevance at the time when individuals are classified... Therefore, the probes control will not be useful in grouping individuals. On occasion, some of the measures made during hybridization may be erroneous; not so with the control variables. In this case, the erroneous probes that were marked during the implementation of the RMA must be eliminated.

Variability: Once both the control and the erroneous probes have been eliminated, the filtering begins. The first stage is to remove the probes that have low variability. This work is carried out according to the following steps:

1. Calculate the standard deviation for each of the probes
2. Standardize the above values
3. Discard of probes for which the values meet the following condition $z < -1.0$.

Uniform Distribution: Finally, all remaining variables that follow a uniform distribution are eliminated. The variables that follow a uniform distribution will not allow the separation of individuals. Therefore, the variables that do not follow this distribution will be really useful variables in the classification of the cases. The contrast of assumptions followed is explained below, using the Kolmogorov-Smirnov [14] test as an example. The selected level of significance $\alpha = 0.05$.

Correlations: At the last stage of the filtering process, correlated variables are eliminated so that only the independent variables remain. To this end, the linear correlation index of Pearson is calculated and the probes meeting the following condition are eliminated. $\alpha = 0.05$.

2.2 Reuse

Once filtered and standardized the data using different techniques of data mining, the system produce a set of values x_{ij} with $i = 1 \dots N$, $j = 1 \dots s$ where N is the total number of cases, s the number of end probes. The next step is to perform the clustering of individuals based on their proximity according to their probes. Since the problem on which this study is based contained no prior classification with which training could take place, a technique of unsupervised classification was used. There is a wide range of

possibilities in data mining. Some of these techniques are artificial neural networks such as SOM [10] (self-organizing map), GNG [11] (Growing neural Gas) resulting from the union of techniques CHL [12] (Competitive Hebbian Learning) and NG [13] (neural gas), GCS [11] (Growing Cell Structure). There are other techniques with less computational cost that provide efficient results. Among them we can find the dendrogram and the PAM method [18] (Partitioning Around Medoids). A dendrogram [19] is a ascendant hierarchical method with graphical representation that facilitates the interpretation of results and allows an easy way to establish groups without prior knowledge. The PAM method requires a selection of the number of clusters previous to its execution.

The dendrograms are hierarchical methods that initially define as conglomerates for each available cases. At each stage the method joins those conglomerates of smaller distance and calculates the distance of the conglomerate with everyone else. The new distances are updated in the matrix of distances. The process finishes when there is one only conglomerate (agglomerative method). The distance metric used in this paper has been the average linkage. This metric calculates the average distance of each pair of nodes for the two groups, and based on these distances mergers the groups. The metric is known as unweighted pair group method using arithmetic averages (UPGMA) [20]. The clustering obtained is compared to the individuals that have already been classified by an expert. The percentage of error represents the confidence level.

Once, the clusters have been made, the new sample is classified. The KNN algorithm [21] (K-Nearest Neighbour) is used. KNN allows setting probabilistic values based on its neighbours. It is necessary to establish a measure of similarity to calculate the distance between individuals. The similarity measure used is as follows:

$$d(n, m) = \sum_{i=1}^s f(x_{ni}, x_{mi}) * w_i \quad (1)$$

Where s is the total number variables, n and m the cases, w_i the value obtained in the uniform test and f the Minkowski [15] Distance that is given for the following equation.

$$f(x, y) = \sqrt[p]{\sum_i |x_i - y_j|^p} \quad \text{con } x_i, y_j \in R^p \quad (2)$$

2.3 Revise and Retain

The revision is carried out by an expert who determines the correction with the group assigned by the system. If the assignation is considered correct, then the retrieve and reuse phases are carried out again so that the system is ready for the next classification. If the prognosis of the expert differs from the system, the case is not incorporated until the final medical diagnosis is carried out.

3 Case Study

In the case study presented in the paper, 232 samples were available from analyses performed on patients through punctures in marrow or blood samples, which have

been hybridized and analyzed through Exon arrays manufactured by Affymetrix. The aim of the tests performed is to determine whether the system is able to classify new patients based on the previous cases that were analyzed and stored.

Figure 1 shows a scheme of the bio-inspired model intended to resolve the problem described in Section 2. The proposed model follows the procedures that are performed in medical centres. As can be seen in Figure 1, a previous phase, external to the model, consists of a set of tests which allow us to obtain data from the chips and are carried out by the laboratory personnel. The chips are hybridized and explored by means of a scanner, obtaining information on the marking of several genes based on the luminescence. At that point, the CBR-based model starts to process the data obtained from the Exon arrays.

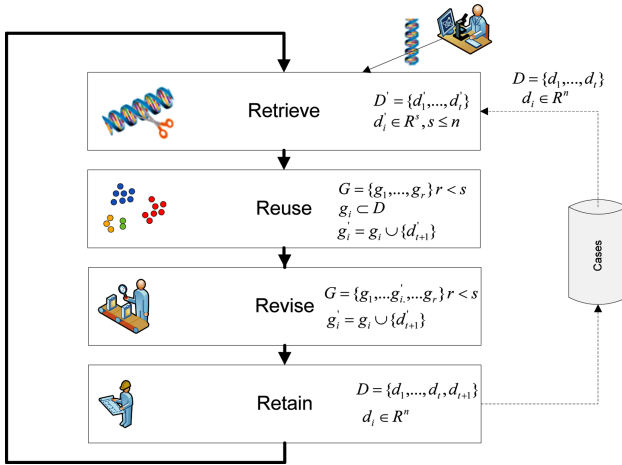


Fig. 1. Proposed CBR model

The retrieve phase receives an array with a patient’s data as input information. It should be noted that there is no filtering of the patients, since it is the work of the researcher conducting this task. The retrieve step filters genes but never patients. The aim of this phase is to reduce the search space to find data from the previous cases which are similar to the current problem. The set of patients is represented as $D = \{d_1, \dots, d_t\}$, where $d_i \in R^n$ represents the patient i and n represents the number of probes taken into consideration and t the number of cases. As explained in Section 0, during the retrieve phase the data are normalized by the RMA algorithm [9] and the dimensionality is reduced bearing in mind, above all, the variability, distribution and correlation of probes. The result of this phase reduces any information not considered meaningful to perform the classification. The new set of patients is defined through s variables $D' = \{d'_1, \dots, d'_s\}$ $d'_i \in R^s, s \leq n$.

The reuse phase uses the information obtained in the previous step to classify the patient into a leukemia group. The patients are first grouped into clusters. The data coming from the retriever phase consists of a group of patients $D' = \{d'_1, \dots, d'_t\}$

with $d'_i \in R^s, s \leq n$, each one characterized by a set of meaningful attributes $d'_i = (x_{i1}, \dots, x_{is})$, where x_{ij} is the luminescence value of the probe i for the patient j . In order to create clusters and consequently obtain patterns to classify the new patient, the reuse phase implements a hierarchical classification method known as dendrogram and explained in section 0. The system classifies the patients by taking into account their proximity and their density, in such a way that the result provided is a set G where $G = \{g_1, \dots, g_r\} r < s$. $g_i \subset D$, $g_i \cap g_j = \emptyset$ with $i \neq j$ and $i, j < r$. The set G is composed of a group of clusters, each of them containing patients with a similar disease. The clusters have been constructed by taking into account the similarity between the patient's meaningful symptoms. Once the clusters have been obtained, the system can classify the new patient by assigning him to one of the clusters. The new patient is defined as d'_{t+1} and his membership to a group is determined by a similarity function defined in [1]. The result of the reuse phase is a group of clusters $G = \{g_1, \dots, g'_i, \dots, g_r\} r < s$ where $g'_i = g_i \cup \{d'_{t+1}\}$.

An expert from the Cancer Institute is in charge of the revision process. This expert determines if $g'_i = g_i \cup \{d'_{t+1}\}$ can be considered as correct. In the retain phase the system learns from the new experience. If the classification is considered successful, then the patient is added to the memory case $D = \{d_1, \dots, d_t, d'_{t+1}\}$.

4 Results and Conclusions

This paper has presented a CBR system which allows automatic cancer diagnosis for patients using data from Exon arrays. The model combines techniques for the reduction of the dimensionality of the original data set and a novel method of clustering for classifying patients. The system works in a way similar to how human specialists operate in the laboratory, but is able to work with great amounts of data and make decisions automatically, thus reducing significantly both the time required to make a prediction, and the rate of human error due to confusion. The CBR system presented in this work focused on identifying the important variables for each of the variants of blood cancer so that patients can be classified according to these variables.

In the study of leukemia on the basis of data from Exon arrays, the process of filtering data acquires special importance. In the experiments reported in this paper, we worked with a database of bone marrow cases from 232 adult patients with five types of leukaemia. The data consisted of 5.500.000 scanned intensities. The retrieve stage of the proposed CBR system presents a technique to reduce the dimensionality of the data. The total number of variables selected in our experiments was reduced to 883, which increased the efficiency of the cluster probe. In addition, the selected variables resulted in a classification similar to that already achieved by experts from the laboratory of the Institute of Cancer. To try to increase the reduction of the dimensionality of the data we applied principal components (PCA) [16], following the method of Eigen values over 1. A total of 112 factors were generated, collecting 96% of the

variability. However, this reduction of the dimensionality was not appropriate in order to obtain a correct classification of the patients. Figure 2 shows the classification performed for patients from all the groups. In the left it is possible to observe the groups identified in the classification process. Cases interspersed represent individuals with different classification to the previous one. As shown in Figure 2 the number of misclassified individuals have been low.

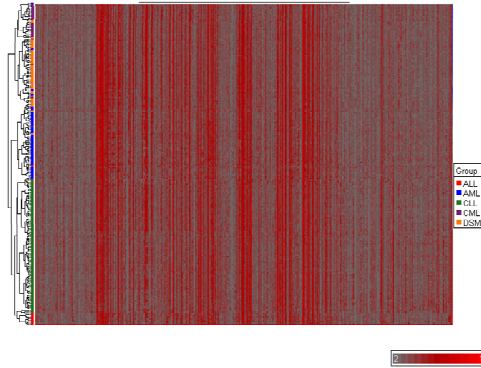


Fig. 2. Classification obtained

As demonstrated, the proposed system allows the reduction of the dimensionality based on the filtering of genes with little variability and those that do not allow a separation of individuals due to the distribution of data. It also presents a technique for clustering based in hierarchical methods. The results obtained from empirical studies are promising and highly appreciated by specialists from the laboratory, as they are provided with a tool that allows both the detection of genes and those variables that are most important for the detection of pathology, and the facilitation of a classification and reliable diagnosis, as shown by the results presented in this paper.

The next step, for future works, consists of incorporating new techniques for knowledge extraction, able to provide a human expert with information about the system-generated classification by means of a set of rules that are provided to support the decision-making process.

Acknowledgments. Special thanks to the Institute of Cancer for the information and technology provided.

References

1. Shortliffe, E.H., Cimino, J.J.: *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*. Springer, Heidelberg (2006)
2. Tsoka, S., Ouzounis, C.: Recent developments and future directions in computational genomics. *FEBS Letters* 480(1), 42–48 (2000)
3. Lander, E.S., et al.: Initial sequencing and analysis of the human genome. *Nature* 409, 860–921 (2001)

4. Rubnitz, J.E., Hijiya, N., Zhou, Y., Hancock, M.L., Rivera, G.K., Pui, C.: Lack of benefit of early detection of relapse after completion of therapy for acute lymphoblastic leukemia. *Pediatric Blood & Cancer* 44(2), 138–141 (2005)
5. Affymetrix, GeneChip Human Exon 1.0 ST Array, <http://www.affymetrix.com/products/arrays/specific/Exon.affx>
6. Quackenbush, J.: Computational analysis of microarray data. *Nature Review Genetics* 2(6), 418–427 (2001)
7. Lipshutz, R.J., Fodor, S.P.A., Gingeras, T.R., Lockhart, D.H.: High density synthetic oligonucleotide arrays. *Nature Genetics* 21, 20–24 (1999)
8. Kolodner, J.: *Case-Based Reasoning*. Morgan Kaufmann, San Francisco (1993)
9. Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U., Speed, T.P.: Exploration, Normalization, and Summaries of High density Oligonucleotide Array Probe Level Data. *Biostatistics* 4, 249–264 (2003)
10. Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 59–69 (1982)
11. Fritzke, B.: A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems* 7, 625–632 (1995)
12. Martinetz, T.: Competitive Hebbian learning rule forms perfectly topology preserving maps. In: *ICANN 1993: International Conference on Artificial Neural Networks*, pp. 427–434 (1993)
13. Martinetz, T., Schulten, K.: A neural-gas network learns topologies. In: Kohonen, T., et al. (eds.) *Artificial Neural Networks*, Amsterdam, pp. 397–402 (1991)
14. Brunelli, R.: Histogram Analysis for Image Retrieval. *Pattern Recognition* 34, 1625–1637 (2001)
15. Garipey, R., Pepe, W.D.: On the Level sets of a Distance Function in a Minkowski Space. *Proceedings of the American Mathematical Society* 31(1), 255–259 (1972)
16. Jolliffe, I.: *Principal Component Analysis*, 2nd edn. Springer Series in Statistics (2002)
17. Riverola, F., Díaz, F., Corchado, J.M.: Gene-CBR: a case-based reasoning tool for cancer diagnosis using microarray datasets. *Computational Intelligence* 22, 254–268 (2006)
18. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York (1990)
19. Saitou, N., Nie, M.: The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol.* 4, 406–425 (1987)
20. Sneath, P.H.A., Sokal, R.R.: *Numerical Taxonomy. The Principles and Practice of Numerical Classification*. W.H. Freeman Company, San Francisco (1973)
21. Fix, E., Hodges, J.L.: Discriminatory analysis, nonparametric discrimination consistency properties, Technical Report 4, United States Air Force, Randolph Field, TX (1951)

Crosstalk and Signalling Pathway Complexity - A Case Study on Synthetic Models

Zheng Rong Yang

School of Biosciences, University of Exeter, UK

Abstract. Crosstalk between signalling pathways have been intensively studied in wet laboratory experiments. More and more experimental evidences show that crosstalk is a very important component for maintaining biology systems robustness. In wet laboratory experiments, crosstalk are normally predicted by applying specific stimulus, i.e., various extra-cellular cues or individual gene suppressors or protein inhibitors. If significant difference between a control group without a specific stimulus and an experimental group with a specific stimulus is found, crosstalk is predicted. In terms of time complexity and cost, such experiments are commonly limited to small scales by using very few sampling time points. At the same time, few mathematical models have been proposed to analyse or predict crosstalk. This work investigates how crosstalk affects signalling pathway complexity and if such effect is significant for discrimination purpose, hence providing evidence for crosstalk prediction. Two crosstalk activities (positive and negative) based on simple synthetic transcription models are used for study. The study has found that crosstalk can change the steady-state gene expression order, hence making signalling pathway complex. The finding indicates that crosstalk is predictable using computer programs in top-down systems biology research.

Keywords: Crosstalk, signalling pathways, transcription, degradation, gene expression order, differential equations, steady-state analysis.

1 Introduction

Over the past decades, many signalling molecules (genes, proteins, lipids and ions) have been identified and the way through which they communicate via signalling pathways have been elucidated. Most signalling pathways are triggered by extra-cellular cues. Each signalling pathway is a cascade chain of a number of genes or proteins. An upstream gene or protein triggers a domino effect on the subsequent downstream genes or proteins. Proteins in a pathway can be physically and chemically modified; e.g. covalent modifications (phosphorylation), recruitment, allosteric activation or inhibition and binding of proteins [1]. A gene can be converted to a transcript and a transcript can be broken down into its constitutive parts for future production of RNA. The former is called a transcription process and the latter is a degradation process. It has become

more and more apparent that signalling does not necessarily occur separately in a linear way, but rather through a large and complex network of interacting signalling networks [2]. It is now understood that inter-pathway crosstalk can reflect underlying complexity within a cellular signalling network causing the output of a signalling pathway to depend non-linearly on the extra-cellular cues, hence being robust [3]. Crosstalk is generally described in biochemistry and molecular biology as unwanted communication between signalling pathways or signalling molecules. The term encompasses positive and negative signalling, layered changes in gene expression and feedback between signalling proteins [4]. Crosstalk is sometimes treated as specific interactions between genes or proteins of multiple signalling pathways or shared components between pathways. While to some researchers crosstalk is unwanted or insignificant exchange of information and specific crosstalk between signalling components are exciting but apparently rare, many biological experiments have documented and confirmed that inhibiting one signalling pathway may have a positive or negative influence on other signalling molecules in other pathways. This reinforces the view that specificity of biological responses is largely generated by the combinatorial integration of pathway crosstalk and the versatility of component functions [5]. Since 1960, biological studies of crosstalk have increased exponentially suggesting that crosstalk is an important phenomenon in cell signalling. There have been two main streams in researching into signalling pathways. The first is to use mathematical modelling approaches like differential equations systems to study systems dynamics [6]. With this type of methods, we can study deterministic mechanism of signalling pathways as well as crosstalk. In most situations, model parameters needed by differential equations are unknown or partially unknown. For instance, in a transcription network, we may have little knowledge about transcription and degradation rates which are fundamental for modelling. In a transduction network, phosphorylation and de-phosphorylation rates may not be precisely known as well. Few computational or mathematical crosstalk models are practically useful so far. The second is to use graphic models or the Bayesian network approach [7] to study the qualitative relationship between signalling molecules. In terms of the reality, the second methods look like better fitting to top-down systems biology research. However, its limitation is the qualitative nature.

Having had a large gene transcription/expression data set on hand, we may need to investigate why certain genes are differentially expressed and why other part of genes are not. It is very often observed that genes from a number of different pathways are differentially expressed simultaneously although only one specific individual signalling pathway is activated under a specific treatment. Biological systems are commonly evolved for better fitting to the environment. Using multiple signalling pathways for one specific task or using one signalling molecule for multiple functions are the important facts for biological systems robustness, hence complexity. In terms of this, it is very important to understand how multiple signalling pathways are combined together for effective and efficient cellular functioning. It is therefore motivated in this study to

investigate if crosstalk is a major component of signalling pathway complexity and how crosstalk plays its role in signalling pathway complexity. The study uses simple synthetic transcription networks with two signalling pathways. Two crosstalk activities (promotion and inhibition) are studied. It has been found in this study that crosstalk is indeed a major component for signalling pathway complexity because it changes the steady-state gene expression order. This finding is therefore useful for crosstalk prediction.

2 Synthetic Transcription Pathway Models

Fig 1 shows three types of synthetic transcription networks, one being a non-crosstalk model and two being crosstalk models. The left panel shows a non-crosstalk model, the middle panel illustrates a positive crosstalk model and the right panel gives a negative crosstalk model. In a positive crosstalk model, the expression of one gene of one signalling pathway may promote the expressions of some genes of the other signalling pathways. In a negative crosstalk model, the expression of one gene of one signalling pathway may suppress the expressions of some genes of the other signalling pathways.

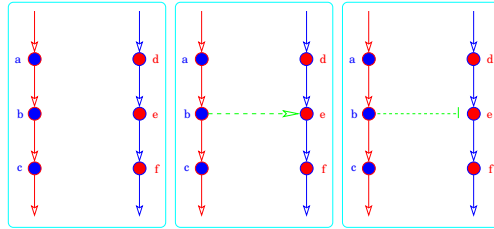


Fig. 1. Synthetic crosstalk models

3 Non-crosstalk Model

If there is no crosstalk between two signalling pathways (the left panel in Fig. 1), the genes of two signalling pathways will be expressed (transcribed) in a linear manner when the extra-cellular cues are applied to two individual signalling pathways separately. Suppose the total numbers of six genes are $N_i \in \mathcal{I}, \forall i \in \{a, b, c, d, e, f\}$. The expression levels (transcripts) of six genes are denoted as integers $0 \leq X_i \leq N_i, \forall i \in \{a, b, c, d, e, f\}$. The system can be expressed by six differential equations as below,

$$\begin{cases} \dot{X}_a = \alpha_a R_A (1 - \frac{X_a}{N_a}) - \beta_a X_a \\ \dot{X}_b = \alpha_b X_a (1 - \frac{X_b}{N_b}) - \beta_b X_b \\ \dot{X}_c = \alpha_c X_b (1 - \frac{X_c}{N_c}) - \beta_c X_c \end{cases} \quad \text{and} \quad \begin{cases} \dot{X}_d = \alpha_d R_B (1 - \frac{X_d}{N_d}) - \beta_d X_d \\ \dot{X}_e = \alpha_e X_d (1 - \frac{X_e}{N_e}) - \beta_e X_e \\ \dot{X}_f = \alpha_f X_e (1 - \frac{X_f}{N_f}) - \beta_f X_f \end{cases} \quad (1)$$

Here $\alpha_i \in \mathcal{R}$ and $\beta_i \in \mathcal{R} (\forall i \in \{a, b, c, d, e, f\})$ are the transcription and degradation rates, respectively while $R_A \in \mathcal{I}$ and $R_B \in \mathcal{I}$ are the extra-cellular signal

(ECS) applied to two individual signalling pathways, respectively. Suppose we use ratios as real numbers for system modelling $u_i = \frac{X_i}{N_i} \in \mathcal{R}$, $\pi_A = \frac{R_A}{N_A} \in \mathcal{R}$ and $\pi_B = \frac{R_B}{N_B} \in \mathcal{R}$ (N_A and N_B are the maximum numbers of receptors for two pathways), and assume that $\alpha_i = \alpha$, $\beta_i = \beta$ ($\forall i \in \{a, b, c, d, e, f\}$), we then have a normalised non-crosstalk model as below

$$\begin{cases} \dot{u}_a = \alpha\pi_A(1 - u_a) - \beta u_a \\ \dot{u}_b = \alpha u_a(1 - u_b) - \beta u_b \\ \dot{u}_c = \alpha u_b(1 - u_c) - \beta u_c \end{cases} \quad \text{and} \quad \begin{cases} \dot{u}_d = \alpha\pi_B(1 - u_d) - \beta u_d \\ \dot{u}_e = \alpha u_d(1 - u_e) - \beta u_e \\ \dot{u}_f = \alpha u_e(1 - u_f) - \beta u_f \end{cases} \quad (2)$$

In this model, the steady state is expressed as below

$$\begin{cases} u_a^* = \frac{\alpha\pi_A}{\alpha\pi_A + \beta} \\ u_b^* = \frac{\alpha u_a^*}{\alpha u_a^* + \beta} \\ u_c^* = \frac{\alpha u_b^*}{\alpha u_b^* + \beta} \end{cases} \quad \text{and} \quad \begin{cases} u_d^* = \frac{\alpha\pi_B}{\alpha\pi_B + \beta} \\ u_e^* = \frac{\alpha u_d^*}{\alpha u_d^* + \beta} \\ u_f^* = \frac{\alpha u_e^*}{\alpha u_e^* + \beta} \end{cases} \quad (3)$$

3.1 Steady-State Gene Expression Order (SEO)

The steady-state gene expression order is studied here because the final gene expression values are commonly the target for various analyses. The gene expression order at steady state has been found being a very important information for predicting signalling complexity, hence crosstalk. Two steady-state gene expression orders are introduced, one being within one pathway referred to as SEO_w and one being cross pathways SEO_c.

We define that a simple signalling pathway network will have its genes' expressions at steady-state in order in terms of these genes' positions in the pathways and ECS applied. Any change of such an order indicates some inherent or inserted complexity.

SEO_w: There are two types of within pathway steady-state gene expression orders, one for genes in a pathway with a small extra-cellular signal (ECS) and the other a large ECS. We will see below that SEO_w will show a different pattern when using either a large ECS or a small ECS for activating pathways. When a large ECS is applied to activate a pathway, we can certainly ensure that an upstream gene has a higher steady-state expression than its downstream gene. Taking pathway *A* in the left panel in Fig. 1 as an example. If pathway *A* has a large ECS (π_A) as $\pi_A \geq 1 - \frac{\beta}{\alpha}$. Note that $\beta < \alpha$. We can have $u_a^* \geq 1 - \frac{\beta}{\alpha}$, hence $u_a^* \geq u_b^*$ and $u_b^* \geq u_c^*$. We refer to this as a large signal SEO_w. If ECS (π_A) is small, an upstream gene will have a lower steady-state expression than its downstream gene. For instance, if $\pi_A \leq 1 - \frac{\beta}{\alpha}$, we then have $u_a^* \leq 1 - \frac{\beta}{\alpha}$, hence $u_a^* \leq u_b^*$ and $u_b^* \leq u_c^*$.

SEO_c: If $\pi_A \gg \pi_B$, it is not very difficult to have $u_a^* > u_d^*$, and hence $u_b^* > u_e^*$ as well $u_c^* > u_f^*$. If $\pi_A \ll \pi_B$, we will have similar simple result. This shows that genes in a pathway with a larger ECS will have higher steady-state expression than genes in a pathway with a smaller ECS.

In conclusion, the signalling activity is in a linear cascade when no crosstalk between two pathways is present. In a linear pathway network, SEOs give a

nearly universal rule for observing the order of gene expressions in terms of the positions of genes in pathways. The SEOs are therefore a good indicator for predicting crosstalk in signalling pathways.

3.2 Simulation Result

We now conduct computer simulations to see if the results meet what we have analysed. We set $\alpha = 0.5$ and $\beta = 0.1$. We constrain two ECSs as $\pi_A + \pi_B = 1$. Each computer simulation starts from zero and ends at 30. From Fig. 2 (a), we can see that SEOW among genes in a pathway is maintained if the pathway is highly activated and no crosstalk is present. The two panels in Fig. 2 (a) provide this evidence. For instance, genes *a*, *b* and *c* of pathway *A* have a clear SEOW that $u_a^* > u_b^* > u_c^*$ (the left panel in Fig. 2 (a), where $\pi_A = 0.9$ and $\pi_B = 0.1$) and the genes *d*, *e* and *f* of pathway *A* have a clear SEOW that $u_d^* > u_e^* > u_f^*$ (the right panel in Fig. 2 (a), where $\pi_A = 0.1$ and $\pi_B = 0.9$). Importantly, we can see from both panels that if a gene of one pathway has a higher expression rate than a gene of the other pathway, their downstream genes will maintain this relationship (for SEOW). We then analyse the steady-state gene expression distribution. In order to analyse this, we have randomly generated 20 repeats for each gene. Each randomly generated gene will have a small Gaussian noise ($\mathcal{G}(0, 0.01)$) added in computer simulation. Two sampling time points are used for a two-dimensional space visualisation. The two sampling time points are 15 ('half time') and 30 ('full time'). Fig. 2 (b) shows the gene expression distributions for two treatments, where clusters with gray colour are for genes of pathway *B* and clusters with dark colour are for genes of pathway *A*. It can be seen that two classes of genes (two pathways *A* and *B*) can obviously be separated using a straight line showing that they demonstrate a linear relationship. We now analyse how steady-state gene expressions vary with the values of π_A and π_B , which is referred to as the sensitivity. Fig. 3 (a) shows such a sensitivity visualisation. It is clearly to see that the steady-state gene expressions of two

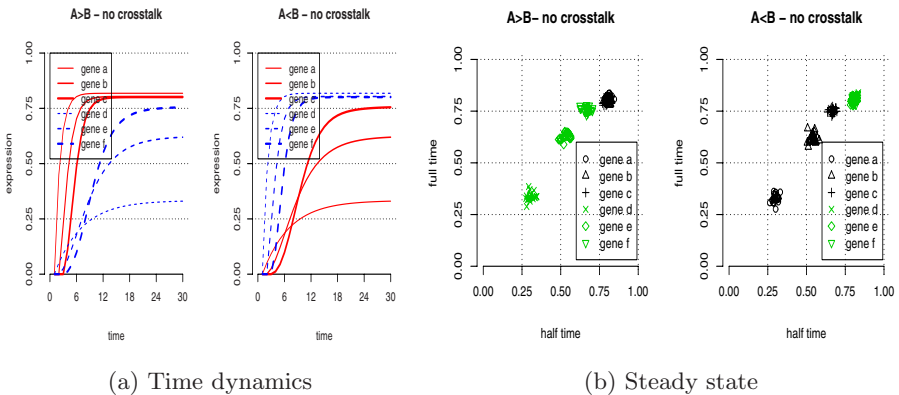
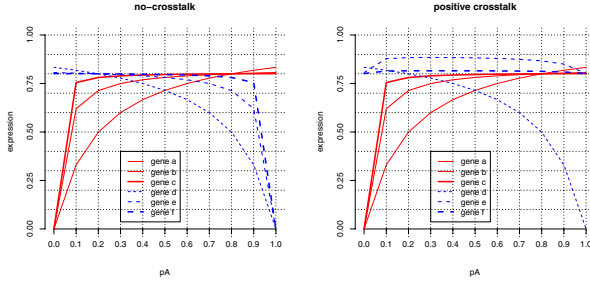


Fig. 2. Non-crosstalk model



(a) non-crosstalk model (b) positive crosstalk model

Fig. 3. Dynamics pattern sensitivity

pathways are symmetrical. On the left side, genes of pathway *B* have higher expressions than genes of pathway *A* while on the right side, genes of pathway *A* have higher expressions than genes of pathway *B*.

4 Positive Crosstalk Model

The structure of a positive crosstalk model is shown in the middle panel in Fig. 1, where we name pathway *A* as the master pathway and pathway *B* as the slave pathway. The model parameters are $\alpha = 0.5$, $\beta = 0.1$ and $\gamma_{b \rightarrow e} = 0.5$. In this model, we consider the positive crosstalk from gene *b* of pathway *A* to gene *e* of pathway *B*, i.e., the expression level of gene *e* is controlled by two genes, one being its upstream gene *c* and the other being the crosstalk gene *b*. The mathematical model is shown as below. It can be seen that the difference between this differential equations system and the differential equations system defined in Eqs. (2) is the addition of a positive crosstalk element for gene *e* ($\gamma_{b \rightarrow e} u_b (1 - u_e)$). The crosstalk is implemented by the crosstalk rate, i.e., the rate that gene *b* interacts with gene *e* for signal communication. We assume that the transcripts after crosstalk will stay as they are and will not be broken into its constitutive parts

$$\begin{cases} \dot{u}_a = \alpha\pi_A(1 - u_a) - \beta u_a \\ \dot{u}_b = \alpha u_a(1 - u_b) - \beta u_b \\ \dot{u}_c = \alpha u_b(1 - u_c) - \beta u_c \end{cases} \quad \text{and} \quad \begin{cases} \dot{u}_d = \alpha\pi_B(1 - u_d) - \beta u_d \\ \dot{u}_e = \alpha u_d(1 - u_e) - \beta u_e + \gamma_{b \rightarrow e} u_b(1 - u_e) \\ \dot{u}_f = \alpha u_e(1 - u_f) - \beta u_f \end{cases} \tag{4}$$

In this situation, the steady state is expressed as

$$\begin{cases} u_a^* = \frac{\alpha\pi_A}{\alpha\pi_A + \beta} \\ u_b^* = \frac{\alpha u_a^*}{\alpha u_a^* + \beta} \\ u_c^* = \frac{\alpha u_b^*}{\alpha u_b^* + \beta} \end{cases} \quad \text{and} \quad \begin{cases} u_d^* = \frac{\alpha\pi_B}{\alpha\pi_B + \beta} \\ u_e^* = \frac{\alpha u_d^* + \gamma_{b \rightarrow e} u_b^*}{\alpha u_d^* + \beta + \gamma_{b \rightarrow e} u_b^*} \\ u_f^* = \frac{\alpha u_e^*}{\alpha u_e^* + \beta} \end{cases} \tag{5}$$

It has been analysed above that both SEOW and SEOC are maintained in a non-crosstalk model. However, they may not be maintained in a crosstalk model.

SEOW: We analyse pathway B here because it is the target for analysis. Moreover, only a model with a large signal applied to the master pathway is analysed because the effect of the positive crosstalk is limited when the master pathway is not well activated, i.e., its positive crosstalk effect is limited ($\pi_A \ll \pi_B$). If $\pi_B + \frac{\beta}{\alpha} \ll 1$, it is almost true that $1 - \pi_B - \frac{\beta}{\alpha} > 0$. Let $\vartheta = \frac{\beta}{\alpha}$, we have $\pi_B + \vartheta - \pi_B - \pi_B \vartheta - \vartheta^2 > 0$ and $1 - \frac{\pi_B}{\pi_B + \vartheta} - \vartheta > 0$. From this and Eq. (5), we can see that $1 - u_d^* - \vartheta > 0$. It is then not difficult to see that $u_d^*(1 - u_d^* - \vartheta) + u_b^*(1 - u_d^*) > 0$ or $u_d^* + u_b^* > (u_d^*)^2 + u_d^*u_b^* + u_d^*\vartheta$. From Eq. (5), $u_e^* > u_d^*$ is derived. It can be seen that SEOW is not maintained because of the crosstalk between two pathways.

SEOC: We now need to see if SEOC maintains when crosstalk is present, i.e., we need to prove if $u_b^* > u_e^*$ when $u_a^* > u_d^*$ (or $\pi_A \gg \pi_B$). For the model with $\pi_A \ll \pi_B$, the effect of crosstalk is limited as mentioned above and is not analysed here. We assume $u_b^* < u_e^*$, i.e., $u_b^* < \frac{u_d^* + u_b^*}{u_d^* + u_b^* + \vartheta}$. It is not difficult to see that $\frac{u_d^* + u_b^*}{u_d^* + u_b^* + \vartheta} > \frac{u_b^*}{u_b^* + \vartheta}$. If $u_b^* < \frac{u_b^*}{u_b^* + \vartheta}$, we can have $u_b^* < 1 - \vartheta$ or $\frac{\pi_A}{\pi_A + \pi_A \vartheta + \vartheta^2} < 1 - \vartheta$. This indicates that $u_b^* < \frac{u_b^*}{u_b^* + \vartheta}$ when $\pi_A < 1 - \vartheta$. When $\vartheta = 0.2$, we can see that SEOC is not maintained if $0 < \pi_A < 0.8$. Because we had an approximation $\frac{u_d^* + u_b^*}{u_d^* + u_b^* + \vartheta} > \frac{u_b^*}{u_b^* + \vartheta}$ above, the condition $\pi_A < 0.8$ can be much more relaxed.

4.1 Simulation Result

We now add a positive crosstalk from gene b to gene e in the network as shown in the middle panel of Fig. 4, where we have three genes for each pathway (A and B). Fig. 4 (a) shows the time dynamics of two treatments ($\pi_A \gg \pi_B$ and $\pi_A \ll \pi_B$). It can be seen that both SEOW and SEOC are not maintained. The left panel in Fig. 4 (a) shows the outcome when pathway A (the master pathway) is highly activated while pathway B is weakly activated, i.e., $\pi_A = 0.9 > \pi_B = 0.1$. It can be seen that SEOW in pathway A is maintained as $u_a^* > u_b^* > u_c^*$, but SEOW in pathway B is changed to $u_e^* > u_f^* > u_d^*$. Importantly, $u_e^* > u_b^*$ although $u_a^* > u_d^*$. Although gene e is a downstream gene of gene d with low activity, it still has a high expression because gene b has a positive crosstalk with it.

The right panel in Fig. 4 (a) demonstrates the result when pathway B is highly activated while pathway A (the master pathway) is weakly activated, i.e., $\pi_B = 0.9 > \pi_A = 0.1$. $u_e^* > u_d^*$ still maintains because of crosstalk although the activation of pathway A is much smaller. In the left panel, we can see that SEOW has been changed from $u_f^* > u_e^* > u_d^*$ (see Fig. 2 (a)) to $u_e^* > u_f^* > u_d^*$ while in the right panel, we can see that SEOW has changed from $u_d^* > u_e^* > u_f^*$ (see Fig. 2 (a)) to $u_e^* > u_f^* > u_d^*$. All are as expected in the theoretical analysis above.

In order to analyse gene expression distribution in a low dimensional space as in the non-crosstalk model, we still use random repeats, i.e., generating 20 repeats each being added a Gaussian noise ($\mathcal{G}(0, 0.01)$). Two sampling time points

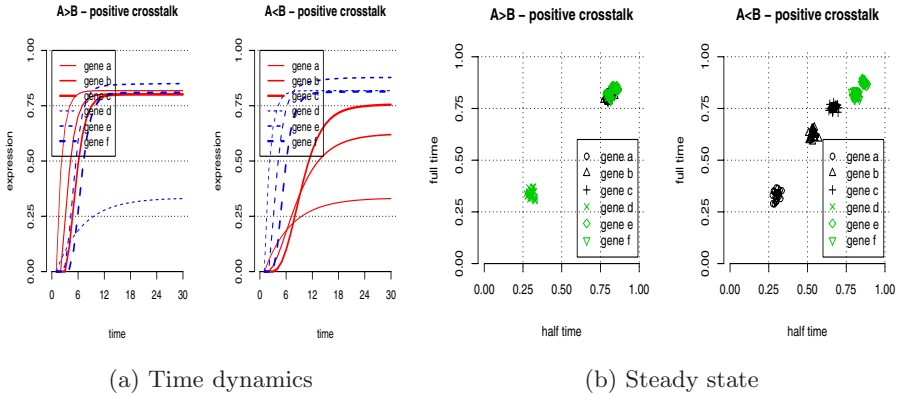


Fig. 4. Positive crosstalk model

(15 and 30) are used for a two-dimensional space visualisation. In Fig. 2 (b), we have seen that there is a linear relationship between two classes (pathways) of gene clusters, i.e., a straight line can be easily found to separate two classes of gene clusters. However, this linear relationship does not exist when crosstalk between two pathways is present, see the left panel in Fig. 4 (b). When pathway A has a high activation, the clusters of gene e and gene f have moved beyond the clusters of genes of pathway A. It can be seen that it is now difficult to find a linear function to well separate two classes (pathways) of gene clusters. This is the evidence that crosstalk makes signalling network complex. However, in the right panel of Fig. 4 (a), we can see that the expression distributions of three gene clusters of pathway A have generally no difference from those in the right panel in Fig. 2 (b). However, the expression distribution of the cluster of gene e has been elevated. This is because of the positive crosstalk. Although pathway A (the master pathway) is weakly activated, a small signal from gene b can still make a difference on gene e of pathway B. Fig. 3 (b) shows the steady-state gene expression sensitivity in comparison with Fig. 3 (a). It can be seen here that gene e and gene f are almost always highly expressed whatever how to change two ECSs for two pathways. This is because they are all affected (directly and indirectly) by crosstalk with gene b of pathway A whatever its signal is small or large. It can also be seen that gene e (the second gene in pathway B) nearly has a higher expression compared with gene b (the second gene in pathway A corresponding to gene e in pathway B) all the times while gene f (the third gene in pathway B) has a high expression most times compared with gene c (the third gene in pathway A corresponding to gene f).

5 Negative Crosstalk Model

A negative crosstalk model structure is seen in the right panel of Fig. 1. The model parameters are $\alpha = 0.5$, $\beta = 0.1$ and $\gamma_{b \rightarrow e} = 0.5$. Six differential equations

for this model are defined as below. The difference between this differential equations system and the differential equations system in Eqs. 2 is a negative crosstalk element for gene e ($-\gamma_{b \rightarrow e} u_b u_e$). Note that this is also different from the positive crosstalk model, where the additive element is $\gamma_{b \rightarrow e} u_b (1 - u_e)$. This is based on the assumption that in a positive crosstalk model, a transcript of one pathway promotes the expression of a gene of the other pathway. This is why we assume the interaction between the transcript of gene b (u_b) and gene e ($1 - u_e$). However, a negative crosstalk is assumed here to be implemented by the effect that a transcript of one pathway interacts with a transcript of the other pathway leading to the increase of degradation. The crosstalk is also implemented by the crosstalk rate ($\gamma_{b \rightarrow e}$), i.e., the rate that the transcript of gene b interacts with the transcript of gene e .

$$\begin{cases} \dot{u}_a = \alpha\pi_A(1 - u_a) - \beta u_a \\ \dot{u}_b = \alpha u_a(1 - u_b) - \beta u_b \\ \dot{u}_c = \alpha u_b(1 - u_c) - \beta u_c \end{cases} \quad \text{and} \quad \begin{cases} \dot{u}_d = \alpha\pi_B(1 - u_d) - \beta u_d \\ \dot{u}_e = \alpha u_d(1 - u_e) - \beta u_e - \gamma_{b \rightarrow e} u_b u_e \\ \dot{u}_f = \alpha u_e(1 - u_f) - \beta u_f \end{cases} \quad (6)$$

In this situation, the steady state is expressed as

$$\begin{cases} u_a^* = \frac{\alpha\pi_A}{\alpha\pi_A + \beta} \\ u_b^* = \frac{\alpha u_a^*}{\alpha u_a^* + \beta} \\ u_c^* = \frac{\alpha u_b^*}{\alpha u_b^* + \beta} \end{cases} \quad \text{and} \quad \begin{cases} u_d^* = \frac{\alpha\pi_B}{\alpha\pi_B + \beta} \\ u_e^* = \frac{\alpha u_d^*}{\alpha u_d^* + \beta + \gamma_{b \rightarrow e} u_b^*} \\ u_f^* = \frac{\alpha u_e^*}{\alpha u_e^* + \beta} \end{cases} \quad (7)$$

SE0w: We analyse pathway B because of the same reason mentioned above. Suppose $u_f^* > u_e^*$ which violates SEOW, we can have $u_e^* < 1 - \vartheta$ or $\frac{u_d^*}{u_d^* + u_b^* + \vartheta} < 1 - \vartheta$. With some algebraic operations, we have $\left(\frac{\pi_A}{\pi_A + \pi_A \vartheta + \vartheta^2} + \vartheta\right) (1 - \vartheta) > \frac{\pi_B}{\pi_B + \vartheta}$ or $(1 - \vartheta - \vartheta^2 - \vartheta^3)\pi_A^2 + (\vartheta^2 + \vartheta^3 - 1)\pi_A + \vartheta^5 < 0$. We have the corresponding quadratic equation as $(1 - \vartheta - \vartheta^2 - \vartheta^3)\pi_A^2 + (\vartheta^2 + \vartheta^3 - 1)\pi_A + \vartheta^5 = 0$. Given $\vartheta = 0.2$, it can be seen that two solutions for the quadratic equation are 0.0002 and 1.263. This means that when $\pi_A \in [0.0002, 1.263]$, the inequality defined above is satisfied meaning that $u_f^* > u_e^*$. SEOW is therefore not maintained in this crosstalk model.

SE0c: If $\pi_A \ll \pi_B$, we will certainly have $u_a^* < u_d^*$. We only analysis this is because the crosstalk effect is limited when $\pi_A \gg \pi_B$. This is because there will be nothing to decrease when genes of the slave pathway are silent. We now need to see if the relationship $u_b^* < u_e^*$ can be maintained. Suppose we assume that this relationship may be changed, the inequality is expressed as below $u_b^* > u_e^* = \frac{u_d^*}{u_d^* + u_b^* + \vartheta}$, where $\vartheta = \frac{\beta}{\alpha}$. It is not difficult to prove the following inequality $\frac{1}{1 + u_b^* + \vartheta} > \frac{u_d^*}{u_d^* + u_b^* + \vartheta}$. From this, we are working on a simpler inequality, $u_b^* > \frac{1}{1 + u_b^* + \vartheta} > u_e^*$. What we need to do now is to find if the first pair of the above inequality is true and what is the condition for it to be true. We replace u_b^* using x and then have $x^2 + (1 + \vartheta)x - 1 > 0$. For this inequality, we need to find if the solutions exist for the corresponding quadratic equation $x^2 + (1 + \vartheta)x - 1 = 0$.

For this quadratic equation, we find one real solution as $x = \frac{-(1+\vartheta) + \sqrt{(1+\vartheta)^2 + 4}}{2}$. If $\beta = 0.1$ and $\alpha = 0.5$, we can find that $x = 0.566$. Because $u_b^* = \frac{u_a^*}{u_a^* + \vartheta}$ (see Eq. (7)), we have $\pi_A = 0.07$. This means that if $\pi_A > 0.07$ we can ensure that $u_b^* > u_c^*$. It is not difficult to find $u_c^* > u_f^*$ as well. It can be seen that SEOC is not maintained in a negative crosstalk model when π_A exceeds a small critical value (0.07 here). Non-linearity has therefore been induced. Limited by page space, no computer simulation result can be given, but is as expected.

6 Conclusion

This paper has presented a study on how crosstalk affects signalling network complexity. The analysis is based on simple transcription networks with two signalling pathways and six genes. Through theoretical analyses and computer simulations, it has been confirmed that crosstalk makes signalling pathway complex through changing the steady-state gene expression orders. The study shows that crosstalk is therefore predictable if some biological knowledge regarding pathway information, pathway categories of genes and experimental data are available.

References

1. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P.: Cell Communications. In: *Molecular Biology of the Cell*. Garland Science, New York, pp. 831–890 (2002)
2. Weng, G., Bhalla, U.S., Iyengar, R.: Complexity in Biological Signalling Systems. *Science* 284, 92–96 (1999)
3. Ferrell Jr., J.E.: Tripping the switch fantastic: how a protein kinase cascade can convert graded inputs into switch-like outputs. *Trends. Biochem. Sci.* 21, 460–466 (1996)
4. Taylor, J.E., McAnish, M.R.: Signalling crosstalk in plants: emerging issues. *J. Exp. Bot.* 55, 147–149 (2003)
5. Kolch, W.: Meaningful relationships: the regulation of the Ras/Raf/MEK/ERK pathway by protein interactions. *Biochem. J.* 351, 289–305 (2000)
6. Cho, K.H., Wolkenhauer, O.: Analysis and modelling of signal transduction pathways in systems biology. *Biochem. Soc. Trans.* 31, 1503–1509 (2003)
7. Woolf, P.J., Prudhomme, W., Daheron, L., Daley, G.O., Lauffenburger, D.A.: Bayesian analysis of signaling networks governing embryonic stem cell fate decisions. *Bioinformatics* 21, 741–753 (2005)

Explore Residue Significance in Peptide Classification

Zheng Rong Yang

School of Biosciences, University of Exeter, UK

Abstract. Although peptide classification has been studied for a few decades, a proper method for studying residue significance has not yet been paid much attention. This paper introduces a novel neural learning algorithm which can be used to reveal residue significance for discriminating between functional and non-functional peptides and for peptide conformation pattern analysis. The algorithm is a revised bio-basis function neural network which was introduced a few years ago.

Keywords: Peptide classification, residue significance, neural learning, bio-basis function.

1 Introduction

One of the most important issues in biology is to annotate function for a novel sequence. From this, biologists are able to study how proteins are functioning in cells. There are two types of approaches for protein function annotation, i.e. using global and local information for model construction. A global protein function annotation approach is normally dealing with whole sequence homology alignment. For instance, the Seller's algorithm [1], the Needleman-Wunsch algorithm [2], and the Smith-Waterman algorithm [3] are three typical algorithms for pairwise sequence homology alignment. Among two sequences, one has experimentally confirmed functions while the other is under investigation. If the homology alignment is able to demonstrate a large alignment score as the similarity, two sequences are believed evolving from the same ancestor, being either orthologous or paralogous. It is then believed that the protein function should be conserved through evolution. Because pairwise homology alignment algorithms are commonly slow and inefficient. Database search tools have been developed and have been widely used. The most commonly used database search tools are FASTA [4] and BLAST [5]. A database search tool will use a data bank of protein sequences with functions experimentally confirmed as a support. The search will find a set of database sequences which have high homology alignment scores with a novel sequence. Based on these database sequences, it is then possible to annotate functions for a novel protein. Using this type of approaches, we can have a whole picture about how a novel sequence is similar to one or a number of known sequences. Because segments which involve critical protein functions will not mutate dramatically, it is very easy to visualise how these conserved sequence segments are similar to a novel sequence.

The second type of approaches is focused on only local sequence segments (normally conserved areas) for function annotation/prediction. The theory behind is that proteins whenever they bind will be functional. Such a binding normally does not need the whole structures of two proteins for binding affinity recognition. This means that two proteins will bind for biochemical reaction if they have good binding surface. Because of this, peptide classification studies how a sequence segment of a protein has a relationship with various binding.

Binding can have two types of biochemical reactions, protease cleavage activities and post-translational modifications. Both are site specific or substrate specific. However, the former will involve two neighbouring residues to “cut” a poly-protein into two functional proteins while the latter will involve one residue for attaching a chemical compound such as phosphor to a protein. Such an attachment will lead to signal transduction, i.e. proteins with and without the attachment will bring different signals for different biochemical functions.

The earliest work on peptide classification was based on frequency estimate. For example, the h function [6], where the frequency of 20 amino acids at each residue is calculated from a set of functional peptides. The estimated frequencies are then stored in a computer program for prediction. The major shortcoming of this method is that they usually have a high sensitivity and a low specificity. Some statistical models like hidden Markov models (HMM) [7], discriminant analysis [8] and quadratic discriminant analysis [9] have also been used for data mining protein peptides. However, a HMM model also has a high sensitivity and a low specificity [10].

Neural networks and the support vector machine [11] have been applied to data mining protein peptides as well. For instance, neural networks have been used in signal peptide cleavage site prediction [13], glycoproteins linkage site prediction [12], enzyme active site prediction [14], phosphorylation site prediction [15], and water active site prediction [16]. The support vector machine has been used for the prediction of translation initiation sites [17], the prediction of phosphorylation sites [18], the prediction of T-cell receptor [19], and the prediction of protein-protein interactions [20].

In dealing with amino acids, the orthogonal coding method where each amino acid is coded using a 20-bit long orthogonal binary vector is normally used [21]. Although it has been widely used for various protein peptide modelling tasks, it may not well code biological information in peptides. The bio-basis function neural network was therefore developed for proper coding of amino acids in 2003 [22]. The bio-basis function neural network has been successfully used for many peptide classification problems.

However, the original bio-basis function has a major limit in data mining peptide data for peptide classification, i.e. being unable to reveal residue significance in a peptide classification application. If residue significance can be explored quantitatively, the information will be helpful in understanding how residues in peptides are contributing to protein binding.

2 The Original and the Revised Bio-basis Functions

Denote by \mathbf{q} an input (query) peptide and \mathbf{s}_n a support peptide ($1 \leq n \leq \ell$, $\|\mathbf{q}\| = \|\mathbf{s}_n\| = D$ and D is the number of residues in peptides), the bio-basis function is defined as $\mathbf{q} \mapsto \phi(\mathbf{q}, \mathbf{s}_n)$. The use of this equation can convert a non-numerical input peptide to a numerical input vector $\phi = (1, \phi(\mathbf{q}, \mathbf{s}_1), \phi(\mathbf{q}, \mathbf{s}_2), \dots, \phi(\mathbf{q}, \mathbf{s}_\ell))^T$. To predict if the peptide is functional is determined by the value of a sigmoid function, $y = \frac{1}{1 + \exp(-\mathbf{w} \cdot \phi)}$, where $\mathbf{w} = (w_0, w_1, w_2, \dots, w_\ell)^T \in R^{\ell+1}$ is a vector of model parameters. Note that y is the prediction corresponding to the target variable t . The target variable takes a value 0 for a non-functional peptide and 1 for a functional peptide. The original bio-basis function is defined as

$$\phi(\mathbf{q}, \mathbf{s}_n) = \exp\left(\alpha \frac{\rho(\mathbf{q}, \mathbf{s}_n) - \rho(\mathbf{s}_n, \mathbf{s}_n)}{\rho(\mathbf{s}_n, \mathbf{s}_n)}\right) \quad (1)$$

Here

$$\rho(\mathbf{q}, \mathbf{s}_n) = \sum_{d=1}^D M_{s_{nd} \rightarrow q_d} = \sum_{d=1}^D x_{nd} \quad (2)$$

where $M_{s_{nd} \rightarrow q_d}$ can be found from various mutation matrices [23]. Note that q_d and s_{nd} are the d^{th} residues of \mathbf{q} and \mathbf{s}_n , respectively. The bio-basis function uses the well-developed mutation matrices, hence being capable of well coding biological information in peptides. However, it has a major limit. Residue significance in peptide classification is not well studied. In order to find the residue significance in peptide classification, a novel bio-basis function is proposed in this study. First, a novel bio-basis function is revised as below

$$\phi(\mathbf{q}, \mathbf{s}_n) = \frac{1}{1 + \exp(-\mathbf{c} \cdot \mathbf{x}_n)} \quad (3)$$

Here \mathbf{c} is a residue significance vector and $\mathbf{x}_n = (x_{n1}, x_{n2}, \dots, x_{nD})$ is the n^{th} mutation vector for the n^{th} support peptide ($x_{nd} = M_{s_{nd} \rightarrow q_d}$). There are many other functions available for acting as the kernel function, but the sigmoid function can *squash* the kernel outputs into the interval between zero and one. This well represents the similarity. If the values of \mathbf{c} can be well estimated, the residue significance can be explore.

In real peptide classification applications, there are always two classes of peptides, i.e. being negative and positive peptides. Positive peptides are referring to functional peptides which have conserved patterns for certain amino acids while negative peptides have less conserved patterns. Because of this, two residue significance vectors are introduced. These two vectors are denoted as \mathbf{c}^+ and \mathbf{c}^- . Therefore, there are two types of bio-basis functions as below

$$\phi^+(\mathbf{q}, \mathbf{s}_n) = \frac{1}{1 + \exp(-\mathbf{c}^+ \cdot \mathbf{x}_n)} \quad \text{and} \quad \phi^-(\mathbf{q}, \mathbf{s}_n) = \frac{1}{1 + \exp(-\mathbf{c}^- \cdot \mathbf{x}_n)} \quad (4)$$

The classification of residue significance vectors are based on the property of a support peptide. If a support peptide (\mathbf{s}_n) is a positive peptide, we used \mathbf{c}^+ otherwise \mathbf{c}^- .

3 Maximum Likelihood Training Procedure

The data set is denoted as $\mathcal{D} = \{\mathbf{x}_n, t_n\}_{n=1}^N$, where $t_n \in \{0, 1\}$ is the n^{th} target. We still use the sigmoid function as the model output mentioned as above. The negative log likelihood function of a classification system using above two replacements is defined as below

$$\mathcal{L} = \lambda \vartheta^T \vartheta - \sum_{n=1}^N \{t_n \log y_n + (1 - t_n) \log(1 - y_n)\} \tag{5}$$

Here, $\vartheta = (\mathbf{w}, \mathbf{c})$ is a set of model parameters and $\lambda \vartheta^T \vartheta$ is the regularisation term with λ as the regularisation constant. The first derivative of \mathcal{L} with respect to a weight (peptide coefficient) w_i ($i \in \{1, 2, \dots, \ell\}$) is

$$\frac{\partial \mathcal{L}}{\partial w_i} = \lambda w_i - \sum_{n=1}^N e_n \phi_{ni} \quad \text{or} \quad \nabla \mathcal{L}_{\mathbf{w}} = \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \lambda \mathbf{w} - \mathbf{\Phi}^T \mathbf{e} \tag{6}$$

Here $\phi_{ni} = \phi(\mathbf{q}_n, \mathbf{s}_i)$ is the similarity between a query peptide \mathbf{q}_n and a support peptide \mathbf{s}_i . The stochastic learning rule for the peptide coefficients (\mathbf{w}) is then

$$\Delta \mathbf{w} = -\eta \nabla \mathcal{L}_{\mathbf{w}} \tag{7}$$

where η is a small positive learning rate. We now consider the learning rule for the residue significance vectors \mathbf{c}^+ and \mathbf{c}^- . The first derivative of \mathcal{L} with respect to c_d^+ ($d \in \{1, 2, \dots, D\}$) is

$$\frac{\partial \mathcal{L}}{\partial c_d^+} = \lambda c_d^+ - \sum_{n=1}^N e_n \sum_{\mathbf{s}_k \in \Omega^+} w_k \phi^+(1 - \phi^+) x_{nkd} \quad \text{or} \quad \nabla \mathcal{L}_{\mathbf{c}^+} = \lambda \mathbf{c}^+ - \mathbf{w}^+ \mathbf{\Phi}^+ \mathbf{X}^T \mathbf{e} \tag{8}$$

Here \mathbf{w}^+ is the weight vector connecting to all the positive support peptides, $\mathbf{\Phi}^+$ is the mapping matrix for using all the positive support peptides. Note that the size of \mathbf{w}^+ is ℓ^+ . The learning rule \mathbf{c}^+ is then

$$\Delta \mathbf{c}^+ = -\eta \nabla \mathcal{L}_{\mathbf{c}^+} \tag{9}$$

The first derivative of \mathcal{L} with respect to c_d^- ($d \in \{1, 2, \dots, D\}$) is

$$\frac{\partial \mathcal{L}}{\partial c_d^-} = \lambda c_d^- - \sum_{n=1}^N e_n \sum_{\mathbf{s}_k \in \Omega^-} w_k \phi^-(1 - \phi^-) x_{nkd} \quad \text{or} \quad \nabla \mathcal{L}_{\mathbf{c}^-} = \lambda \mathbf{c}^- - \mathbf{w}^- \mathbf{\Phi}^- \mathbf{X}^T \mathbf{e} \tag{10}$$

Here \mathbf{w}^- is the weight vector connecting all the negative peptides and $|\mathbf{w}^-| = \ell^- = \ell - \ell^+$. $\mathbf{\Phi}^-$ is the mapping matrix using all the negative support peptides. The learning rule \mathbf{c}^- is then

$$\Delta \mathbf{c}^- = -\eta \nabla \mathcal{L}_{\mathbf{c}^-} \tag{11}$$

An iterative learning procedure is employed here for estimating three sets of parameters, i.e. \mathbf{w} , \mathbf{c}^+ , and \mathbf{c}^- .

4 Results

Three data sets are used for the evaluation of the proposed algorithm. They are the HIV-1 protease data [24], Hepatitis C virus protease data [25], and O-linkage data [26]. Ten-fold cross-validation is used. The model accuracy is estimated using the final confusion matrix, i.e. estimate on collective results for all peptides as each peptide will take part in only one testing. The λ value is optimised in the range $\{0.5, 0.2, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001\}$.

4.1 HIV-1 Protease Data

The HIV-1 protease data was published by Cai et al. [24] with 248 negative peptides and 114 positive peptides, each having eight residues. The 8-mer is expressed as $P_4P_3P_2P_1P_1'P_2'P_3'P_4'$ with the cleavage sites located between P_1 and P_1' . Table 1 shows the confusion matrix generated for the data set using 10-fold cross-validation, where λ was optimised to 0.1.

Figure 1 shows the final residue significance distribution. The left panel is for the negative residue significance vector and the right panel is for the positive residue significance. It can be seen that residue P_2' has the largest residue significance value while P_4 has the smallest value all the times. The next two important

Table 1. The confusion matrix for the HIV-1 protease data

	HIV			HCV			O-linkage		
	negative	positive	percent	negative	positive	percent	negative	positive	percent
negative	226	18	91%	726	26	97%	81	31	72%
positive	10	104	91%	16	152	91%	19	171	90%

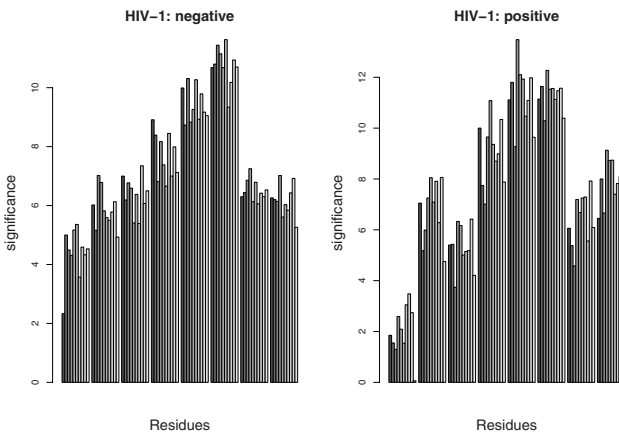


Fig. 1. Residue significance for both negative and positive peptides in the HIV-1 protease data

residues are the cleavage sites (P_1 and $P_{1'}$). The positive residue significance vector has a large variance compared with the negative residue significance vector. This is expected that the positive peptides should have some conserved residues for enzyme activity.

4.2 HCV Protease Data

The HCV protease data was published by Narayanan et al. [25] with 752 negative peptides and 168 positive peptides, each having ten residues. The 10-mer is expressed as $P_6P_5P_4P_3P_2P_1P_{1'}P_{2'}P_{3'}P_{4'}$ with the cleavage sites located between P_1 and $P_{1'}$. Table 1 shows the confusion matrix for the data set, where λ was optimised to 0.5. Figure 2 shows the final residue significance distribution. The left panel is for the negative residue significance vector and the right panel is for the positive residue significance. It can be seen that residues P_1 and $P_{4'}$ have the largest residue significance value while $P_{1'}$ has the smallest value. The positive residue significance vector has a large variance compared with the negative residue significance vector.

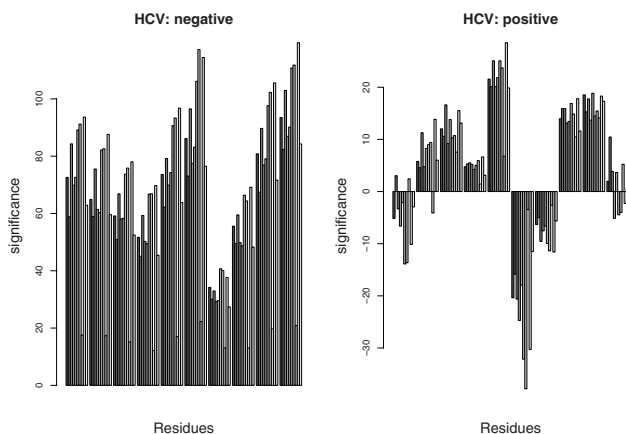


Fig. 2. Residue significance for both negative and positive peptides in the HCV protease data

4.3 O-Linkage Data

The O-linkage protease data was published by Chou [26] with 112 negative peptides and 190 positive peptides, each having nine residues. The 10-mer is expressed as $P_4P_3P_2P_1P_0P_{1'}P_{2'}P_{3'}P_{4'}$ with the post-translation modification site located at P_0 . Table 1 shows the confusion matrix for the data set, where λ was optimised to 0.1. Figure 3 shows the final residue significance distribution. The left panel is for the negative residue significance vector and the right panel is for the positive residue significance. It can be seen that the positive support peptides have more conserved residue significance values compared with the negative support peptides.

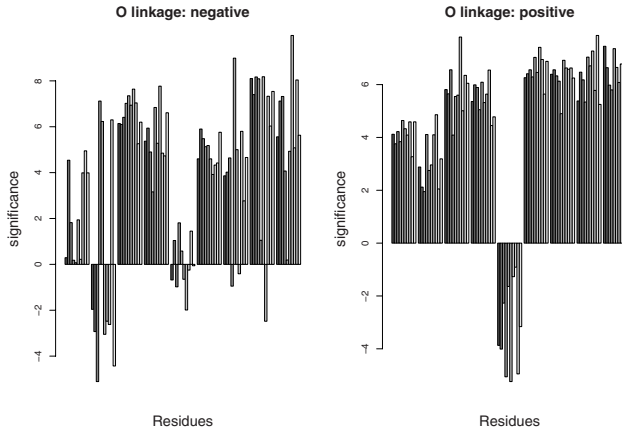


Fig. 3. Residue significance for both negative and positive peptides in the O-linkage data

5 Conclusion

This work has studied the possibility of revealing residue significance quantitatively in peptide classification applications. The algorithm proposed in this work is a revised version of the bio-basis function neural network. The basic idea is to introduce a residue significance parameter for each residue in peptides. Meanwhile, in order to recognise different contributions of negative and positive peptides towards peptide classification, two separate residue significance vectors are used. The whole learning procedure based on the maximum likelihood learning procedure is given. The algorithm has been evaluated on three peptide data sets.

References

1. Sellers, P.H.: On the theory and computation of evolutionary distances. *SIAM J. Appl. Math.* 26, 787–793 (1974)
2. Needleman, S., Wunsch, C.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443–453 (1970)
3. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. *Journal of Molecular Biology* 147, 195–197 (1981)
4. Wilbur, W.J., Lipman, D.J.: Rapid similarity searches of nucleic acid and protein data banks. *PNAS* 80, 726–730 (1993)
5. 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)
6. Poorman, R.A., Tomasselli, A.G., Heinrikson, R.L., Kezdy, F.J.: A cumulative specificity model for protease from human immunodeficiency virus types 1 and 2, inferred from statistical analysis of an extended substrate data base. *J. Biol. Chem.* 22, 14554–14561 (1991)

7. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 257–286 (1989)
8. Nakata, K., Maizel, J.V.: Prediction of operator-binding protein by discriminant analysis. *Gene. Anal. Tech.* 6, 111–119 (1989)
9. Chen, C.P., Rost, B.: State-of-the-art in membrane protein prediction. *Applied Bioinformatics* 1, 21–35 (2002)
10. Senawongse, P., Dalby, A., Yang, Z.R.: Predicting the phosphorylation sites using hidden Markov models and Machine Learning methods. *Journal of Chemical Information and Computer Science* 45, 1147–1152 (2005)
11. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, New York (1995)
12. Hansen, J.E., Lund, O., Engelbrecht, J., Bohr, H., Nielsen, J.O.: Prediction of O-glycosylation of mammalian proteins: specificity patterns of UDP-GalNAc: polypeptide N-acetylgalactosaminyltransferase. *Biochem. J.* 30, 801–813 (1995)
13. Nielsen, M., Lundegaard, C., Worning, P., Lauemoller, S.L., Lamberth, K., Buss, S., et al.: Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Science* 12, 1007–1017 (2003)
14. Gutteridge, A., Bartlett, G.J., Thornton, J.M.: Using a neural network and spatial clustering to predict the location of active sites in enzymes. *Journal of Molecular Biology* 330, 719–734 (2003)
15. Blom, N., Gammeltoft, S., Brunak, S.: Sequence and structure based prediction of eukaryotic protein phosphorylation sites. *J. Mol. Biol.* 24, 1351–1362 (1999)
16. Ehrlich, L., Reczko, M., Bohr, H., Wade, R.C.: Prediction of protein hydration sites from sequence by modular neural networks. *Protein. Eng.* 11, 11–19 (1998)
17. Zien, A., Ratsch, G., Mika, S., Scholkopf, B., Lengauer, T., Muller, K.R.: Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics* 16, 799–807 (2000)
18. Kim, J.H., Lee, J., Oh, B., Kimm, K., Koh, I.: Prediction of phosphorylation sites using SVMs. *Bioinformatics* 20, 3179–3184 (2006)
19. Zhao, Y., Pinilla, C., Valmori, D., Martin, R., Simon, R.: Application of support vector machines for T-cell epitopes prediction. *Bioinformatics* 19, 1978–1984 (2003)
20. Koike, A., Takagi, T.: Prediction of protein-protein interaction sites using support vector machines. *Protein. Eng. Des. Sel.* 17, 165–173 (2004)
21. Qian, N., Sejnowski, T.: Predicting the secondary structure of globular proteins using neural network models. In: *Proceeding of Int J. Conf. On Neural Networks*, pp. 865–884 (1998)
22. Thomson, R., Hodgman, T., Yang, Z.R., Doyle, A.: Characterising proteolytic cleavage site activity using bio-basis function neural networks. *Bioinformatics* 19, 1741–1747 (2003)
23. Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C.: A model of evolutionary change in proteins. *Matrices for detecting distant relationships. Atlas of protein sequence and structure* 5, 345–358 (1978)
24. Cai, Y.D., Yu, H., Chou, K.C.: Artificial neural network method for predicting HIV protease cleavage sites in protein. *J. Protein. Chem.* 17, 607–615 (1998)
25. Narayanan, A., Wu, X., Yang, Z.R.: Mining viral protease data to extract cleavage knowledge. *Bioinformatics* 18, S5–S13 (2002)
26. Chou, K.C.: A key driving force in determination of protein structural classes. *Biochem. Biophys. Res. Commun.* 264, 216–224 (1995)

Analysis of Non-stationary Neurobiological Signals Using Empirical Mode Decomposition

Zareen Mehboob and Hujun Yin

School of Electrical and Electronic Engineering

The University of Manchester, Manchester, M60 1QD, UK

zareen.mehboob@postgrad.manchester.ac.uk, h.yin@manchester.ac.uk

Abstract. This paper describes the use of empirical mode decomposition and Hilbert transform for the analysis of non-stationary signal. The effectiveness of the method is shown on examples of both artificial signals and neurobiological recordings. The method can decompose a signal into data-driven intrinsic mode functions, which can be regarded as underlying oscillations contained in the signal. Then the Hilbert transform is used to extract instantaneous frequencies. Better time-frequency resolution can be obtained compared to the Fourier or wavelet transforms. Initial attempt has also been made on clustering decomposed and extracted signal components. The study suggests that using the instantaneous frequency as the matching criteria can reveal hidden features that may not be observable using features from other transformation methods.

1 Introduction

Many existing signal processing techniques were derived mainly for analysis of stationary signals. In practice, signals are often non-stationary. In other words, the properties of the signals such as statistical moments are changing over time. In biology and neuroscience, recordings are often the results of interactions of oscillations from individual cells or neurons or regions. Data collected from different sources is used to analyze and understand the working and coordination mechanism of different sub-systems within the brain. Different methods are used to record the activities of neurons and neuronal populations at different levels. For instance, spikes or spike trains are the spiking activity of individual or small population neurons; electroencephalography (EEG) is the scalp potential reflected on the electrical activity of large synchronized groups of neurons inside the brain; and local field potentials (LFPs) are the sum of synaptic potentials of a population of neurons [1-3]. Spikes represent high frequency synaptic activity of the neurons, whereas EEG and LFPs are generally low frequency dendrite signals.

Analysis of these signals poses two main challenges, the large quantity and non-stationarity of the data. Though efforts have been made in developing efficient methods and tools to analyze this huge amounts of various datasets [4,5], still there is scope of building authenticated theories to understand how groups of neurons encode, represent and exchange information about the stimuli and produce sensory events. EEG and LFPs are non-stationary time signals with the underlying properties changing rapidly.

This paper explores the use of a recently proposed method, namely empirical mode decomposition (EMD) [6], for the analysis of LFPs and the subsequent cluster analysis. Although various other methods have already been applied for the study of non-stationary data such as the short-time Fourier transform (STFT) and wavelet transform, the EMD has the advantages that the basis functions or narrow-band components are derived from the data as the intrinsic mode functions (IMFs) and better time and frequency resolution can be achieved [6]. The EMD has already been applied to neuroscience data analysis [7].

This paper aims to give a brief overview on the methodology and the significance of the EMD for the analysis of non-stationary time-series through examples and comparison with the Fourier transform and wavelet transform. Then investigation on a neurobiological data set using the EMD and Hilbert transform is conducted, followed by a cluster analysis using the self-organizing maps (SOM) [8] on the processed signals. The rest of the paper is organized as follows. In Section 2 the EMD method and a comparison with other signal analysis methods are given. Section 3 shows several examples and the significance of the EMD method on non-stationary signals. Then application of the EMD on an LFP data set is presented in Section 4, followed by SOM clustering on the decomposed IMFs and IFs in Section 5. Section 6 summarizes and concludes the paper.

2 Hilbert Transform and Empirical Mode Decomposition

Various mathematical procedures exist for analyzing stationary and non-stationary signals. A stationary signal has infinite periodicity unlike the non-stationary ones. Mathematical transformation is applied to such a signal to obtain its properties not visible in the raw signal. The Fourier transform (FT) gives the frequency components contained in the signal. But FT lacks in the time localization of the spectral components. In STFT, a non-stationary signal is divided into small windows, assuming each segment being stationary. However due to the Heisenberg uncertainty principle, choosing the window size can be problematic for STFT. Wavelet transform (WT) was developed to resolve the resolution problems of STFT. While STFT gives a fixed resolution at all times, WT gives varying time and frequency resolutions. In WT, a high frequency component can be located well in time than a low frequency component whereas a low frequency component can be located better in frequency compared to high frequency component. Even with optimized time-frequency resolution WT however suffers with the uncertainty principle. Besides, in WT usually the same basic wavelet is used throughout for all the data.

The concept of instantaneous frequency (IF) offers a natural solution to the problem of representing frequencies at each time-instant. The Hilbert transform (HT) can be used to obtain the IF of a signal and gives a good time- frequency resolution. Given a signal $x(t)$, its HT is defined as,

$$H(t) = \frac{1}{\pi} PV \left[\int_{-\infty}^{\infty} \frac{x(u)}{t-u} du \right] \quad (1)$$

where PV denotes the Cauchy Principal Value.

The analytic signal is then defined as,

$$z(t) = x(t) + iH(t), \quad (2)$$

further in the amplitude and phase form,

$$z(t) = x(t) + iH(t) = A(t)e^{i\theta(t)} \quad (3)$$

where $A(t)$ and $\theta(t)$ are the amplitude and phase of the analytic signal respectively, and can be calculated by,

$$A(t) = \sqrt{x^2(t) + H^2(t)}, \quad \theta(t) = \arctan \frac{H(t)}{x(t)} \quad (4)$$

The instantaneous frequency is then obtained from,

$$\omega(t) = \frac{d\theta(t)}{dt} \quad (5)$$

As the HT can only provide one value at each time and in reality there can be many IFs at a given time-instant in the time-series, it is necessary to separate or decompose these various IFs before IFs are extracted. Hilbert-Huang Transform (HHT) [6] provides a solution to this problem. The HHT consists of two parts: the EMD (the procedure is given in the next section) and HT. In HHT, the EMD first decomposes the signal into several intrinsic mode functions (IMFs) and usually each IMF reflects a narrow band-pass component; and then the HT is applied to each IMF to extract instantaneous frequency. The method has proved to be viable for nonlinear and non-stationary data analysis, especially for time-frequency-energy representations.

3 EMD of Non-stationary Signals

The EMD adaptively decomposes a non-stationary time series into zero-mean amplitude or frequency modulated functions known as intrinsic mode functions (IMFs). The EMD considers the signal oscillations at local level and can reveal important temporal structures that are not achievable by using FT and WT [6].

The EMD assumes that any data consists of different, simple intrinsic modes of oscillations. Thus an intrinsic mode represents an oscillation having the same number of extrema and zero-crossings, symmetric with respect to the local mean. At any given time, the data may have many such different, coexisting modes of oscillation, one superimposing on the others. The result is the final complicated signal. Each of these oscillatory modes is represented by an IMF and can be extracted by the following procedure.

In the given data set, the number of extrema and the number of zero-crossings must either be equal or differ at most by one. At any point, the mean value of the envelopes defined by the local maxima and the local minima should be zero. An IMF can also have a variable amplitude and frequency as functions of time [6].

Given series $x(t)$, the EMD is conducted by a sifting procedure as follows:

- (1) Identify all the local extrema; then connect all the local maxima by a cubic spline to form the upper envelope.
- (2) Repeat the procedure for the local minima to produce the lower envelope.
- (3) The upper and lower envelopes should cover all the data between them.

- (4) Find the local mean envelope $m(t)$ by averaging the two envelopes.
- (5) To get the zero-mean IMF extract $m(t)$ from $x(t)$.
- (6) Repeat the above steps with the residual.

This sifting process is stopped by limiting the size of standard deviation (SD) computed from two consecutive sifting results. SD is normally set to 0.2-0.3. For details and implementation in Matlab see [9]. Using the HT on these IMFs, one can then obtain the instantaneous frequencies of the signal.

Examples of the EMD process on two examples are shown in Fig. 1. The first signal consists of two cosine signals of different frequencies and the second is mixture of several signals. The power spectra of the two signals are shown as indiscriminating on time, and their FT and HT are given in Fig. 1. The wavelet analysis for the same signals is shown in Fig. 2.

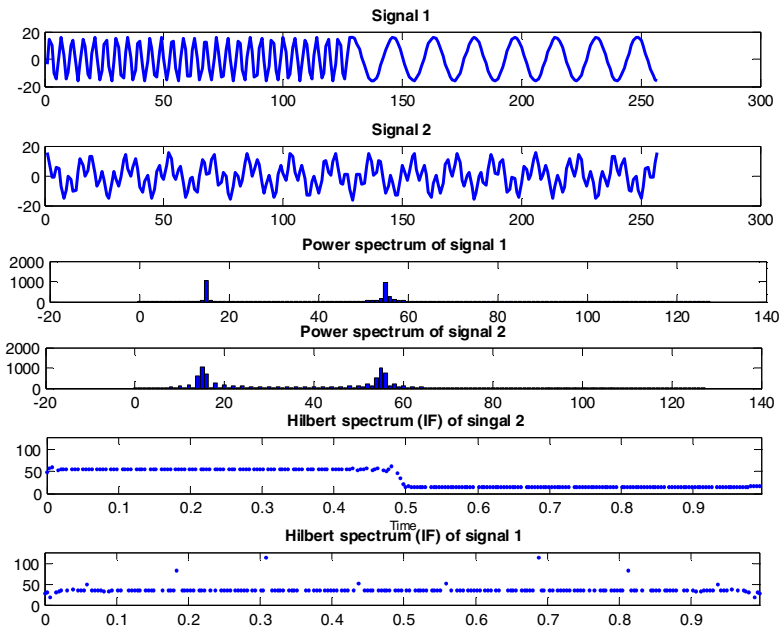


Fig. 1. Two signals and their Fourier spectra with frequency across x-axis and amplitude across y-axis. Hilbert spectra show IF at time instant with time on x-axis and frequency on y-axis.

4 HHT Analysis of LFPs

EMD and HHT have proved to be a useful analysis technique for feature analysis of EEG and other low frequency data such as LFPs. At times a particular spectral component occurring at any time can be of particular interest in studying the brain activity. For example, in EEG, the latency of an event-related potential is of particular interest. Recently EMD has also been applied to EEG and LFPs for analyzing transient brain activities and visual spatial attention [7, 10].

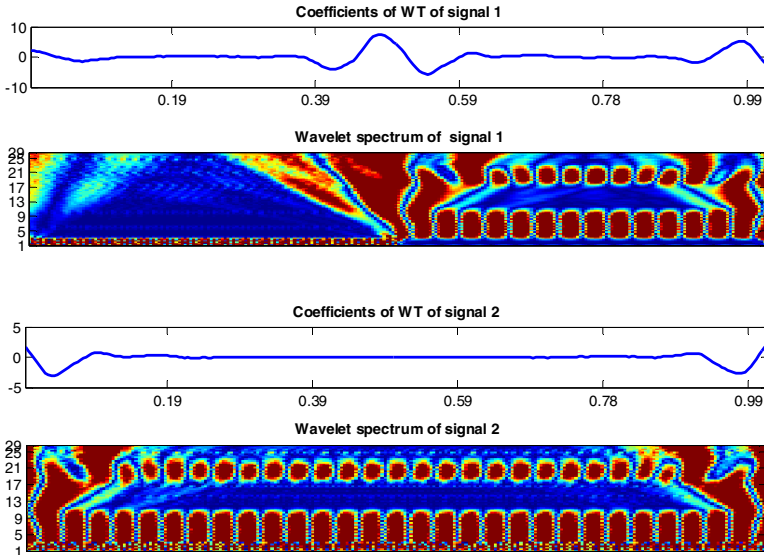


Fig. 2. The Wavelet Transform of Signal 1 and Signal 2. The coefficient line is showing the frequency change for signal 1 across the time-points on x-axis. Note that the plot for frequency change is not as sharp as HT of signal 1 as shown in Fig. 1.

A typical LFP and its EMD is showing in Fig 3. The IFs obtained from the IMFs are shown in Fig 4. IMF 6 is in fact the residual.

5 Clustering Based on IMFs and IFs

The data set used for the cluster analysis consisted of simultaneously recorded spike responses and LFPs obtained from the primary visual cortex in the macaques taken from the previously published series of experiments of Logothetis, Panzeri and colleagues [11, 12]. The stimuli consisted of naturalistic visual signals. Recorded signals (electrodes) were analyzed using standard neurophysiologic criteria and only channels containing well detectable spiking responses were retained for further analysis. Full details on the experiments are reported in [11, 12].

In this study, a set of 3 simultaneously recorded responses in spike format and local field potentials were used. The spike train data is in the form of bit-arrays (0 = no spike and 1= a spike in the considered time bin). In each trial, the 0-200ms post-stimulus time was divided into 100 2ms long time bins. The data for local field potentials is in the form of continuous signals. The LFP of 200ms post stimulus window is shown in Fig 3 (top panel).

In this paper, we have taken the LFP and spikes recorded against 13s of the movie stimulus. The data is sampled at 500Hz. We have initially divided the response against 13s movie in 65 non-overlapping windows, each representing a stimulus. This made one response equal to 200ms. For the clustering purpose we took the first IMF

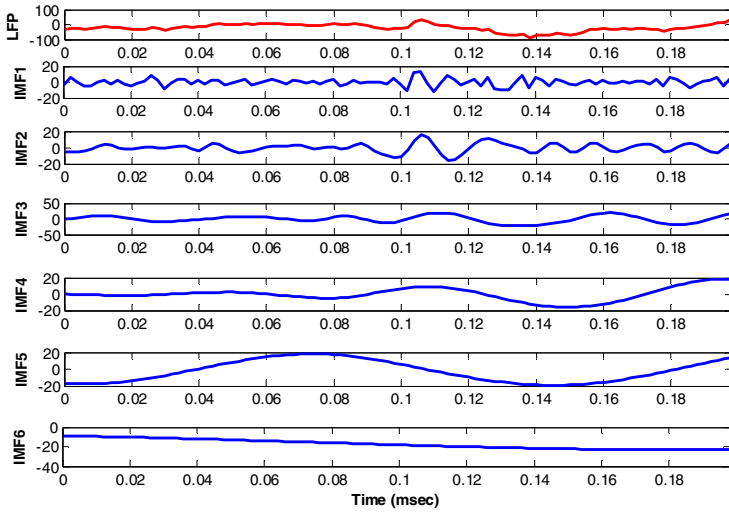


Fig. 3. A typical LFP non-stationary time series (top panel) and EMD decomposed intrinsic mode functions (IMF1 to IMF5). IMF6 is the residual.

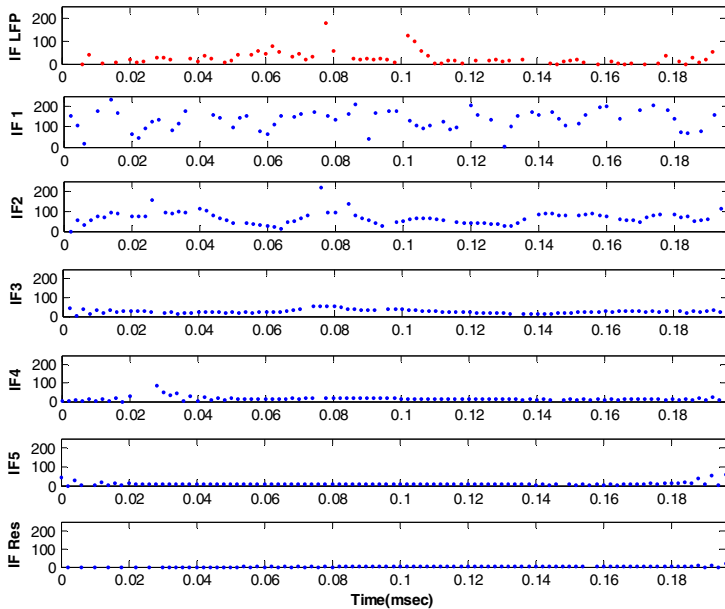


Fig. 4. A typical LFPs and extracted instantaneous frequencies by the Hilbert transform on decomposed intrinsic mode functions of Fig. 3

of each LFP trial having the maximum number of frequency components and clustered the LFPs using the IMF itself and then using the IF as the matching criteria. The mutual information (MI) obtained on the basis of clustering results is show in Fig 5. The SOM based cluster analysis has been previously applied to spike trains and has shown to preserve mutual information of the data set [13, 14].

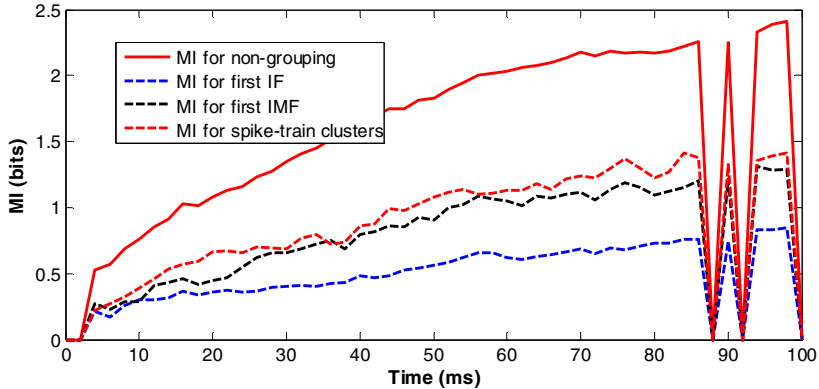


Fig. 5. MI analysis for clustering results based on various features: first IF and first IMF of LFP and spike trains

6 Conclusions

In this paper, the empirical mode decomposition (EMD) method and the Hilbert transform are explored for the analysis of non-stationary signals such as neurobiological recordings. The EMD can decompose a signal into intrinsic mode functions (IMFs), which are said to be some underlying oscillations contained in the signal or its generation. Then the Hilbert transform is used to further extract instantaneous frequencies (IFs) from decomposed IMFs. Such analysis can provide better time-frequency resolution compared to the short-time Fourier or wavelet transforms and is particularly suited for analyzing neuronal recordings. Initial attempt has also been made on clustering decomposed IMFs and extracted IFs. Future work will extend this initial study to a full scale of analysis of LFPs of the data set. The significance and contributions of individual IMFs and IFs with respect to the on-sit stimuli will be examined in terms of information theoretical framework. Such a study will be compared with the previous studies and results obtained on the same data set.

References

1. Brown, E.N., Kass, R.E., Mitra, P.P.: Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience* 7, 456–461 (2004)
2. Nicolelis, M.A.L.: *Methods for Neural Ensemble Recordings*. CRC Press, Boca Raton (1999)
3. Buzsaki, G.: Large scale recordings of neural ensembles. *Nature Neuroscience* 7, 446–451 (2004)

4. Nelken, I.: Encoding stimulus information by spike numbers and mean response time in primary auditory cortex. *Journal of Computational Neuroscience* 19, 199–221 (2005)
5. Arabzadeh, E.P., Diamond, M.E.: Encoding of whisker vibration by rat barrel cortex neurons: implications for texture discrimination. *Journal of Neuroscience* 23, 9146–9154 (2003)
6. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.-C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. Roy. Soc. Lond. A* 454, 903–995 (1998)
7. Liang, H., Bressler, S.L., Buffalo, E.A., Desimone, R., Fries, P.: Empirical mode decomposition of local field potentials from macaque V4 in visual spatial attention. *Biological Cybernetics* 92, 380–392 (2005)
8. Kohonen, T.: *Self-Organizing Maps*. Springer, Heidelberg (1997)
9. Tan, A.: Matlab toolbox function: Hilbert-Huang Transform. (2008)
10. Sweeney-Reed, C.M., Nasuto, S.J.: A novel approach to the detection of synchronisation in EEG based on empirical mode decomposition. *Journal of Computational Neuroscience* 23, 79–111 (2007)
11. Belitski, A., Gretton, A., Magri, C., Murayama, Y., Montemurro, M.A., Logothetis, N.K., Panzeri, S.: Local field potentials and spiking activity in primary visual cortex convey independent information about movie stimuli. *Journal of Neuroscience* 28, 5696–5709 (2008)
12. Montemurro, M.A., Rasch, M.J., Murayama, Y., Logothetis, N.K., Panzeri, S.: Phase-of-firing coding of natural visual stimuli in primary visual cortex. *Current Biology* 18, 375–380 (2008)
13. Mehboob, Z., Panzeri, S., Diamond, M.E., Yin, H.: Topological clustering of synchronous spike trains. In: *Proc. IJCNN 2008* (2008)
14. Yin, H., Panzeri, S., Mehboob, Z., Diamond, M.E.: Decoding population neuronal responses by topological clustering. In: *ICANN 2008*. LNCS, vol. 5164. Springer, Heidelberg (in press, 2008)

Approximate Versus Linguistic Representation in Fuzzy-UCS

Albert Orriols-Puig¹, Jorge Casillas², and Ester Bernadó-Mansilla¹

¹ Grup de Recerca en Sistemes Intel·ligents, Enginyeria i Arquitectura La Salle,
Universitat Ramon Llull, 08022 Barcelona (Spain)
{aorriols, esterb}@salle.url.edu

² Dept. Computer Science and Artificial Intelligence
University of Granada, 18071, Granada (Spain)
casillas@ugr.es

Abstract. This paper introduces an approximate fuzzy representation to Fuzzy-UCS, a Michigan-style Learning Fuzzy-Classifier System that evolves linguistic fuzzy rules, and studies whether the flexibility provided by the approximate representation results in a significant improvement of the accuracy of the models evolved by the system. We test Fuzzy-UCS with both approximate and linguistic representation on a large collection of real-life problems and compare the results in terms of training and test accuracy and interpretability of the evolved rule sets.

Keywords: Genetic algorithms, learning classifier systems, genetic fuzzy systems, supervised learning.

1 Introduction

Fuzzy-UCS [1] is a *Michigan-style learning fuzzy-classifier system* that evolves a fuzzy rule set with descriptive or linguistic representation. One of the main novelties of Fuzzy-UCS with respect to other *genetic fuzzy systems* is that it evolves the rule set on-line from a stream of examples. Fuzzy-UCS represents the knowledge with linguistic fuzzy rules, which are highly interpretable since they share a common semantic. However, as this representation implies the discretization of the feature space, a single rule may not have the granularity required to define the class boundary of a given domain accurately. Thus, Fuzzy-UCS creates a set of overlapping fuzzy-rules around the decision boundaries which match examples of different classes, and the output depends on how the reasoning mechanism combines the knowledge of all these overlapping rules. Fuzzy-UCS proposed three inference schemes which led to a tradeoff between the amount of information used for the inference process and the size of the rule set.

To achieve better accuracy rates in fuzzy modeling, several authors have introduced the so-called *approximate* rule representation (also known as non-grid-oriented fuzzy systems, prototype-based representation, or fuzzy graphs), which proposes that the variables of fuzzy rules define their own fuzzy sets instead of representing linguistic

variables [2]. In this way, approximate fuzzy rules are semantic free, being able to tune the fuzzy sets of any variable of each rule independently. However, this also results in a degradation of interpretability of the rule set, since the fuzzy variables no longer share a unique linguistic interpretation. In this paper, we analyze whether the flexibility provided by the approximate representation allows the system to achieve higher performance and how this affects the interpretability of the evolved rule sets. That is, the approximate representation is more powerful than the linguistic one since it enables fuzzy systems to evolve independent fuzzy sets for each attribute that fit the class boundaries of each particular problem accurately. Nonetheless, the search space also increases since the semantics is evolved together with the fuzzy rules, posing more difficulties to the learner. Moreover, this flexibility could also result in overfitting the training instances in complex, noisy environments. Therefore, the aim of the present work is to study the frontier in the accuracy-interpretability tradeoff, clearly identifying the advantages and disadvantages—in terms of accuracy and readability of the rule sets—of having a more flexible knowledge representation in the field of on-line learning. For this purpose, we include the approximate representation to Fuzzy-UCS and adapt several mechanisms to deal with it. This new algorithm is addressed as Fuzzy-UCS with Approximate representation, i.e., Fuzzy-UCSa. We compare the behavior of Fuzzy-UCS and Fuzzy-UCSa in a large collection of real-life problems.

The remainder of this paper is organized as follows. Section 2 describes Fuzzy-UCSa focusing on the new fuzzy representation. Section 3 explains the analysis methodology and presents the results. Finally, Section 4 concludes the work.

2 The Approximate Fuzzy-UCS Classifier System

Fuzzy-UCSa is a system that extends Fuzzy-UCS [1] by introducing an approximate fuzzy representation. Fuzzy-UCSa works in two different models: exploration or training and exploitation or test. As follows, we describe the system focusing on the changes introduced with respect to Fuzzy-UCS. For further details, the reader is referred to [1].

2.1 Knowledge Representation

Fuzzy-UCS evolves a population [P] of classifiers which consist of a fuzzy rule and a set of parameters. The fuzzy rule follows the structure

$$\text{IF } x_1 \text{ is FS}_1^k \text{ and } \dots \text{ and } x_n \text{ is FS}_n^k \text{ THEN } c^k \text{ WITH } w^k, \tag{1}$$

where each input variable x_i is represented by a fuzzy set FS_i . In our experiments, we used triangular fuzzy sets; so, each FS_i is defined by the left vertex a , the middle vertex b , and the right vertex c of the triangle, i.e., $\text{FS}_i = (a, b, c)$. The consequent of the rule indicates the class c^k which the rule predicts. w^k is a weight ($0 \leq w^k \leq 1$) that denotes the soundness with which the rule predicts class c^k . The matching degree $\mu_A^k(e)$ of an example e with a classifier k is computed as the T-norm (we use the product) of the membership degree of each input attribute e_i with the corresponding fuzzy set FS_i . We enable the system to deal with missing values by considering that $\mu_A^k(e) = 1$ if e_i is not known.

Each classifier has four main parameters: 1) the fitness F , which estimates the accuracy of the rule; 2) the correct set size cs , which averages the sizes of the correct sets in which the classifier has participated (see Sect. 2.2); 3) the experience exp , which computes the contributions of the rule to classify the input instances; and 4) the numerosity n , which counts the number of copies of the rule in the population.

2.2 Learning Interaction

At each learning iteration, the system receives an input example e that belongs to class c . Then, it creates the match set $[M]$ with all the classifiers in $[P]$ that have a matching degree $\mu_A^k(e)$ greater than zero. Next, in exploration mode, the classifiers in $[M]$ that advocate class c form the correct set $[C]$. In exploitation mode, the system returns the class of the rule that maximizes $F^k \cdot \mu_A^k(e)$ and no further action is taken. If none of the classifiers in $[C]$ match e with $\mu_A^k(e) > 0.5$, the covering operator is triggered, which creates the classifier that maximally matches the input example. The covering operator creates an independent triangular-shape fuzzy set for each input variable with the following supports

$$(\text{rand}(\min_i - (\max_i - \min_i)/2, e_i), e_i, \text{rand}(e_i, \max_i + (\max_i - \min_i)/2)), \tag{2}$$

where \min_i and \max_i are the minimum and maximum value that the attribute i can take, e_i is the attribute i of the example e for which covering has been fired, and rand generates a random number between both arguments. The parameters F , n , and exp of the new classifiers are set to 1. The new classifier is inserted into the population, deleting another one if there is not room for it.

2.3 Parameters Update

In the end of each learning iteration, Fuzzy-UCSa updates the parameters of the rules in $[M]$. First, the experience of the rule is incremented according to the current matching degree: $exp_{t+1}^k = exp_t^k + \mu_A^k(e)$. Next, the fitness is updated. For this purpose, each classifier internally maintains a vector of classes $\{c_1, \dots, c_m\}$ and a vector of associated weights $\{v_1^k, \dots, v_m^k\}$. Each weight v_j^k indicates the soundness with which rule k predicts class j for an example that fully matches this rule. The class c^k advocated by the rule is the class with the maximum weight v_j^k . Thus, given that the weights may change due to successive updates, the class that a rule predicts may also vary.

To update the weights, we first compute the sum of correct matchings cm^k for each class j : $cm_{jt+1}^k = cm_{jt}^k + m(k, j)$, where $m(k, j) = \mu_A^k(e)$ if the class predicted by the classifier equals the class of the input example and zero otherwise. Then, cm_{jt+1}^k is used to calculate the weights v_{jt+1}^k : $v_{jt+1}^k = cm_{jt+1}^k / exp_{t+1}^k$. Note that the sum of all the weights is 1.

The fitness is computed from the weights with the aim of favoring classifiers that match examples of a single class. We use $F_{t+1}^k = v^k \max_{t+1} - \sum_{j|j \neq \max} v_{jt+1}^k$, where we subtract the values of the other weights from the weight with maximum value $v^k \max$. The fitness F^k is the value used as the weight w^k of the rule (see Equation 1). Next, the correct set size of all the classifiers in $[C]$ is calculated as the arithmetic average of the sizes of all the correct sets in which the classifier has participated.

2.4 Discovery Component

Fuzzy-UCSa uses a steady-state *genetic algorithm* (GA) [3] to discover new promising rules. The GA is triggered in [C] when the average time since its last application upon the classifiers in [C] exceeds a certain threshold θ_{GA} . It selects two parents p_1 and p_2 from [C] using proportionate selection [3], where the probability of selecting a classifier k is proportional to $(F^k)^v \cdot \mu_A^k(e)$, in which $v > 0$ is a constant that fixes the pressure toward maximally accurate rules (in our experiments, we set $v=10$). Rules with negative fitness are not considered for selection. The two parents are copied into offspring ch_1 and ch_2 , which undergo crossover and mutation with probabilities χ and μ respectively. The crossover operator generates the fuzzy sets for each variable of the offspring as

$$b_{ch1} = b_{p1} \alpha + b_{p2} (1-\alpha) \text{ and } b_{ch2} = b_{p1} (1-\alpha) + b_{p2} \alpha \tag{3}$$

where $0 \leq \alpha \leq 1$ is a configuration parameter. As we wanted to generate offspring whose middle vertex b was close to the middle vertex of one of his parents, we set $\alpha=0.005$ in our experiments. Next, for both offspring, the procedure to cross the most-left and most-right vertices is the following. First, the two most-left and two most-right vertices are chosen

$$\min_{left} = \min(a_{p1}, a_{p2}, b_{ch}) \text{ and } \text{mid}_{left} = \text{middle}(a_{p1}, a_{p2}, b_{ch}) , \tag{4}$$

$$\text{mid}_{right} = \text{middle}(c_{p1}, c_{p2}, b_{ch}) \text{ and } \max_{right} = \max(c_{p1}, c_{p2}, b_{ch}) , \tag{5}$$

And then, these two values are used to generate the vertices a and c .

$$a_{ch} = \text{rand}(\min_{left}, \text{mid}_{left}) \text{ and } c_{ch} = \text{rand}(\text{mid}_{right}, \max_{right}) , \tag{6}$$

where the functions *min*, *middle*, and *max* return respectively the minimum, the middle, and the maximum value among their arguments.

The mutation operator decides randomly if each vertex of a variable has to be mutated. The central vertex is mutated as follows:

$$b = \text{rand}(b - (b - a) \cdot m_0, b + (c - b) \cdot m_0) , \tag{7}$$

where m_0 ($0 < m_0 \leq 1$) defines the strength of the mutation. The left-most vertex is mutated as

$$a = \text{rand}(a - m_0(b-a)/2, a) \quad \text{if } F > F_0 \text{ \& no crossover,} \tag{8}$$

$$a = \text{rand}(a - m_0(b-a)/2, a + m_0(b-a)/2) \quad \text{otherwise.} \tag{9}$$

And the right-most vertex

$$c = \text{rand}(c, c + m_0(c-b)/2) \quad \text{if } F > F_0 \text{ \& no crossover,} \tag{10}$$

$$c = \text{rand}(c - m_0(c-b)/2, c + m_0(c-b)/2) \quad \text{otherwise.} \tag{11}$$

That is, if the rule is accurate enough ($F > F_0$) and has not been generated through crossover, mutation forces to generalize it. Otherwise, it can be either generalized or specified. In this way we increase the pressure toward maximum general and accurate rule sets.

The new offspring are introduced into the population. First, we check whether there exists a classifier in [C] that subsumes the new offspring. If it exists, the numerosity of the subsumer is increased. Otherwise, the new classifier is inserted into the population. We consider that a classifier $k1$, which is experienced ($\exp^{k1} > \theta_{\text{sub}}$) and accurate enough ($F^{k1} > F_0$), can subsume another classifier $k2$ if for each variable i , $a_{k1}^i \leq a_{k2}^i$, $c_{k1}^i \geq c_{k2}^i$, and $b_{k1}^i - (b_{k1}^i - a_{k1}^i)\delta \leq b_{k2}^i \leq b_{k1}^i + (c_{k1}^i - b_{k1}^i)\delta$, where δ is a discount parameter (in our experiments we set $\delta=0.001$). Thus, a rule condition subsumes another if the supports of the subsumed rule are enclosed in the supports of the subsumer rule and the middle vertices of their triangular-shaped fuzzy sets are close in the feature space.

If the population is full, excess classifiers are deleted from [P] with probability proportional to their correct set size estimate cs_k and their fitness F^k [1].

3 Experiments

In this section, we analyze if the approximate representation a) permits to fit the training instances more accurately, b) whether this improvement is also present in the prediction of previously unseen instances, and c) the impact on the interpretability of the evolved rule set. As follows we explain the methodology and present the obtained results.

3.1 Methodology

We compare Fuzzy-UCSa as defined in the previous section with Fuzzy-UCS with the three types of reasoning schemes defined in [1], that is, weighted average (wavg), in which all matching rules emit a vote for the class they predict; action winner (awin), in which the class of the rule that maximizes $Fk \cdot \mu_A^k(e)$ is chosen as output; and most numerous and fit rules (nfit), in which only the most numerous and fit rules are kept in the final population and all the matching rules emit a vote for the class they predict. Moreover, we also included C4.5 in the comparison to analyze how Fuzzy-UCS performs with respect to one of the most influential learners. We employed the same collection of twenty real-life problems used in [1] for the analysis.

We used the accuracy, i.e., the proportion of correct predictions, and the number of rules in the population to compare the performance and interpretability of the different approaches. To obtain reliable estimates of these metrics, we employed a ten-fold cross validation procedure. The results were statistically analyzed following the recommendations pointed out in [4]. We applied the multiple-comparison test of Friedman to contrast the null hypothesis that all the learning algorithms performed equivalently on average. If Friedman's test rejected the null hypothesis, we used the non-parametric Nemenyi test to compare all learners with each other. We complemented the statistical analysis by comparing the performance of each pair of learners by means of the non-parametric Wilcoxon signed-ranks test. For further information about the statistical tests see [4].

We configured both systems as (see [1] for notation details): $N=6,400$, $F_0 = 0.99$, $v = 10$, $\{\theta_{GA}, \theta_{del}, \theta_{sub}\} = 50$, $\chi = 0.8$, $\mu = 0.04$, $\delta=0.1$ and $P \# = 0.6$. Moreover, for Fuzzy-UCS, we set the number of linguistic terms to 5.

3.2 Results

Our first concern was to compare the precision in fitting the training instances of Fuzzy-UCSa with respect to Fuzzy-UCS and show how both systems perform with respect to C4.5. Thus, we computed the training accuracy obtained with the five approaches. Table 1 summarizes the average rank of each algorithm (the detailed results are not included due to space limitations). As a case study, Fig. 1(a) shows the domain of one of the tested problems, tao, Figs. 1(b) and 1(c) plot the decision boundaries learned by Fuzzy-UCS awin with 5 and 15 linguistic terms per variable—the grid in the two figures indicates the partitions in the feature space made by the cross-points of the triangular membership functions associated to the different fuzzy sets—, and Fig. 1(d) shows the decision boundaries learned by Fuzzy-UCSa. The results clearly show that the flexibility provided by the approximate representation enabled Fuzzy-UCSa to fit the training instances more accurately.

The multi-comparison test permitted to reject the null hypothesis that all the learners were equally accurate at $\alpha=0.001$. The post-hoc Nemenyi test, at $\alpha=0.1$, indicated that Fuzzy-UCSa achieved significantly better training performance than Fuzzy-UCS with any inference type and equivalent results to C4.5. Moreover, Fuzzy-UCS awin significantly degraded the training performance achieved with Fuzzy-UCS nfit. The pairwise comparisons by means of the non-parametric Wilcoxon signed-ranks test at $\alpha=0.05$ confirmed the conclusions extracted by the Nemenyi test.

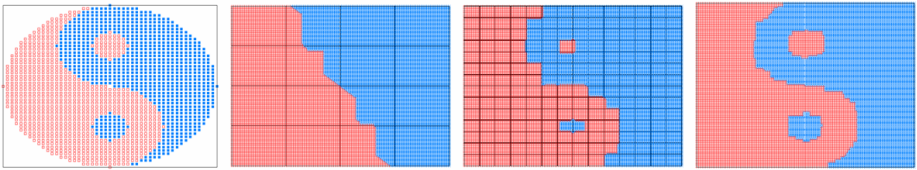


Fig. 1. Tao domain (a) and decision boundaries obtained by Fuzzy-UCS awin with 5 (b) and 15 (c) linguistic terms per variable and Fuzzy-UCSa (d). The training accuracy achieved in each case is 83.24%, 94.74%, and 96.94% respectively.

As expected, the approximate representation enabled Fuzzy-UCSa to fit the training examples more accurately; since there was no semantic shared among all variables, each variable could define its own fuzzy sets. Next, we analyzed if this improvement was also present in the test performance. Table 1 shows the average rank of test performance. The multi-comparison test rejected the hypothesis that all the learners performed the same on average at $\alpha = 0.001$. The Nemenyi procedure, at $\alpha=0.1$, identified two groups of techniques that performed equivalently. The first group included Fuzzy-UCS wavg, Fuzzy-UCSa, and C4.5. The second group comprised Fuzzy-UCSa, Fuzzy-UCS awin, and Fuzzy-UCS nfit. The same significant differences were found by the pairwise comparisons.

Further analysis pointed out that Fuzzy-UCSa was overfitting the training data in some of the domains. To contrast this hypothesis, we monitored the evolution of the training and test performance of the problems in which Fuzzy-UCSa degraded the

Table 1. Comparison of the average rank of train accuracy, test accuracy, and rule set size of linguistic Fuzzy-UCS with weighted average (wavg), action winner (awin), and most numerous and fitted rules inference (nfit), and Fuzzy-UCSa on a set of twenty real-world problems. The training and test accuracy of C4.5 is also included.

		Fuzzy-UCS			Fuzzy-UCSa	C4.5
train	Rank	3.35	4.20	3.10	1.75	2.60
	Pos	4	5	3	1	2
test	Rank	2.05	3.25	3.80	2.95	2.95
	Pos	1	4	5	2.5	2.5
size	Rank	3.95	2.05	1.00	3.00	-
	Pos	4	2	1	3	-

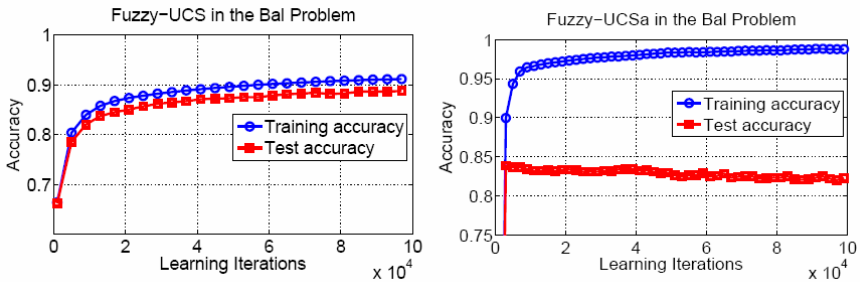


Fig. 2. Evolution of the training and test accuracies obtained with Fuzzy-UCS wavg and Fuzzy-UCSa in the *bal* problem

results obtained by Fuzzy-UCS with any inference type. Figure 2 plots the evolution of the training and test performance in the *bal* problem for Fuzzy-UCS wavg and Fuzzy-UCSa. During the first 5,000 learning iterations, both training and test performances of Fuzzy-UCSa rapidly increased, achieving about 90% and 84% accuracy rate respectively. After that, the training performance continued increasing while the test performance slightly decreased. After 100,000 iterations, the training performance reached 98%; nonetheless, the test performance decreased to 82%. Thus, at a certain point of the learning, the flexibility of the approximate representation led Fuzzy-UCSa to overfit the training instances in order to create more accurate classifiers, which went in detriment of the test performance. On the other hand, the training and test performance of Fuzzy-UCS wavg continuously increased, showing no signs of overfitting.

Finally, Table 1 also shows the average rank of the number of rules evolved for Fuzzy-UCS and Fuzzy-UCSa. The Friedman test rejected the hypothesis that the population sizes were equivalent on average at $\alpha=0.001$. The post-hoc Nemenyi test supported the hypothesis that the four learners evolved populations with significantly different sizes. Fuzzy-UCS wavg created the biggest populations, closely followed by Fuzzy-UCSa. Nonetheless, Fuzzy-UCSa uses an approximate representation, in which

rules do not share the same semantic, thus, impairing the readability of the rule sets. Fuzzy-UCS awin and, specially, Fuzzy-UCS nfit resulted in the smallest populations.

4 Conclusions

This paper analyzed the advantages and disadvantages provided by the flexibility of the approximate representation in detail. We showed that the approximate representation enabled Fuzzy-UCSa to fit the training data more accurately. Nonetheless, there was no statistical evidence of this improvement in the test performance and it was identified that Fuzzy-UCSa may overfit the training instances in complex domains; furthermore, the approximate representation degraded the readability of the final rule sets. Therefore, the analysis served to identify that the flexibility provided by the approximate representation does not produce any relevant improvement to Fuzzy-UCS, strengthening the use of a linguistic, more readable representation.

Acknowledgments

The authors thank the support of *Ministerio de Educación y Ciencia* under projects TIN2005-08386-C05-01 and TIN2005-08386-C05-04 and *Generalitat de Catalunya* under grants 2005FI-00252 and 2005SGR-00302.

References

1. Orriols-Puig, A., Casillas, J., Bernadó-Mansilla, E.: Fuzzy-UCS: a Michigan-style learning fuzzy-classifier system for supervised learning. *IEEE Transactions on Evolutionary Computation* (in press)
2. Alcalá, R., Casillas, J., Cordon, O., Herrera, F.: Building fuzzy graphs: features and taxonomy of learning for non-grid-oriented fuzzy rule-based systems. *Journal of Intelligent and Fuzzy Systems* 11(3-4), 99–119 (2001)
3. Goldberg, D.E.: *Genetic algorithms in search, optimization & machine learning*, 1st edn. Addison-Wesley, Reading (1989)
4. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)

Fuzzy Classification with Multi-objective Evolutionary Algorithms

Fernando Jiménez, Gracia Sánchez, José F. Sánchez, and José M. Alcaraz

Facultad de Informática Campus de Espinardo,
30071 Murcia Spain
{fernan, gracia}@um.es, jsanchez@ditec.um.es,
josem.alcaraz@carm.es

Abstract. In this work we propose, on the one hand, a multi-objective constrained optimization model to obtain fuzzy models for classification considering criteria of accuracy and interpretability. On the other hand, we propose an evolutionary multi-objective approach for fuzzy classification from data with real and discrete attributes. The multi-objective evolutionary approach has been evaluated by means of three different evolutionary schemes: Preselection with niches, NSGA-II and ENORA. The results have been compared in terms of effectiveness by means of statistical techniques using the well-known standard Iris data set.

Keywords: Multi-objective Evolutionary algorithms. Fuzzy classification.

1 Introduction

An effective form to approach the classification problem is by means of fuzzy modeling, where the most important problem is the identification of a fuzzy model using input-output data. Given a data set with some functional dependency, the question is to obtain a set of fuzzy rules from the data that describes the behavior of the output function maximizing the accuracy and the interpretability. Owing to the complexity of the problem, the evolutionary computation has been one of the more accepted techniques for rule generation.

The evolutionary computation [3] has been applied successfully to learn fuzzy models [4, 7]. This has produced many complex algorithms and, as it is said in [11] and [12], often the interpretability of the resulting rule base is not considered to be of importance.

Current evolutionary approaches for multi-objective optimization consist of multiobjective EAs based on the Pareto optimality notion, in which all objective are optimized simultaneously to find multiple non-dominated solutions in a single run.

The fuzzy modeling approach can also be considered from the multi-objective evolutionary perspective [6]. Current research lines in fuzzy modeling mostly tackle improving accuracy in descriptive models, and improving interpretability in approximative models [1]. This paper deals with the second issue.

In this paper, we propose a multi-objective neuro-evolutionary optimization approach to generate fuzzy classification models considering accuracy and interpretability criteria. Section 2 describes the fuzzy classification model and the criteria considered in the optimization process. A multi-objective constrained optimization model is proposed. Section 3 shows the main components of the three multi-objective EAs used in this paper. Section 4 shows the experiments and the results obtained for the problem of classification of the Iris data set. Section 5 concludes the paper.

2 Fuzzy Classification

2.1 Fuzzy Model Identification

We consider a set of N training input data composed as follows:

- An input real attributes vector $\mathbf{x} = (x_1, \dots, x_p)$, $x_i \in [l_i, u_i] \subset \mathfrak{R}$, $i = 1, \dots, p$, $p \geq 0$.
- An input discrete attributes vector $\mathbf{w} = (w_1, \dots, w_q)$, $w_i \in \{1, \dots, v_i\}$, $i = 1, \dots, q$, $q \geq 0$, where v_i is the amount of classes for the i -th parameter,
- A value for the output discrete attribute $y \in \{1, \dots, z\}$, where z is the amount of classes for this output attribute.

So, a boolean value for an input or output attribute can be represented as an discrete attribute w_i or y considering $v_i = 2$ or $z = 2$.

We consider a fuzzy classification model formed by M rules R_1, \dots, R_M . There must exist at least a rule for each z output class, therefore $M \geq z$. Each rule R_j ($j = 1, \dots, M$) contains p fuzzy sets A_{ij} ($i = 1, \dots, p$) associated to p real input attributes, q discrete values B_{ij} ($i = 1, \dots, q$) associated to q discrete input attributes, and a discrete value C_j associated to the discrete output attribute. So, the structure of a rule R_j is:

$$\begin{aligned}
 R_j : \quad & \text{If} \quad x_1 \text{ is } A_{1j} \quad \text{and} \dots \text{and} \quad x_p \text{ is } A_{pj} \quad \text{and} \\
 & w_1 \text{ is } B_{1j} \quad \text{and} \dots \text{and} \quad w_q \text{ is } B_{qj} \\
 & \text{then} \quad y \text{ is } C_j
 \end{aligned} \tag{1}$$

Each fuzzy set A_{ij} ($i = 1, \dots, p$) ($j = 1, \dots, M$) is described by his membership function $\mu_{A_{ij}} : X_i \rightarrow [0, 1]$, where X_i is the range of the real input attribute x_i . In our model have been used Gaussian membership functions:

$$\mu_{A_{ij}}(x_i) = \exp \left[-\frac{1}{2} \left(\frac{x_i - a_{ij}}{\sigma_{ij}} \right)^2 \right]$$

where $a_{ij} \in [l_i, u_i]$ is the center, and $\sigma_{ij} > 0$ is the variance.

The model output $\psi(\mathbf{x}, \mathbf{w})$ for an input data struct with a real attributes vector \mathbf{x} and a discrete attributes vector \mathbf{w} belongs to the output class $C \in \{1, \dots, z\}$ whose model activation value $\lambda_C(\mathbf{x}, \mathbf{w})$ is maximum. This is: $\psi(\mathbf{x}, \mathbf{w}) = \arg \max_{C=1}^z \lambda_C(\mathbf{x}, \mathbf{w})$.

The activation value of the model $\lambda_C(\mathbf{x}, \mathbf{w})$ for an input (\mathbf{x}, \mathbf{w}) and an output class $C \in \{1, \dots, z\}$ is calculated adding the firing strength $\varphi_j(\mathbf{x}, \mathbf{w})$ of each rule R_j ($j = 1, \dots, M$) whose value for the discrete output attribute C_j is C , this is:

$$\lambda_C(\mathbf{x}, \mathbf{w}) = \sum_{\substack{j=1, \dots, M \\ C_j=C}} \varphi_j(\mathbf{x}, \mathbf{w})$$

The firing strength $\varphi_j(\mathbf{x}, \mathbf{w})$ for the R_j rule for the input (\mathbf{x}, \mathbf{w}) is calculated:

$$\varphi_j(\mathbf{x}, \mathbf{w}) = (\phi_j(\mathbf{w}) + 1) \prod_{i=1}^p \mu_{A_{ij}}(x_i)$$

where $\phi_j(\mathbf{w})$ is the number of input discrete attributes and $w_j = B_{ij}$.

2.2 Criteria for Fuzzy Classification

We consider three main criteria: accuracy, transparency, and compactness.

Accuracy. Classification Rate: $CR = \frac{\Phi}{N}$ where Φ is the amount of data for which $\psi(\mathbf{x}, \mathbf{w}) = y$.

Transparency. Similarity [10]:

$$S = \max_{\substack{i, k=1, \dots, M \\ j=1, \dots, p \\ A_{ij} \neq A_{kj}}} S(A_{ij}, A_{kj}) \quad \text{Where } S(A, B) = \max \left\{ \frac{|A \cap B|}{|A|}, \frac{|A \cap B|}{|B|} \right\} \quad (2)$$

Compactness. Number of rules, (M) and the number of different fuzzy sets (L).

2.3 Optimization Model

According to the previous remarks, we propose the following multi-objective constrained optimization model, with two objectives and one constraint:

$$\begin{aligned} & \text{Maximize } f_1 = CR \\ & \text{Minimize } f_2 = M \\ & \text{Subject to } g_1 = S - g_s \leq 0 \end{aligned} \quad (3)$$

It is assumed that the number of different fuzzy sets (L) is smaller in those models with a number of rules (M) smaller. Therefore in the optimization model has been used only M . On the other hand, we want to minimize the similarity (S) only until a

certain threshold (g_s), since models with little similar fuzzy sets will not be sufficiently accurates.

3 Multi-objective Evolutionary Algorithms

We propose a hybrid learning system to find multiple Pareto-optimal solutions simultaneously, considering accuracy, transparency and compactness criteria. We study different multi-objective EAs to evolve the structure and parameters of the rule set, together with a rule set simplification operator that is used to encourage rule base transparency and compactness.

3.1 Representation of Solutions

Solutions are represented with codification of real and discrete numbers using a Pittsburgh approach. Each individual of the population contains a number of rules M that can vary between min and max , where the values min and max are chosen by a decision maker taking into account that it must have at least one rule by each output class, that is to say: $z \leq min \leq M \leq max$. Each rule R_j , $j = 1, \dots, M$, is represented as follows:

- Fuzzy sets associated to the real input attributes x_i , $i = 1, \dots, p$, using the numbers $a_{ij} \in [l_i, u_i]$ y $\rho_{ij} = 2\sigma_{ij}^2 > 0$, that define the centers and variances respectively.
- Discrete values associated to the discrete input attributes w_i , $i = 1, \dots, q$, using integer numbers $b_{ij} \in \{1, \dots, v_i\}$.
- The discrete value associated to the output discrete attribute, using an integer value $c_j \in \{1, \dots, z\}$.
- In order to carry out adaptive crosses and mutation, each individual has two discrete parameters $d \in (0, \delta)$ and $e \in (0, \varepsilon)$ associated to the crossing and mutation, where δ is the number of crossing operators and ε is the number of mutation operators. It has, also, two real parameters $e_c, e_v \in [0, 1]$ that respectively define the amplitude of the mutation of the centers and variances of the fuzzy sets.

3.2 Initial Population and Constraint Handling

Individuals are initialized randomly generating a uniform distribution within the limits of the search space. It is included, in addition, another condition to the limits of the variances of the Gaussian fuzzy sets, so that:

$$\alpha_i = \frac{u_i - l_i}{\gamma} \leq \rho_{ij} \leq u_i - l_i = \beta_i, i = 1, \dots, p$$

where γ is a value that is used to calculate the minimum variances. It's necessary that a rule exists for each output value between 1 and z . With regard to constraints, we use the constraint handling rule proposed in [8].

3.3 Variation Operators

In order to achieve an appropriate exploitation and exploration of the potential solutions in the search space, variation operators working in the different levels of the individuals are necessary. So, different mutation and crossing operators are defined that are apply in an adaptive way with a probability p_v chosen by the user.

- **Fuzzy Sets Crossover.** Changes two fuzzy sets randomly selected.
- **Rule Crossover.** Changes two rules randomly selected.
- **Rule Crossover by Increase.** Adds to both individuals a rule randomly selected from the other one.
- **Gaussian Set Center Mutation.** Mutation of the center of one or several randomly selected fuzzy sets.
- **Gaussian Set Variance Mutation.** Mutation of the variance of one or several randomly selected fuzzy sets.
- **Gaussian Set Mutation.** Mutation of a fuzzy set randomly selected..
- **Mutation by Rules Increase.** Adds a new rule with random values.
- **Mutation of a Discrete Value.** Mutation of a discrete value randomly selected.

3.4 Rule Set Simplification

The method to simplify the rule set is the proposed in [5] and [11], merging the sets when $S(A_{ij}, A_{ik}) > \eta_1$ and splitting them when $\eta_1 > S(A_{ij}, A_{ik}) > \eta_2$. The values η_1 and η_2 are the threshold to carry out the fusion or the split ($0 < \eta_2 < \eta_1 < 1$).

3.5 Decision Process

Let $S = \{s_1, \dots, s_D\}$ be the set of non-dominated solutions such that $CR(s_i) \geq CR^{min}$, ($i = 1, \dots, D$) where CR^{min} is the minimum value for the classification rate tolerated by a decision maker. We will select the solution s_i with less amount of rules and fuzzy sets such that s_i is enough accurate and interpretable.

4 Experiments and Results

The experiments aim is to solve the iris data classification problem. We randomly choose 99 instances as the training instances and 50 instances as the testing instances.

We have to deal with a problem with $p = 4$ real-coded inputs, $q = 0$ discrete inputs and $z = 3$ possible values for the output discrete attribute.

Preselection with niches, NSGA-II and ENORA algorithms have been executed 100 times with the next parameters: $N = 100$, $No\ of\ Evaluations = 10^5$, $min = 10$, $max = 20$, $\gamma = 100$, $p_v = 0.1$, $g_s = 0.2$, $\eta_1 = 0.6$, $\eta_2 = 0.2$ Preselection with niches algorithm uses: $No\ of\ Children = 10$, and $No\ min\ Niches = 5$. Table 1 shows the better non-dominated solutions obtained, and the selected solution after the decision process

Table 1. Non-dominated solutions (best results after 100 runs) obtained in this paper for the Iris data set classification problem. The chosen solution by the decision process are bolded.

	M	L	CR -training	CR -evaluation	S
Preselection w. niches	10	9	0.989899	0.94	0.198820
NSGA-II	10	9	0.989899	0.94	0.194545
	10	9	0.989899	0.94	0.196403
	10	9	0.989899	0.94	0.196397
	10	9	0.989899	0.94	0.199681
	10	9	0.989899	0.94	0.198885
	10	9	0.989899	0.94	0.197146
	10	9	0.989899	0.94	0.197635
ENORA	10	8	0.989899	0.92	0.199105
	13	8	1.000000	0.94	0.199827

Table 2. Fuzzy model with 13 rules for the problem obtained with ENORA

x₁	x₂	x₃	x₄	y
(5.05;1.73)	(3.94;0.55)	(2.86;1.79)	(0.33;0.99)	1
(5.05;1.73)	(2.53;0.50)	(2.86;1.79)	(0.33;0.99)	1
(5.05;1.73)	(2.53;0.50)	(2.86;1.79)	(0.33;0.99)	2
(7.51;0.73)	(3.94;0.55)	(5.99;3.43)	(0.33;0.99)	2
(5.05;1.73)	(2.53;0.50)	(5.99;3.43)	(0.33;0.99)	2
(5.05;1.73)	(3.94;0.55)	(5.99;3.43)	(0.33;0.99)	2
(5.05;1.73)	(3.94;0.55)	(5.99;3.43)	(2.49;0.80)	2
(7.51;0.73)	(2.53;0.50)	(2.86;1.79)	(0.33;0.99)	2
(5.05;1.73)	(3.94;0.55)	(2.86;1.79)	(2.49;0.80)	2
(5.05;1.73)	(3.94;0.55)	(5.99;3.43)	(2.49;0.80)	3
(5.05;1.73)	(2.53;0.50)	(5.99;3.43)	(0.33;0.99)	3
(7.51;0.73)	(3.94;0.55)	(5.99;3.43)	(2.49;0.80)	3
(5.05;1.73)	(2.53;0.50)	(2.86;1.79)	(2.49;0.80)	3

(section 3.6) for each algorithm is bolded. Finally, the solution obtained by ENORA with 13 (table 2, Figure 1) rules is chosen.

To compare the algorithms, we use the hypervolume indicator (v) which calculates the fraction of the objective space which is non-dominated by any of the solutions obtained by the algorithm [2, 9, 13].

The statistics showed in Table 3 indicate that ENORA obtains lower localization values and dispersion than Preselection with niches and NSGA-II. Finally, the 95% confidence intervals for the mean obtained with t-test show that ENORA obtains lower values than Preselection with niches and NSGA-II. That is, the approximation sets obtained by ENORA are preferable to those of Preselection and those of NSGA-II under hypervolume indicator v . t-test is robust with samples which are greater than

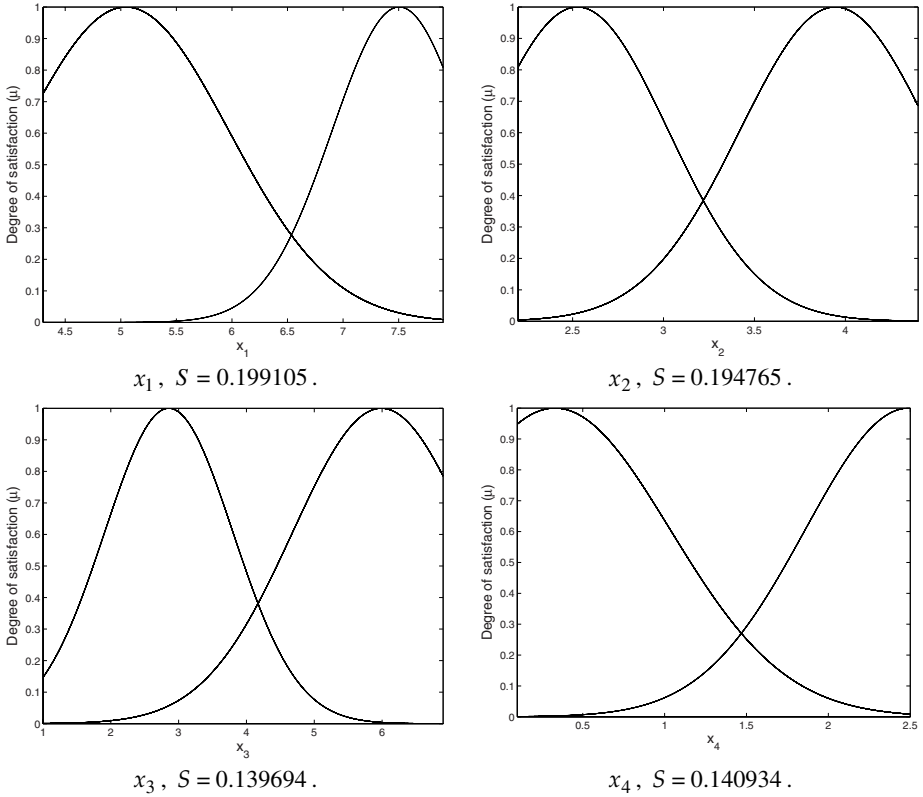


Fig. 1. Fuzzy sets of the fuzzy classification model obtained with ENORA

30 individuals, so the results are significant and we can conclude that there is statistical difference between the hypervolume values obtained by the algorithms. The box-plot showed in Figure 2 confirms the above conclusions.

The statistical analysis shows, therefore, that for the kind of multi-objective problems we are considering, Pareto search based on the space search partition in linear slots is most efficient than general search strategies exclusively based on diversity

Table 3. Statistics for the hypervolume obtained after 100 runs for the Iris data set classification problem. S.D. = Standard Deviation of Average, T.I. = Trusted Interval of Average (95%).

	Preselection with niches	NSGA	ENORA
Minimum	0.1297	0.1297	0.1216
Maximum	0.1472	0.2176	0.1472
Average	0.1373	0.1463	0.1351
S.D.	0.0034	0.0159	0.0048
Lower T.I.	0.1367	0.1431	0.1341
Upper T.I.	0.1380	0.1494	0.1360

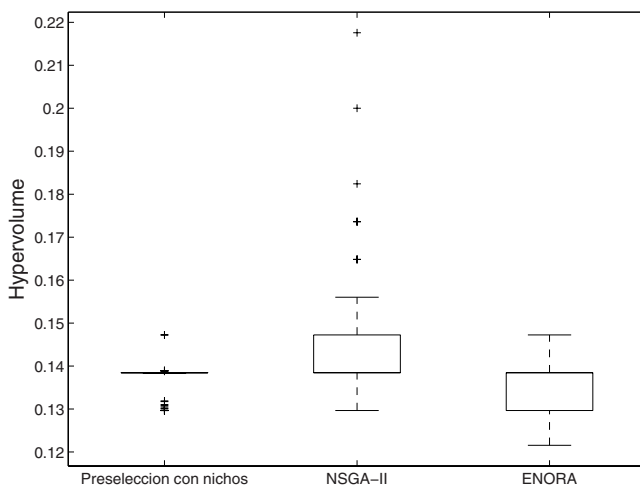


Fig. 2. Boxplot for the hypervolume obtained after 100 runs for the problem

functions, as in NSGA-II, or on diversity schemas with explicitly niching formation, as in Preselection with niches.

5 Conclusions

We have presented a new method for handling the Iris data classification problem. A multi-objective constrained optimization model for fuzzy classification is proposed and criteria such as accuracy, transparency and compactness have been taken into account.

Three multi-objective EAs (Preselection with niches, ENORA and NSGA-II) have been implemented in combination with rule simplification techniques. The proposed technique identifies a set of alternative solutions. Statistical tests have been performed over the hypervolume quality indicator to compare the algorithms and it has shown that, for this problem, ENORA obtains better results than Preselection with niches and NSGA-II algorithms.

Future improvements of the algorithms will be the automatic parameter tuning, and a next application of these techniques will be on medicine data.

This research is supported in part by MEC and FEDER under project PET2006 0406 and TIN 2006-15460-C04.

References

- [1] Casillas, J., Cordón, O., Herrera, F., Magdalena, L.: Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview. In: Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.) *Interpretability Issues in Fuzzy Modeling, Studies in Fuzziness and Soft Computing*, pp. 3–22. Springer, Heidelberg (2003)

- [2] Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, LTD, Chichester (2001)
- [3] Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
- [4] Gómez-Skarmeta, A.F., Jiménez, F.: Fuzzy modeling with hibrid systems. *Fuzzy Sets and Systems* 104, 199–208 (1999)
- [5] Gómez-Skarmeta, A.F., Jiménez, F., Sánchez, G.: Improving Interpretability in Approximative Fuzzy Models via Multiobjective Evolutionary Algorithms. *International Journal of Intelligent Systems* 22, 943–969 (2007)
- [6] Ishibuchi, H., Murata, T., Türksen, I.: Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets and Systems* 89, 135–150 (1997)
- [7] Ishibuchi, H., Nakashima, T., Murata, T.: Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* 29(5), 601–618 (1999)
- [8] Jiménez, F., Gómez-Skarmeta, A.F., Sánchez, G., Deb, K.: An evolutionary algorithm for constrained multi-objective optimization. In: *Proceedings IEEE World Congress on Evolutionary Computation* (2002)
- [9] Laumanns, M., Zitzler, E., Thiele, L.: On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) *EMO 2001*. LNCS, vol. 1993, pp. 181–196. Springer, Heidelberg (2001)
- [10] Setnes, M.: *Fuzzy Rule Base Simplification Using Similarity Measures*. M.Sc. thesis, Delft University of Technology, Delft, the Netherlands (1995)
- [11] Setnes, M., Babuska, R., Verbruggen, H.B.: Rule-based modeling: Precision and transparency. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications & Reviews* 28, 165–169 (1998)
- [12] Valente de Oliveira, J.: Semantic constraints for membership function optimization. *IEEE Transactions on Fuzzy Systems* 19(1), 128–138 (1999)
- [13] 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)

Cooperation between the Inference System and the Rule Base by Using Multiobjective Genetic Algorithms

Antonio Márquez, Francisco Alfredo Márquez, and Antonio Peregrín

Information Technologies Department
University of Huelva, 21071 Huelva, Spain
{amarquez,alfredo.marquez,peregrin}@dti.uhu.es

Abstract. This paper¹ presents an evolutionary Multiobjective learning model achieving positive synergy between the Inference System and the Rule Base in order to obtain simpler and still accurate linguistic fuzzy models by learning fuzzy inference operators and applying rule selection. The Fuzzy Rule Based Systems obtained in this way, have a better trade-off between interpretability and accuracy in linguistic fuzzy modeling applications.

Keywords: Linguistic fuzzy modeling, interpretability-accuracy trade-off, Multiobjective genetic algorithms, adaptive inference system, adaptive defuzzification, rule selection.

1 Introduction

Interpretability and accuracy are usually contradictory requirements in the design of linguistic fuzzy models (FMs). In practice, designers must find an adequate trade-off between them for the specific application, increasing the interest of this matter in the literature [1],[2].

Two important tasks in the design of a linguistic FM for a particular application are: The derivation of the linguistic rule base (RB) and the setup of the inference system and defuzzification method. In the framework of the trade-off between *interpretability* and *accuracy* in fuzzy modeling, adaptive inference system and defuzzification method acquired greater importance [3],[4].

Recently, the use of Multiobjective Evolutionary Algorithms (MOEA) has been applied to improve the aforementioned trade-off between interpretability and accuracy of linguistic fuzzy systems [5],[6],[7],[8]. Most of these works [5],[6] obtain the complete Pareto (the set of non-dominated solutions with different trade-off) by selecting or learning the set of rules better representing the example data, i.e., improving the system accuracy and decreasing the fuzzy RB system complexity. In [7],[8], authors also propose the tuning of the membership functions together with the rule selection to obtain simpler and still accurate linguistic FMs.

Following these ideas on the advantage of the use of parametric operators and the use of MOEAs to improve the trade-off between interpretability and accuracy, in this

¹ Supported by Projects TIN2005-08386-C05-01, P05-TIC-00531 and P07-TIC-03179.

work we present a MOEA to learn the fuzzy inference (including inference and defuzzification) and to perform rule selection for Mamdani linguistic fuzzy systems. The proposed model tries to achieve a positive synergy (this is the concept of cooperation we use) between the fuzzy operators and the RB to improve the accuracy at the same time that the RB is simplified to improve the interpretability.

In order to do this, Section 2 introduces the parametric fuzzy operators, Section 3 is devoted to describe the MOEA learning proposal, Section 4 develops an experimental study, and finally, Section 5 presents some concluding remarks.

2 Adaptive Fuzzy Operators

In this section we show the adaptive inference system as well as the adaptive defuzzification method used in our learning proposal.

2.1 Adaptive Inference System

Linguistic FRBSs for system modeling use IF - THEN rules of the following form:

$$R_i : \text{If } X_{i1} \text{ is } A_{i1} \text{ and } \dots \text{ and } X_{im} \text{ is } A_{im} \text{ then } Y \text{ is } B_i$$

with $i = 1$ to N , the number of rules of the RB, and with X_{i1} to X_{im} and Y being the input and output variables respectively, and with A_{i1} to A_{im} and B_i being the involved antecedents and consequent labels, respectively.

The expression of the Compositional Rule of Inference in fuzzy modeling with punctual fuzzification is the following one: $\mu_{B'}(y) = I(C(\mu_{A1}(x_1), \dots, \mu_{Am}(x_m)), \mu_B(y))$, where $\mu_B(\cdot)$ is the membership function of the inferred consequent, $I(\cdot)$ is the rule connective, $C(\cdot)$ is the conjunction operator, $\mu_{Ai}(x_i)$ are the values of the matching degree of each input of the system with the membership functions of the rule antecedents, and $\mu_B(\cdot)$ is the consequent of the rule.

The two components, conjunction ($C(\cdot)$) and rule connective ($I(\cdot)$) are suitable to be parameterized in order to adapt the inference system. Our previous studies in [3] show that the model based on the adaptive conjunction is a more valuable option than the one based on the adaptive rule connective. Hence, we selected the use of the adaptive conjunction in this study, in order to insert parameters in the inference system.

Taking into account the studies in [3], we have selected the Dubois adaptive t-norm with a separate connector for every rule, which expression is showed in (1).

$$T_{\text{Dubois}}(x, y, \alpha) = x \cdot y / \text{Max}(x, y, \alpha), \quad (0 \leq \alpha \leq 1) \quad (1)$$

2.2 Adaptive Defuzzification Interface

The most used methodology in practice, due to its fine performance, efficiency and easier implementation, is to apply the defuzzification function to every rule inferred fuzzy set (getting a characteristic value) and to compute then by a weighted average operator. This way of working is named FITA (First Infer, Then Aggregate) [9].

Attending to the studies developed in [10], in this work we consider to use a product functional term of the matching degree between the input variables and the rule

antecedent fuzzy sets (h_i), $f(h_i) = h_i \cdot \beta_i$ where β_i corresponds to one parameter for each rule R_i , $i=1$ to N , in the RB. The adaptive defuzzification formula selected is (2).

$$y_0 = \frac{\sum_i^N h_i \cdot \beta_i \cdot V_i}{\sum_i^N h_i \cdot \beta_i}, \quad (2)$$

where V_i represents a characteristic value of the fuzzy set inferred from rule R_i , the Maximum Value or the Gravity Center (GC), the one selected in this paper.

3 Cooperative Evolutionary Selection of Fuzzy Rules and Learning of Adaptive Fuzzy Operators with Multiobjective Algorithms

This section describes the basics of two of the most representative second generation MOEAs, SPEA2 [12] and NSGA-II [13], as two general propose MOEAs used in this work, and later the adaptations we propose to perform the cooperative adaptation of the fuzzy operators and fuzzy rule selection.

3.1 SPEA2 and NSGA-II

The SPEA2 algorithm [11] (*Strength Pareto Evolutionary Algorithm for multiobjective optimization*) is one of the most known techniques in the Multiobjective problem solving. It is characterized by the following two aspects: a *fitness* assignment strategy, that takes into account both dominating and dominated solutions for each individual, and a density function that is estimated employing the nearest neighbourhood, which guides the search more efficiently. A deeper description of the algorithm may be found in the aforementioned paper [11].

NSGA-II algorithm [12] is also another of the most well-known and frequently-used MOEAs for general multi-objective optimization in the literature. It is a parameterless approach with many interesting principles: a binary tournament selection based on a fast non-dominated sorting, an elitist strategy and a crowding distance method to estimate the diversity of a solution. As was commented before, a deeper description may be found in [12].

3.2 Questions Related to the MOEAs

The evolutionary model uses a chromosome with threefold coding scheme ($C_C+C_D+C_S$) where:

- C_C encodes the α parameters of the conjunction connective, that is N real coded parameters (genes) for each rule, R_i , of the linguistic RB. Each gene can take any value in the interval $[0,1]$, that is, among minimum and algebraic product.
- C_D encodes the β_i parameters of the defuzzification. They are N real coded parameters for each rule of the linguistic RB. Each gene can take any value in the interval $[0,10]$. This interval has been selected according with the study developed in [10]. It allows attenuation as well as enhancement of the matching degree.

- C_s encodes the rule selection. It is a binary string of N genes, each one representing a candidate rule of the initial RB. Depending on whether a rule is selected or not, values '1' or '0' are respectively assigned to the corresponding gene.

The initial population is randomly initialized with the exception of a single chromosome with the following setup:

- Fuzzy operators part: C_c with the N genes is initiated to 0 in order to make Dubois t-norm equivalent to Minimum t-norm initially. C_D also with the N genes is initiated to 1 with the objective to begin like the standard WCOA method.
- Rule selection part, C_s , with the N rules obtained by the WM-method [13], that is, with all the genes initialized to '1'.

The crossover operator employed for the fuzzy operators part is BLX-0.5 [14] while the one used for rule selection part is HUX [15].

The fitness based on the interpretability (using the number of rules) and the accuracy (using the error measure) must be minimized.

3.3 Improvements for SPEA2 and NSGA-II

This subsection is devoted to describe the two improvements we propose for SPEA2 [11] and NSGA-II [12] respectively to perform better the searching process we pretend, that is to guide the search towards the desired Pareto zone with high accuracy with the least possible number of rules.

The *Improved Accuracy NSGA-II* ($NSGA-II_{IA}$) algorithm:

We propose two main changes on the NSGA-II algorithm, based on the changes proposed by [8] with a few modifications to perform better with our problem, who has a larger real part comparatively. They are the following:

- We use a restarting mechanism carried out twice at 1/3 and 2/3 of the execution of the algorithm. Each time, the most accurate individual is maintained as the sole individual in the external population. The remaining individuals are reinitiated with the same rule configuration of the best individual and fuzzy operator parameters randomly generated within the corresponding variation intervals.
- In each of the three algorithm stages (before the first restart, after the first restart and after the second restart), the number of solutions in the external population considered to form the mating pool is progressively reduced, by focusing only on those with the best accuracy. To do that, the solutions are sorted from the best to the worst (considering accuracy as sorting criterion) and the number of solutions considered for selection is reduced progressively from 100% to 50 % at the beginning 1/3, since 75% to 50% in the middle, and since 66% to 50% at the end.

The *Improved Accuracy SPEA2* ($SPEA2_{IA}$) algorithm:

In order to guide the searching process of the SPEA2, we propose to employ a method called Guided Domination Approach [16], which gives priority to the accuracy objective through a weighted function of the objectives. Focusing the searching process we can reduce the effort of the search, and a better precision in the non-dominated solutions can be obtained, because the searching effort is concentrated in a reduced zone

of the Pareto, being the density of the obtained solutions higher. The weighted function of the objectives is defined in (3),

$$\Omega_i(f(x)) = f_i(x) + \sum_{j=1, j \neq i}^M a_{ij} f_j(x), \quad i=1,2,\dots, M \tag{3}$$

where a_{ij} is the amount of gain in the j -th objective function for a loss of one unit in the i -th objective function, and M being the number of objectives. The above set of equations require fixing the matrix a , which has a one in its diagonal elements. This method redefines the domination concept as follows: *A solution $x^{(1)}$ dominates another solutions $x^{(2)}$, if $\Omega_i(f(x^{(1)})) \leq \Omega_i(f(x^{(2)}))$ for all $i = 1, 2, \dots, M$, and the strict inequality is satisfied at least for one objective.*

Thus, if we have two ($M=2$) fitness functions, the two weighted functions are showed in (4).

$$\Omega_1(f_1, f_2) = f_1 + a_{12} f_2, \quad \Omega_2(f_1, f_2) = a_{21} f_1 + f_2 \tag{4}$$

4 Experimental Study

In order to analyze the proposed methods, we built several FMs using the learning methods showed in Table 1. WM method is considered as a reference. S and C-D are methods that perform rule selection and adaptation of fuzzy operators respectively. S + C-D means rule selection and fuzzy operators adaptation together. SPEA2, SPEA2_{IA}, NSGA-II and NSGA-II_{IA} are the methods that learn the fuzzy operators and the rule selection together, as was previously commented.

Table 1. Methods considered for comparisson

Ref.	Method	Description
[13]	WM	Wang & Mendel algorithm
[17]	WM + S	Rule Selection
[4],[10]	WM + C-D	Adaptive Fuzzy Operators
-	WM + S + C-D	Rule Selection and Adaptive Fuzzy Operators
[11]	SPEA2	SPEA2 Algorithm
-	SPEA2 _{IA}	Improved Accuracy SPEA2
[12]	NSGA-II	NSGA-II Algorithm
-	NSGA-II _{IA}	Improved Accuracy NSGA-II

4.1 Application Selected and Comparison Methodology

The fuzzy partition used for inputs and output has 5 labels. The application selected to test the evolutionary model is an electrical distribution problem [18] that has got a data set of 1059 cities with four input variables and a single output. The RB is composed of 65 linguistic rules achieved with the Wang and Mendel method [13].

We considered a 5-foder cross-validation model, i.e., 5 random partitions of the data each with 20% (4 of them with 211 examples, and one of them with 212

examples) and the combination of 4 of them (80%) as training, and the remaining one as test. We achieved a total of 30 trials for every evolutionary process, because for each one of the data partitions, the learning methods have been run 6 times. We show the averaged values of the medium square error (MSE) as a usual performance measure, computed considering the most accurate solution from each Pareto obtained for the Multiobjective algorithm. This way to work was also employed in [8] in order to compare the single objective methods with the Multiobjective ones based in to consider only the accuracy objective, letting us to see that the Pareto fronts are not only wide but also optimal, so similar solutions obtained with the WM + S + C-D should be included in the final Pareto. The MSE is computed with expression (5),

$$MSE(S)_B = \frac{1}{2} \sum_{k=1}^P (y_k - S(x_k))^2 / P \tag{5}$$

where S denotes the fuzzy model whose inference system uses the Dubois t-norm as conjunction operator showed in expression (2), inference operator minimum t-norm, and the adaptive defuzzificación method showed in expression (3). This measure uses a set of system evaluation data formed by P pairs of numerical data $Z_k = (x_k, y_k)$, $k=1, \dots, P$, with x_k being the values of the input variables, and with y_k being the corresponding values of the associated output variables.

The MOGAs population size was fixed to 200. The external population size of the SPEA2 and SPEA2_{IA} was 61.

The parameters a_{12} , a_{21} used for the SPEA2_{IA} have been determined after several test and fixed to 0 and 8 respectively, and give more importance to the accuracy.

4.2 Results and Analysis

The results obtained are shown in Table 2, where #R is the average number of rules, MSE_{tra} and MSE_{tst} are the average MSE for training and test respectively, σ is the standard deviation and *t-test* is the result of applying a *test t-student* (with 95 percent confidence), with the following interpretation: * represents the best average result; + means that the best result has better performance than that of the corresponding row.

Analysing the results obtained we can point out that NSGA-II_{IA} and SPEA2_{IA} shows a simmilar accuracy (slightly better in training) compared with the WM + S + C-D with a significative reduction in the number of rules, particularly for NSGA-II_{IA}. Modifications proposed in order to improve the accuracy for NSGA-II and SPEA2 let

Table 2. Results obtained

Method	#R	MSE _{tra}	σ_{tra}	t-test	MSE _{tst}	σ_{tst}	t-test
WM	65	56135.75	4321.42	+	56359.42	5238.57	+
WM + S	40.9	41517.01	4504.85	+	44064.67	906.64	+
WM + C-D	65	22561.77	3688.27	=	25492.77	830.76	*
WM + S + C-D	52.8	22640.95	2125.05	=	26444.43	854.95	+
SPEA2	38.4	24077.42	8225.82	+	29664.50	874.18	+
SPEA2 _{IA}	47.7	22450.72	3949.57	=	25562.74	850.07	=
NSGA-II	39.1	23303.50	6295.92	+	27920.42	877.83	+
NSGA-II _{IA}	41.1	22108.66	4695.30	*	26229.72	868.95	+

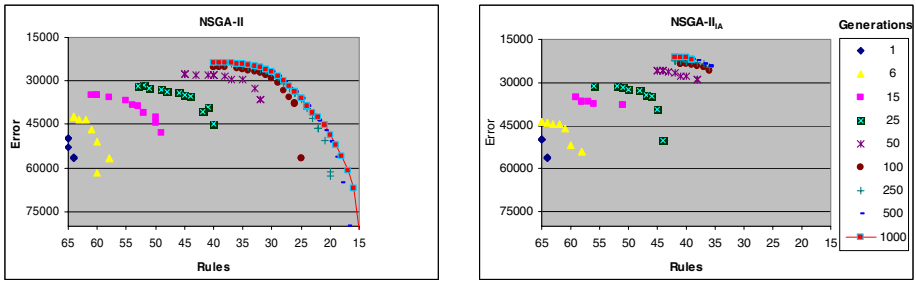


Fig. 1. Example of the Pareto front for NSGA-II and NSGA-II_A

both models improve their accuracy compared with the standard versions of both algorithms. Due to the adaptive fuzzy parameters search space is large, it is necessary to focus the searching process on the Pareto zone with higher accuracy to reach similar accuracy level than the mono-objective algorithm based on the accuracy. Figure 1 shows the difference between the searching process performed by the original NSGA-II (left side) versus the improved accuracy version (right side).

5 Conclusions

In the framework of the trade-off between accuracy and interpretability, the use of Multiobjective genetic algorithm gives a set of solutions with different level of conciliation of both features. In this work we have proposed a Multiobjective evolutionary learning model where the adaptive fuzzy operator parameters are learnt together with the rule base selection. This fact allows both elements to cooperate, improving the accuracy as well as the interpretability. Methodologies to focus the searching process in a specific zone of the Pareto have also been shown useful when an objective must prevail over the other.

References

1. Casillas, J., Cordón, O., Herrera, F., Magdalena, L.: Interpretability issues in fuzzy modeling. Springer, Heidelberg (2003)
2. Casillas, J., Cordón, O., Herrera, F., Magdalena, L.: Accuracy improvements in linguistic fuzzy modeling. Springer, Heidelberg (2003)
3. Alcalá-Fdez, J., Herrera, F., Márquez, F.A., Peregrín, A.: Increasing Fuzzy Rules Cooperation Based On Evolutionary Adaptive Inference Systems. *Int. J. of Intelligent Systems* 22(9), 1035–1064 (2007)
4. Márquez, F.A., Peregrín, A., Herrera, F.: Cooperative Evolutionary Learning of Fuzzy Rules and Parametric Aggregation Connectors for Mamdani Linguistic Fuzzy Systems. *IEEE Trans. on Fuzzy Syst.* 15(6), 1162–1178 (2007)
5. Ishibuchi, H., Yamamoto, T.: Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Syst.* 141, 59–88 (2004)

6. Narukawata, K., Nojima, Y., Ishibuchi, H.: Modification of evolutionary Multiobjective optimization algorithms for Multiobjective design of fuzzy rule-based classification systems. In: Proc. 2005 IEEE Int. Conf. on Fuzzy Syst., pp. 809–814. Reno (2005)
7. Alcalá, R., Alcalá-Fdez, J., Gacto, M.J., Herrera, F.: A Multi-Objective Evolutionary Algorithm for Rule Selection and Tuning on Fuzzy Rule-Based Systems. In: Proc. of the 16th IEEE Int. Conf. on Fuzzy Syst. FUZZ-IEEE 2007, London, pp. 1367–1372 (2007)
8. Alcalá, R., Gacto, M.J., Herrera, F., Alcalá-Fdez, J.: A Multi-Objective Genetic Algorithm for tuning and Rule Selection to obtain Accurate and Compact Linguistic Fuzzy Rule-Based Systems. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Syst.* 15(5), 539–557 (2007)
9. Buckley, J.J., Hayashi, Y.: Can approximate reasoning be consistent? *Fuzzy Sets and Syst.* 65(1), 13–18 (1994)
10. Cordon, O., Herrera, F., Márquez, F.A., Peregrín, A.: A Study on the Evolutionary Adaptive Defuzzification Methods in Fuzzy Modelling. *Int. J. of Hybrid Intelligent Syst.* 1(1), 36–48 (2004)
11. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2001)*, pp. 95–100 (2001)
12. 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)
13. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. *IEEE Trans. on Syst. Man, and Cybernetics* 22(6), 1414–1427 (1992)
14. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. *Found. of Genetic Algorithms* 2, 187–202 (1993)
15. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Found. of Genetic Algorithms* 1, 265–283 (1991)
16. Branke, J., Kaubler, T., Schmeck, H.: Guiding multi-objective evolutionary algorithms towards interesting region. Technical Report No. 399, Institute AIFB, University of Karlsruhe, Germany (2000)
17. Casillas, J., Cordon, O., del Jesus, M.J., Herrera, F.: Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Trans. on Fuzzy Syst.* 13(1), 13–29 (2005)
18. Cordon, O., Herrera, F., Sánchez, L.: Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Appl. Intell.* 10, 5–24 (1999)

Knowledge Base Learning of Linguistic Fuzzy Rule-Based Systems in a Multi-objective Evolutionary Framework

P. Ducange¹, R. Alcalá², F. Herrera², B. Lazzerini¹, and F. Marcelloni¹

¹ University of Pisa, Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica, Telecomunicazioni, 56122 Pisa, Italy

{p.ducange, b.lazzerini, f.marcelloni}@iet.unipi.it

² University of Granada, Dept. Computer Science and A.I., E-18071 Granada, Spain
{alcala, herrera}@decsai.ugr.es

Abstract. We propose a multi-objective evolutionary algorithm to generate a set of fuzzy rule-based systems with different trade-offs between accuracy and complexity. The novelty of our approach resides in performing concurrently learning of rules and learning of the membership functions which define the meanings of the labels used in the rules. To this aim, we represent membership functions by the linguistic 2-tuple scheme, which allows the symbolic translation of a label by considering only one parameter, and adopt an appropriate two-variable chromosome coding. Results achieved by using a modified version of PAES on a real problem confirm the effectiveness of our approach in increasing the accuracy and decreasing the complexity of the solutions in the approximated Pareto front with respect to the single objective-based approach.

Keywords: Multi-objective learning, accuracy-interpretability trade-off.

1 Introduction

Recent research on genetic fuzzy systems has focused on methods aimed at generating fuzzy rule-based systems (FRBSs) with an appropriate trade-off between accuracy and interpretability [3]. Indeed, in real applications, one is interested not only in improving performance but also in understanding the relation between inputs and output of the system. This is the well-known advantage of using FRBSs with respect to other regression or classification techniques. Unfortunately, accuracy and interpretability are often in conflict and therefore to satisfy both criteria to a large extent is not generally possible. Given its multi-criteria nature, this problem is typically tackled by using Multi-Objective Evolutionary Algorithms (MOEAs) [5]. In particular, MOEAs have been applied to generate Mamdani FRBSs mainly for classification problems ([9], [10]) and rarely for regression problems ([2], [4]). To the best of our knowledge, no approach proposed in the literature performs concurrently learning of rules (Rule Base –RB– identification) and tuning of the meanings of the linguistic values used in the rules (Data Base –DB– learning). Only a post-processing method has been recently proposed to perform rule selection together with a tuning of the Membership Function (MF) parameters starting from an initial Knowledge Base (KB) [2].

In this paper, we propose a technique to generate a set of Mamdani FRBSs with different optimal trade-offs between accuracy and complexity by performing both the RB identification and the DB learning in the same multi-objective framework. To this aim, we adopt the linguistic 2-tuple representation introduced in ([1], [8]) and the RB identification approach based on a purposely adapted version of PAES proposed in [4]. The 2-tuple representation allows the lateral displacement of a label by only considering one parameter rather than the classical three parameters. On the other hand, the version of PAES has proved to be very effective in RB identification. We tested the proposed approach on a real world regression problem. Results confirm a positive synergy between the RB identification and the DB learning, increasing the accuracy and decreasing the complexity of the solutions in the approximated Pareto front with respect to the single objective-based approach.

This paper is arranged as follows. Next section explains how we codify the linguistic RB and presents the linguistic 2-tuples. Section 3 describes the proposed MOEA. Section 4 shows the experiments and Section 5 points out some conclusions.

2 Preliminaries

This section presents the type of rule (Mamdani) and how we represent the RB in order to better describe the proposed approach. Besides, it also introduces the linguistic 2-tuple representation used for the learning of MFs.

2.1 Linguistic Fuzzy Rule Base Representation

Let $X = \{X_1, \dots, X_F\}$ be the set of input variables and $X_{(F+1)}$ be the output variable. Let U_f be the universe of the f -th variable. Let $P_f = \{A_{f,1}, \dots, A_{f,T_f}\}$ be a fuzzy partition with T_f linguistic terms (labels) on the f -th variable. The m -th rule ($m = 1, \dots, M$) of a Mamdani fuzzy system can be expressed as:

$$R^m: \text{If } X_1 \text{ is } A_{1,j_1^m} \text{ and } \dots \text{ and } X_F \text{ is } A_{F,j_F^m} \text{ Then } X_{(F+1)} \text{ is } A_{(F+1),j_{(F+1)}^m}$$

where $j_f^m \in [1, T_f]$ identifies the index of the label (among the T_f linguistic terms of partition P_f), which has been selected for X_f in rule R_m . Then, a linguistic RB could be completely described by the following matrix, $J \in \mathfrak{R}^{M \times (F+1)}$,

$$J = \begin{bmatrix} j_1^1 & \dots & j_F^1 & j_{(F+1)}^1 \\ \dots & \dots & \dots & \dots \\ j_1^m & \dots & j_F^m & j_{(F+1)}^m \\ \dots & \dots & \dots & \dots \\ j_1^M & \dots & j_F^M & j_{(F+1)}^M \end{bmatrix}$$

2.2 The Linguistic 2-Tuple Representation

In [1], a new tuning model of FRBSs was proposed considering the linguistic 2-tuples representation scheme introduced in [8], which allows the lateral displacement of the support of a label and maintains the interpretability in a good level. This proposal introduces a new model for rule representation based on the concept of symbolic translation [8]. The symbolic translation of a label is a number in $[-0.5, 0.5)$, expressing this interval the domain of a label when it is moving between its two adjacent lateral labels (see Figure 1.a). Let us consider a set of labels S representing a fuzzy partition. Formally, to represent the symbolic translation of a label in S we have the 2-tuple,

$$(s_i, \alpha_i), s_i \in S, \alpha_i \in [-0.5, 0.5).$$

The symbolic translation of a label involves the lateral displacement of its associated MF. Figure 1 shows the symbolic translation of a label represented by the pair $(s_2, -0.3)$ together with the associated lateral displacement.

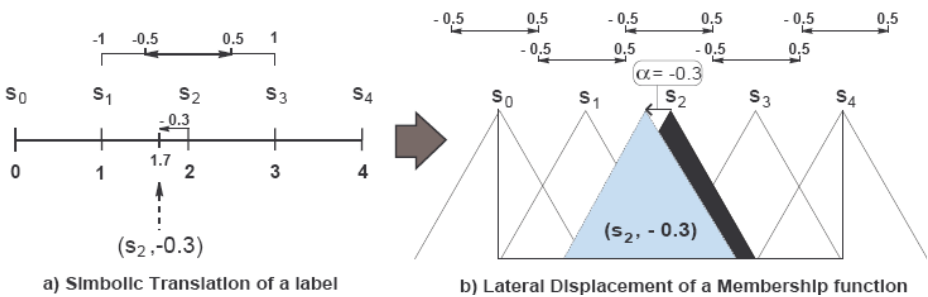


Fig. 1. Symbolic Translation of a Label and Lateral Displacement of the involved MF

In [1], two different rule representation approaches were proposed, a global and a local approach. In this case, the learning is applied to the level of linguistic partitions (global approach). The pair (X_i, label) takes the same α value in all the rules where it is considered, i.e., a global collection of 2-tuples is considered by all the fuzzy rules. Notice that from the parameters α applied to each label we could obtain the equivalent triangular MFs, by which a FRBS based on linguistic 2-tuples could be represented as a classic Mamdani FRBS. See [1] to obtain more details on this approach.

3 A Multi-objective Approach for Learning KBs

To obtain a set of KBs with different trade-offs between accuracy and interpretability we are going to obtain together the J matrix representing a linguistic RB and the displacement parameters of the corresponding MFs in the associated DB. To do this, we exploit a specific MOEA considering a double coding scheme (coding of rules and coding of parameters). We adopted a modified version of the (2+2)PAES [11], denoted *modified PAES* [4]. Unlike classical (2+2)PAES, which uses only mutation to generate new candidate solutions, *modified PAES* exploits the one-point crossover and two appropriately defined mutation operators. The authors experimentally

verified that, in these kinds of problems (linguistic modeling), crossover helps create an approximation of the Pareto front where solutions are uniformly distributed. The crossover and the mutation operators were proposed in [4] to evolve linguistic RBs and should be combined here with appropriate operators for the real coding part. In the next, we describe the double coding scheme, the objectives considered and the specific operators applied to this concrete problem.

3.1 Coding Scheme

A double coding scheme ($C = C_{RB} + C_{DB}$) to represent both parts, RB and DB, is used:

- Rules (C_{RB}): The RB of a FRBS is represented by the integer part of the chromosome shown in Figure 2, where $j_f^m \in [0, T_f]$ identifies the index of the label which has been selected for variable X_f in rule R^m (among the T_f linguistic terms of partition P_f plus 0 if the antecedent condition is removed).
- Translation parameters (C_{DB}): This part is the joint of the α parameters associated to the labels of each fuzzy partition P_f . Then, the DB of an FRBS is represented by the part with real coding of the chromosome shown in Figure 2 (where each gene is the translation parameter of each label).

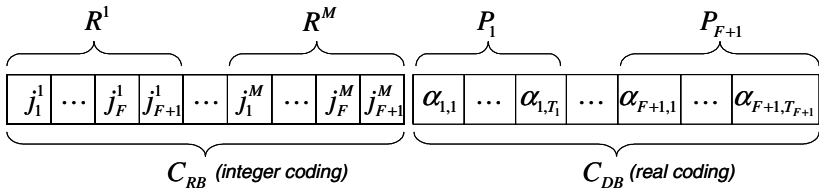


Fig. 2. Chromosome representation

3.2 Objectives

Two objectives are minimized: the total number of antecedent conditions in the obtained rules (interpretability) and the Mean Squared Error (accuracy),

$$MSE = \frac{1}{2 \cdot |E|} \sum_{l=1}^{|E|} (F(x^l) - y^l)^2$$

with $|E|$ being the data set size, $F(x^l)$ being the output obtained from the FRBS decoded from such chromosome when the l -th example is considered and y^l being the known desired output. The fuzzy inference system considered to obtain $F(x^l)$ is the *center of gravity weighted by the matching* strategy as defuzzification operator and the *minimum t-norm* as implication and conjunctive operators.

3.3 Genetic Operators and Their Application

As regards the crossover operator we consider the classical one-point crossover for the C_{RB} part combined with the BLX-0.5 [7] operator for the C_{DB} part. Let C^1 and C^2

be the two current solutions. The one-point crossover operator cuts the chromosomes C_{RB}^1 and C_{RB}^2 at some chosen common gene and swaps the resulting sub-chromosomes. The common gene is chosen by extracting randomly a number in $[M_{min}, \rho_{min}]$, where M_{min} is the minimum number of rules, which must be present in a RB, and ρ_{min} is the minimum number of rules in C_{RB}^1 and C_{RB}^2 . Finally, the BLX-0.5 operator is applied two times considering C_{DB}^1 and C_{DB}^2 in order to obtain the C_{DB} parts of both offspring.

On the other hand, three different mutation operators can be applied (the first two for the C_{RB} part and the other for the C_{DB} part). The first mutation operator adds γ rules to the RB, where γ is randomly chosen in $[1, \gamma_{max}]$. The upper bound γ_{max} is fixed by the user. If $\gamma + M > M_{max}$, then $\gamma = M_{max} - M$, where M_{max} is the maximum number of rules which must be present in a RB. For each rule R^m added to the chromosome, we generate a random number $t \in [1, F]$, which indicates the number of input variables used in the antecedent of the rule. Then, we generate t natural random numbers between 1 and F to determine the input variables which compose the antecedent part of the rule. Finally, for each selected input variable X_f , we generate a random natural number j_f^m between 1 and T_f , which determines the linguistic label A_{f, j_f^m} to be used

in the corresponding input variable of rule R^m . To select the consequent linguistic term, a random number between 1 and $T_{(F+1)}$ is generated. The second mutation operator randomly changes δ elements of matrix J . The number is randomly generated in $[1, \delta_{max}]$. The upper bound δ_{max} is fixed by the user. For each element to be modified, a number is randomly generated in $[0, T_f]$, where f is the input variable corresponding to the selected matrix element. The third mutation operator simply changes a gene value at random in the C_{DB} part.

The probability of applying the crossover operator is 0.5. When the application of the crossover operator is selected, the RB mutation is applied with probability 0.01; otherwise this mutation is always applied. When the application of the RB mutation is selected, the first mutation operator is applied with probability 0.55 and the second mutation operator is applied when the first is not applied. The third mutation operator (DB mutation) is always applied with probability 0.2.

4 Experiments

To evaluate the usefulness of the method proposed, we have considered a real-world problem [6] with 4 input variables that consists of estimating the maintenance costs of medium voltage lines in a town. Two different algorithms have been analyzed considering the same codification and genetic operators (with and without DB learning). They are the modified (2+2)PAES and a Single Objective based Genetic Algorithm (SOGA) with the MSE as the only objective. Methods considered for the experiments are shown in Table 1. Both algorithms are executed as explained in [4].

The initial fuzzy partitions are comprised by *five linguistic terms* with equally distributed triangular MFs. We set the maximum/minimum number of rules in a RB to 30/5, the minimum number of antecedent conditions in a RB to 1, the maximum number of evaluations to 200000 and the population/archive size to 64.

Table 1. Methods Considered for Comparison

Method	Ref.	Description
Algorithms without learning of MFs		
$SOGA_{RB}$	[4]	RB Learning by SOGA
$PAES_{RB}$	[4]	RB Learning by PAES [†]
Algorithms with learning of MFs		
$SOGA_{KB}$	–	RB+DB Learning by SOGA
$PAES_{KB}$	–	RB+DB Learning by PAES [†]

[†] modified version of PAES.

4.1 Problem Description and Experiments

Estimating the maintenance costs of the medium voltage electrical network in a town [6] is a complex but interesting problem. Since a direct measure is very difficult to obtain, it is useful to consider models. These estimations allow electrical companies to justify their expenses. Moreover, the model must be able to explain how a specific value is computed for a certain town. Our objective will be to relate the *maintenance costs of the medium voltage lines* with the following four variables: *sum of the lengths of all streets in the town*, *total area of the town*, *area that is occupied by buildings*, and *energy supply to the town*. We will deal with estimations of minimum maintenance costs based on a model of the optimal electrical network for a town in a sample of 1,059 towns.

To develop the experiments, we consider a *5-fold cross-validation model*, i.e., 5 random partitions of data each with 20%, and the combination of 4 of them (80%) as training and the remaining one as test. For each one of the 5 data partitions, the studied methods have been run 6 times, showing for each problem the averaged results of a total of 30 runs. In the case of methods with multi-objective approach, the averaged values have been calculated considering the most accurate solution from each Pareto obtained. In this way, the multi-objective algorithms can be statistically compared with the single objective based methods taking into account that at least one solution with equivalent accuracy should be obtained as a part of the final Pareto fronts.

The results obtained by the analyzed methods are shown in Table 2, where $\#R$ stands for the number of rules, $MSE_{train/test}$ for the averaged error obtained over the training/test data, σ for their respective standard deviations and t for the results of applying a *test t-student* (with 95 percent confidence) in order to ascertain whether differences in the performance of the best results are significant when compared with that of the other algorithms in the table. The interpretation of this column is:

- * represents the best averaged result.
- + means that the best result performs better than that of the corresponding row.

4.2 Results and Analysis

Analysing the results shown in table 2 we can highlight the following facts:

- Learning the definition parameters of the MFs within the evolutionary identification process allows us to obtain solutions with better accuracy in the case of both multi-objective and single-objective algorithms.
- The multi-objective approaches outperform the single-objective approach both on training and test sets, with a lower standard deviation.

Table 2. Results obtained by the studied methods

Method	#R	MSE _{tra}	σ_{tra}	t-test	MSE _{test}	σ_{test}	t-test
SOGA _{RB}	30.0	40618	24594	+	46827	26846	+
PAES _{RB}	29.9	16419	4135	+	18343	4602	+
SOGA _{KB}	30.0	34252	22161	+	42811	28839	+
PAES _{KB}	29.7	11921	3099	*	13528	3712	*

In Figure 3, we show a representative Pareto front approximation, both on training and test sets, provided by PAES_{KB}. Solutions in Figure 3 are represented in the Complexity-Error plane, i.e., the actual plane of the optimized objectives. Besides, we also show the same solutions represented in the Rules-Error plane. Obviously, in the latter graph, we can notice solutions with different errors but the same number of rules. These solutions are associated with different number of total conditions in the RB, i.e., different complexities. It is interesting to notice that the Pareto front is widely spread and good trade-offs between complexity and accuracy are obtained. Finally, the non-dominated FRBSs have also good generalization ability on the test set.

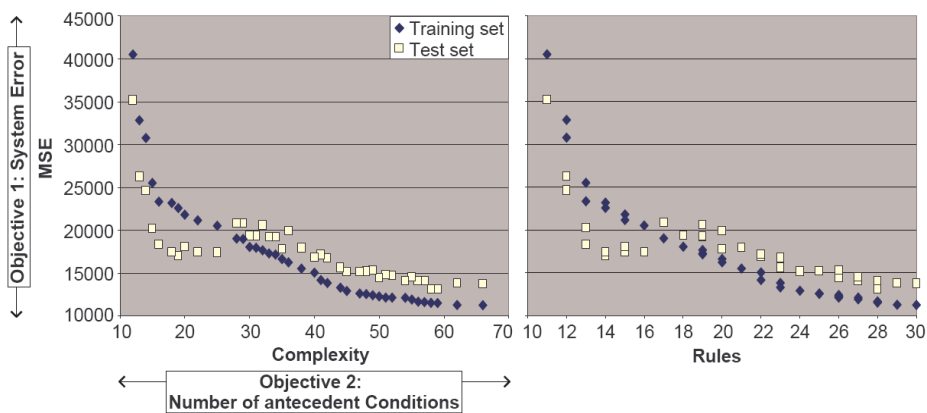


Fig. 3. A Pareto front approximation obtained from PAES_{KB} (Complexity-Error plane and Rules-Error plane)

5 Concluding Remarks

In this contribution, we propose a multi-objective approach to learn a set of KBs with different trade-offs between accuracy and interpretability (measured as the number of rule antecedent conditions). The proposed method is able to learn RBs together with the MF parameters of the associated linguistic labels in the corresponding DB: this allows taking the existing dependencies into account but involves a huge search space. In order to better handle the huge search space it represents, we exploit the positive synergy existing between an interesting RB identification mechanism and the linguistic 2-tuples representation model for the learning of MFs. Results have confirmed this positive synergy and that, by considering a multi-objective framework, it

is possible to obtain simpler FRBSs and with even better accuracy than the ones provided by considering the accuracy as the only objective.

References

1. Alcalá, R., Alcalá-Fdez, J., Herrera, F.: A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. *IEEE T. Fuzzy Systems* 15(4), 616–635 (2007)
2. Alcalá, R., Gacto, M.J., Herrera, F., Alcalá-Fdez, J.: A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems* 15(5), 539–557 (2007)
3. Casillas, J., Cordon, O., Herrera, F., Magdalena, L. (eds.): Accuracy improvements in linguistic fuzzy modelling. *Studies in Fuzziness and Soft Computing*, vol. 129. Springer, Heidelberg (2003)
4. Cococcioni, M., Ducange, P., Lazzerini, B., Marcelloni, F.: A Pareto-based multi-objective evolutionary approach to the identification of mamdani fuzzy systems. *Soft Computing* 11, 1013–1031 (2007)
5. Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, Dordrecht (2002)
6. Cordon, O., Herrera, F., Sánchez, L.: Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence* 10, 5–24 (1999)
7. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202 (1993)
8. Herrera, F., Martínez, L.: A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE T. Fuzzy Systems* 8(6), 746–752 (2000)
9. Ishibuchi, H., Yamamoto, T.: Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems* 141(1), 59–88 (2004)
10. Ishibuchi, H., Nojima, Y.: Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *Int. J. of Approximate Reasoning* 44(1), 4–31 (2007)
11. Knowles, J.D., Corne, D.W.: Approximating the non dominated front using the Pareto archived evolution strategy. *Evolutionary Computation* 8(2), 149–172 (2000)

Effects of Diversity Measures on the Design of Ensemble Classifiers by Multiobjective Genetic Fuzzy Rule Selection with a Multi-classifier Coding Scheme

Yusuke Nojima and Hisao Ishibuchi

Dept. of Computer Science and Intelligent Systems, Graduate School of Engineering,
Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
{nojima,hisaoi}@cs.osakafu-u.ac.jp

Abstract. We have already proposed multiobjective genetic fuzzy rule selection with a multi-classifier coding scheme for the design of ensemble classifiers. An entropy-based diversity measure was used as an objective to be maximized for increasing the diversity of base classifiers in an ensemble. In this paper, we examine the use of other diversity measures in the design of ensemble classifiers. Experimental results show that the choice of a diversity measure has a large effect on the performance of designed ensemble classifiers.

Keywords: Genetic Fuzzy Systems, Ensemble Classifiers, Diversity Measures.

1 Introduction

The use of multiple classifiers as an ensemble is a promising approach to the design of reliable classifiers [1, 2]. In the design of ensemble classifiers, it is essential to generate a number of base classifiers with a large diversity. Some studies [1, 2] rely on diversity maintenance mechanisms to generate base classifiers while others [3, 4] use heuristic measures to evaluate the diversity of base classifiers. Such a heuristic measure can be incorporated into evolutionary approaches for generating a number of base classifiers. Evolutionary multiobjective optimization (EMO) algorithms have been frequently used in the design of ensemble classifiers where the accuracy and the diversity of base classifiers are simultaneously optimized [5-9].

In our former studies [8, 9], we have proposed multiobjective genetic fuzzy rule selection with a multi-classifier coding scheme for the design of ensemble classifiers. Genetic fuzzy rule selection was originally proposed in [10] where a small number of fuzzy rules were selected from a large number of candidate rules to generate a fuzzy rule-based classifier. This method was extended to multiobjective formulations in [11, 12]. A binary string was used to represent a fuzzy rule-based classifier (i.e., a subset of candidate rules) in these studies [10-12]. The use of binary strings was generalized by introducing a multi-classifier coding scheme in our former studies [8, 9] where an integer string was used to represent an ensemble classifier (i.e., an ensemble of disjoint subsets of candidate rules). Each ensemble was evaluated by its classification accuracy

and the entropy of outputs from its base classifiers. In this paper, we examine the use of various diversity measures [4] in our evolutionary ensemble method with the multi-classifier coding scheme.

In this paper, we first explain our EMO-based approach to the design of ensemble classifiers. Next we examine the use of various diversity measures in our approach in Section 3. Finally we conclude this paper in Section 4.

2 Fuzzy Ensemble Classifiers by Genetic Fuzzy Rule Selection

Our ensemble design method consists of two stages: candidate fuzzy rule extraction and ensemble classifier optimization. In the first stage (i.e., candidate fuzzy rule extraction), a prespecified number of fuzzy rules are extracted in a heuristic manner.

Let us assume that we have m training patterns $\mathbf{x}_p = (x_{p1}, \dots, x_{pn})$, $p = 1, 2, \dots, m$ from M classes where x_{pi} is the attribute value of the p th training pattern for the i th attribute ($i = 1, 2, \dots, n$). We also assume that all attribute values have already been normalized into real numbers in the unit interval $[0, 1]$. For our n -dimensional M -class problem, we use fuzzy rules of the following form:

$$\text{Rule } R_q: \text{ If } x_1 \text{ is } A_{q1} \text{ and } \dots \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \tag{1}$$

where R_q is the label of the q th rule, A_{qi} is an antecedent fuzzy set, C_q is a class label, and CF_q is a rule weight. As antecedent fuzzy sets, we use the 14 fuzzy sets in Fig. 1 and “don’t care” represented by the unit interval $[0, 1]$. Thus the total number of combinations of the n antecedent fuzzy sets in (1) is 15^n . Among these possible rules, we examine only short fuzzy rules with a small number of antecedent conditions (i.e., short fuzzy rules with many don’t care conditions) to generate candidate rules.

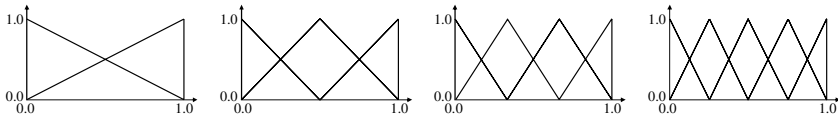


Fig. 1. Fourteen antecedent fuzzy sets used in this paper

For determining the consequent class C_q and the rule weight CF_q , first the confidence of the fuzzy rule “ $\mathbf{A}_q \Rightarrow \text{Class } h$ ” is calculated for each class h as follows:

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{\sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{\sum_{p=1}^m \mu_{\mathbf{A}_q}(\mathbf{x}_p)}, \quad h = 1, 2, \dots, M, \tag{2}$$

where $\mu_{\mathbf{A}_q}(\mathbf{x}_p)$ is the compatibility grade of \mathbf{A}_q . The product operator is used to calculate the compatibility grade of each training pattern \mathbf{x}_p with the antecedent part \mathbf{A}_q of the fuzzy rule R_q in (1). The consequent class C_q is specified as the class with the maximum confidence:

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max_h \{c(\mathbf{A}_q \Rightarrow \text{Class } h)\}. \quad (3)$$

The rule weight is specified by the difference between the confidence of the consequent class and the sum of the confidences of the other classes as follows:

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^M c(\mathbf{A}_q \Rightarrow \text{Class } h). \quad (4)$$

In this manner, the consequent class and the rule weight of each fuzzy rule can be easily determined from training patterns (for details, see [13]).

Using the above-mentioned heuristic manner, we can generate a large number of short fuzzy rules as candidate rules in multiobjective fuzzy rule selection. In our former studies [8, 9], we used the SLAVE criterion [14] to prescreen candidate rules. The use of this criterion, however, tends to exclude many specific and accurate rules. In this paper, we use two threshold values: the minimum *support* and the minimum *confidence*. The fuzzy version of support can be written as

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{1}{m} \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p). \quad (5)$$

We exclude fuzzy rules that do not satisfy these two thresholds. Among short fuzzy rules satisfying these two thresholds, we choose a prespecified number of the best candidate rules for each class with respect to $s(R_q) \cdot c(R_q)$.

2.1 Multiobjective Genetic Fuzzy Rule Selection

In the second stage, we use the NSGA-II algorithm [15] to generate non-dominated ensemble classifiers from candidate rules extracted in the first stage. To represent an ensemble classifier as a string, we use a multi-classifier coding scheme. Each ensemble classifier is represented by an integer string of length N as $S = s_1 s_2 \cdots s_N$ where N is the number of candidate rules and s_j is an integer (s_j denotes the base classifier to which the j th candidate rule is included). For example, a string “2110201033” denotes an ensemble classifier with three base classifiers: Classifier 1 with R_2, R_3, R_7 , Classifier 2 with R_1, R_5 , and Classifier 3 with R_9, R_{10} . As shown in this example, the j th candidate rule is not used in any base classifier when $s_j = 0$.

Each ensemble classifier S is evaluated by its accuracy and the diversity of its base classifiers. We use the following two objective functions:

$f_1(S)$: Classification rate of the ensemble classifier S on training patterns,

$f_2(S)$: A diversity measure to evaluate the diversity of the base classifiers in S .

Let S_i be the i th base classifier in the ensemble classifier S . When an input pattern \mathbf{x}_p is to be classified by S_i , a single winner rule R_w is chosen from S_i as

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q \mid R_q \in S_i\}. \quad (6)$$

When multiple rules with different consequent classes have the same maximum value in (6), the classification of \mathbf{x}_p by S_i is rejected. The classification of \mathbf{x}_p by S_i is also rejected when there are no compatible rules with \mathbf{x}_p in S_i . The final classification

of \mathbf{x}_p by the ensemble S is performed through the majority voting (strictly speaking “plurality voting”) based on the classification result by each base classifier S_j . When multiple classes have the same maximum number of votes, the final classification of \mathbf{x}_p is rejected in this paper whereas random tie-break was employed in [8, 9].

2.2 Various Diversity Measures of Base Classifiers

Various diversity measures have been proposed in the literature [3, 4]. We can use such a diversity measure as the second objective $f_2(S)$ in the previous subsection. In this subsection, we explain the entropy-based diversity measure in our former studies [8, 9] and five other diversity measures in [4], which are examined in the next section.

Entropy based-diversity measure (E): This diversity measure was used in [8, 9] where the entropy E of classification results by base classifiers was calculated as

$$E = \frac{1}{m} \sum_{p=1}^m \sum_{c=1}^M (-P_{pc} \log P_{pc}) . \tag{7}$$

In this formulation, m is the number of patterns, M is the number of classes, and P_{pc} is the ratio of base classifiers which classify the pattern \mathbf{x}_p as Class c . Let us assume that the number of base classifiers in an ensemble classifier is $N_{\text{classifier}}$. If three base classifiers classify the first pattern as Class 1, P_{11} is calculated as $3/N_{\text{classifier}}$. Of course, a larger value of the entropy E means a larger diversity of base classifiers.

Disagreement measure (dis): The disagreement measure between two base classifiers S_j and S_k is defined as

$$dis_{j,k} = \frac{n(1, -1) + n(-1, 1)}{n(1, 1) + n(-1, 1) + n(1, -1) + n(-1, -1)} , \tag{8}$$

where $n(1, -1)$ is the number of training patterns that are correctly classified by S_j and misclassified by S_k (i.e., “1” and “-1” represent *true* and *false*, respectively). $n(1, -1)$, $n(1, 1)$ and $n(-1, -1)$ are defined in the same manner as $n(1, -1)$. The disagreement measure among more than two base classifiers is defined as

$$dis = \frac{2}{L(L-1)} \sum_{j=1}^{L-1} \sum_{k=j+1}^L dis_{j,k} , \tag{9}$$

where L is the number of base classifiers. A larger value of this disagreement measure means a larger diversity of base classifiers.

Double-fault measure (DF): The double-fault measure for two base classifiers S_j and S_k is the ratio of training patterns that are misclassified by both classifiers:

$$DF_{j,k} = \frac{n(-1, -1)}{n(1, 1) + n(-1, 1) + n(1, -1) + n(-1, -1)} . \tag{10}$$

The double-fault measure for more than two base classifiers is calculated as

$$DF = \frac{2}{mL(L-1)} \sum_{j=1}^{L-1} \sum_{k=j+1}^L DF_{j,k} , \tag{11}$$

where m is the number of training patterns, and L is the number of base classifiers. A smaller value of the double-fault measure means a large diversity of base classifiers.

Kohavi-Wolpert variance (KW): The Kohavi-Wolpert variance is calculated as

$$KW = \frac{1}{mL^2} \sum_{i=1}^m l_i(L-l_i), \quad (12)$$

where l_i is the number of base classifiers that misclassify the training pattern \mathbf{x}_i . A larger value of this measure means a larger diversity of base classifiers.

Measurement of interrater agreement (κ): This measure is calculated as

$$\kappa = 1 - \frac{\sum_{i=1}^m l_i(L-l_i)}{mL(L-1)P(1-P)}, \quad (13)$$

where P is the average classification rate over L base classifiers. That is,

$$P = 1 - \frac{\sum_{i=1}^m l_i}{mL}. \quad (14)$$

A smaller value of κ means a larger diversity of base classifiers.

Measure of difficulty ($diff$): The measure of difficulty is defined as the variance of the average classification rate V_i of L base classifiers for each training pattern \mathbf{x}_p :

$$diff = \text{var}(V_i), \quad (15)$$

where $V_i = (L-l_i)/L$ for each pattern \mathbf{x}_i . A smaller value of this difficulty measure means a larger diversity of base classifiers.

3 Computational Experiments

Our computational experiments were performed for five data sets in Table 1. Attribute values were normalized to real numbers in the unit interval $[0, 1]$. We used the 10-fold cross validation procedure for performance evaluation of ensemble classifiers.

In the candidate rule extraction phase, we specified the minimum confidence and support, and the upper bound on the number of antecedent conditions as 0.6, 0.01, and

Table 1. Five data sets from the UCI Machine Learning Repository

Data set	Attributes	Patterns	Classes
Breast W	9	683*	2
Diabetes	8	768	2
Glass	9	214	6
Iris	4	150	3
Sonar	60	208	2

* Incomplete patterns with missing values are not included.

three (two only for the Sonar data), respectively. From qualified fuzzy rules, we chose the best 300 rules for each class using the product of support and confidence.

In the genetic rule selection phase, we used the following parameters in NSGA-II:

Population size: 200 strings,

Number of classifiers: 5,

Crossover probability: 0.8,

Biased mutation probabilities:

$$p_m(0 \rightarrow a) = 1/300M \text{ and } p_m(a \rightarrow 0) = 0.1 \text{ where } a \in \{1, 2, \dots, 5\},$$

Stopping condition: 5000 generations.

Experimental results by a single run of NSGA-II for the Breast W data set are summarized in Fig. 2. Figure 2 shows the relation between the classification rate of each non-dominated ensemble classifier and its value of each diversity measure. In Fig. 2, a large number of non-dominated ensemble classifiers were obtained by a single run of NSGA-II. We can observe a clear tradeoff relation between the accuracy of each ensemble classifier for training patterns and the diversity of its base classifiers. In some plots (e.g., Fig. 2 (a)), we can observe the overfitting of ensemble classifiers. This may suggest the existence of an appropriate value for each diversity measure.

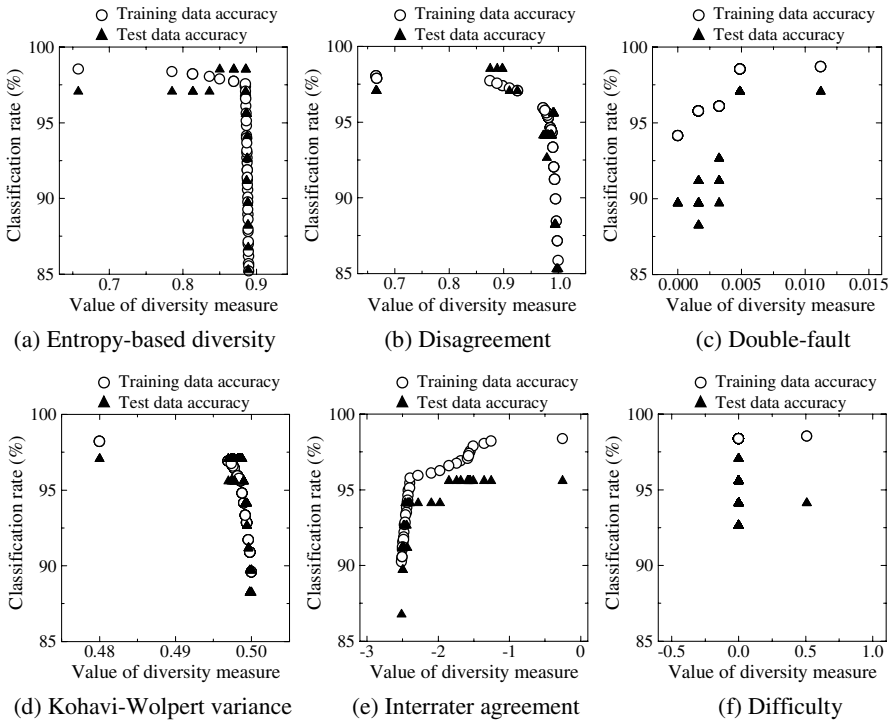


Fig. 2. The accuracy of each ensemble classifier and its diversity (Breast W data)

Table 2 shows the average test data accuracy of the ensemble classifier with the best *training* data accuracy. The highest classification rate on test data for each data set was obtained in each row in Table 2 by a different diversity measure. On the other hand, Table 3 shows the average test data accuracy of the ensemble classifier with the best *test* data accuracy. In almost all runs of NSGA-II, the ensemble classifier with the highest training data accuracy was different from the ensemble classifier with the highest test data accuracy. By the diversity measure based on interrater agreement (κ), the best results on test data were obtained for many data sets.

Table 4 shows the standard deviation of the normalized values of each diversity measure. Before calculating the standard deviation, values of each diversity measure were normalized into real numbers in [0, 1] by its minimum and maximum values over ten runs in the 10-fold cross-validation procedure. When the standard deviation is small, the diversity measure seems to be reliable in the design of ensemble classifiers. In the experimental results in Table 4, the measurement of interrater agreement

Table 2. Average classification rate (%) on test data of the ensemble classifier with the best training data accuracy

	<i>E</i>	<i>dis</i>	<i>DF</i>	<i>KW</i>	κ	<i>diff</i>
Breast W	96.04	95.90	96.48	94.87	95.61	95.75
Diabetes	75.13	74.35	74.62	75.13	74.48	74.74
Glass	62.12	59.26	62.12	61.21	63.48	60.69
Iris	96.00	94.00	94.67	96.67	96.00	95.33
Sonar	74.90	72.55	73.00	73.88	72.07	75.31

Table 3. Average classification rate (%) on test data of the ensemble classifier with the best test data accuracy

	<i>E</i>	<i>dis</i>	<i>DF</i>	<i>KW</i>	κ	<i>diff</i>
Breast W	97.07	97.21	97.21	96.33	97.22	96.48
Diabetes	77.21	76.44	76.18	76.70	76.44	76.70
Glass	64.42	63.44	63.51	64.87	70.00	63.44
Iris	97.33	96.00	96.00	96.67	98.67	96.00
Sonar	80.71	74.95	75.43	76.33	81.17	79.64

Table 4. Standard deviation of the normalized values of diversity measure

	<i>E</i>	<i>dis</i>	<i>DF</i>	<i>KW</i>	κ	<i>diff</i>
Breast W	0.154	0.352	0.012	0.165	0.119	0.045
Diabetes	0.247	0.100	0.054	0.142	0.157	0.099
Glass	0.277	0.198	0.27	0.257	0.034	0.182
Iris	0.006	0.017	0.264	0.306	0.109	0.000
Sonar	0.09	0.138	0.179	0.268	0.156	0.280
Average	0.155	0.161	0.156	0.228	0.115	0.121

(κ) seems to be reliable to specify the diversity of base classifiers. This observation, however, was based on only computational experiments for ensembles with five base classifiers. More computational experiments are needed in future work.

4 Conclusions

In this paper, we examined the use of various diversity measures in our EMO approach to the design of ensemble classifiers. Experimental results showed that the performance of designed ensemble classifiers depended on the choice of a diversity measure. Good results were obtained by the measurement of interrater agreement (κ).

Acknowledgements

This work was partially supported by Grant-in-Aid for Scientific Research on Young Scientists (B): KAKENHI (18700228).

References

1. Breiman, L.: Bagging Predictors. *Machine Learning* 24, 123–140 (1996)
2. 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)
3. Kuncheva, L.I., Whitaker, C.J.: Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning* 51, 181–207 (2003)
4. Tang, E.K., Suganthan, P.N., Yao, X.: An Analysis of Diversity Measures. *Machine Learning* 65, 247–271 (2006)
5. Abbass, H.A.: Pareto Neuro-evolution: Constructing Ensemble of Neural Networks using Multi-objective Optimization. In: *Proc. of 2003 IEEE Congress on Evolutionary Computation*, pp. 2074–2080 (2003)
6. Jin, Y., Okabe, T., Sendhoff, B.: Evolutionary Multi-objective Optimization Approach to Constructing Neural Network Ensembles for Regression. In: Coello, C.A.C., Lamont, G.B. (eds.) *Applications of Multi-Objective Evolutionary Algorithms*, pp. 653–673. World Scientific, Singapore (2004)
7. Chandra, A., Yao, X.: DIVACE: Diverse and Accurate Ensemble Learning Algorithm. In: Yang, Z.R., Yin, H., Everson, R.M. (eds.) *IDEAL 2004. LNCS*, vol. 3177, pp. 619–625. Springer, Heidelberg (2004)
8. Nojima, Y., Ishibuchi, H.: Designing Fuzzy Ensemble Classifiers by Evolutionary Multiobjective Optimization with an Entropy-based Diversity Criterion. In: *Proc. of 6th International Conference on Hybrid Intelligent Systems and 4th Conference on Neuro-Computing and Evolving Intelligence CD-ROM* (4 pages) (2006)
9. Nojima, Y., Ishibuchi, H.: Genetic Rule Selection with a Multi-Classifer Coding Scheme for Ensemble Classifier Design. *International Journal of Hybrid Intelligent Systems* 4(3), 157–169 (2007)
10. Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Selecting Fuzzy If-then Rules for Classification Problems using Genetic Algorithms. *IEEE Trans. on Fuzzy Systems* 3, 260–270 (1995)

11. Ishibuchi, H., Murata, T., Turksen, I.B.: Single-objective and Two-objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems. *Fuzzy Sets and Systems* 89, 135–150 (1997)
12. Ishibuchi, H., Nakashima, T., Murata, T.: Three-objective Genetics-based Machine Learning for Linguistic Rule Extraction. *Information Sciences* 136, 109–133 (2001)
13. Ishibuchi, H., Nakashima, T., Nii, M.: *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Springer, Berlin (2004)
14. Ishibuchi, H., Yamamoto, T.: Comparison of Heuristic Criteria for Fuzzy Rule Selection in Classification Problems. *Fuzzy Optimization and Decision Making* 3, 119–139 (2004)
15. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation* 6, 182–197 (2002)

Author Index

- Abraham, Ajith 281, 289
Alcalá, R. 747
Alcaraz, José M. 730
Alimi, Adel M. 533
Almeida, Leandro M. 156
Alonso, Teresa 425
Aracil, Rafael 649
Aranda, G. 507
Araujo, Carmen Paz Suárez 353
Arcay, Bernardino 196, 212
Arteaga, Jaime Muñoz 543
- Baca, José 649
Badii, Atta 132
Bajo, Javier 38
Banković, Zorana 132
Barbero, Álvaro 369
Baruque, Bruno 378
Bellas, Francisco 633
BenAbdelhafid, Abdellatif 533
Benderskaya, Elena N. 408
Bernadó-Mansilla, Ester 722
Bojanić, Slobodan 132
Bošanský, Branislav 86
Boticario, Jesús G. 329
Botti, Vicente 95, 507
Brom, Cyril 86
Brotóns, Rafael 188
- Caamaño, Pilar 633
Caballero-Gil, P. 475
Cano, Rosa 46
Carbó, Javier 62, 124
Caro, Andrés 425
Carrascosa, C. 507
Carricajo, Iciar 196
Casillas, Jorge 722
Castro H., Alberto 558
Ceravolo, Francesco 322
Chaves-González, José M. 257
Chira, Camelia 148, 273
Consalter, Luiz Airton 265
Corchado, Emilio 247, 378
Corchado, Juan M. 378, 688
- Cordón, Oscar 599
Couso, Inés 608
Cruz-Barbosa, Raúl 392
Cui, Zhiming 483
- d'Anjou, Alicia 665
Dafonte, Carlos 196, 212
Damas, Sergio 599
da Silva Maximiano, Marisa 257
De Felice, Matteo 322
de la Cal, Enrique 583
de la Puente, Paloma 441
Delgado, O. 475
del-Hoyo, Rafael 220
del Jesus, Maria Jose 140
de Lope, Javier 625, 641
del Olmo, Ricardo 567
De Paz, Juan F. 688
De Paz, Yanira 46
Dorronsoro, José R. 369
Ducange, P. 747
Dumitrescu, D. 148, 273
Durán, Maria Luisa 425
Duran, Orlando 265
Duro, Richard J. 633, 641
- Escalera, Juan A. 649
- Fang, Wei 483
Ferre, Manuel 649
Flórez-Revuelta, Francisco 451
Fraile, Juan A. 54
Fúster-Sabater, A. 475
- Gabrys, Bogdan 2
Galán, José M. 567
Galiana B., Gladis M. 558
Gao, Song 103
García Báez, Patricio 353
García, José Regidor 353
García, Salvador 4
García-Chamizo, Juan Manuel 451
García-Rodríguez, José 451
Gaya, Maria Cruz 78

- Georgieva, Antoniya 298
 Ghédira, Khaled 533
 Gil, Arturo 680
 Giraldez, J. Ignacio 78
 Gog, Anca 148
 Gómez-Pulido, Juan A. 257
 González B., Javier 558
 González B., Juan J. 551
 Graña, Manuel 641, 657, 665, 673
 Grosan, Crina 164
 Guan, Sheng-Uei 113
 Guijarro, María 345
 Guo, Dongwei 30
 Gutiérrez, Álvaro 616
- Hassan, Samer 523
 Hauptmann, Werner 466
 Heras, Stella 95, 515
 Hernandez, Angeles 575
 Herrera, Francisco 4, 747
 Herrero, Álvaro 247
 Hollmén, Jaakko 204
 Horzyk, Adrian 229
 Hsairi, Lobna 533
 Hu, Pengyu 483
 Hundertmark, Claudia 385
 Hurej, Arletta 400
- Ibáñez, Oscar 599
 Innocenti, Bianca 70
 Ishibuchi, Hisao 755
- Jackowski, Konrad 361
 Javier Herrera, P. 345
 Jiménez, Fernando 730
 Jordanov, Ivan 298
 Julián, Vicente 95, 515
 Jurado, Francisco 329
- Karras, D.A. 237
 Klawonn, Frank 385
 Kwolek, Bogdan 433
- Lacueva-Pérez, Francisco José 220
 Laitila, Erkki 499
 Lara, Ana Maria 188
 Layeb, Abdesslem 172
 Lazzerini, B. 747
 Li, Dan 30
 Lipinski, Piotr 180
- Liu, Hongbo 281
 Liu, Lei 417
 Liu, Yanbin 30
 López, Beatriz 70
 López, Jorge 369
 López, Raul 441
 López-Guede, José Manuel 673
 López-Paredes, Adolfo 567
 Ludermir, Teresa B. 156
- Manteiga, Minia 196, 212
 Marcelloni, F. 747
 Margain Fuentes, Ma. De Lourdes 543
 Markowska-Kaczmar, Urszula 400
 Márquez, Antonio 739
 Márquez, Francisco Alfredo 739
 Martínez F., José A. 551, 558
 Martín H., José Antonio 625
 Martín-del-Brio, Bonifacio 220
 Mata, Aitor 378
 Matía, Fernando 441, 616
 Medrano, Nicolás 220
 Mehboob, Zareen 714
 Mello, Paola 337
 Mendez, Gerardo M. 575
 Molina, José Manuel 62, 124
 Monasterio-Huelin, Félix 616
 Montana, Giovanni 591
 Morales-Rodríguez, María Lucila
 551, 558
 Moreno, María N. 458
 Moreno, Ramón 665
- Navarro, Iñaki 616
 Navarro, Marti 95
 Nieto, Octavio 132
 Nojima, Yusuke 755
 Nusser, Sebastian 466
- Ochoa Ortíz Zezzatti,
 Carlos Alberto 543
 Ordóñez, Diego 212
 Orriols-Puig, Albert 722
 Ortega, Manuel 329
 Otte, Clemens 466
- Pacheco, Joaquín 188
 Pajares, Gonzalo 345
 Palacios, Ana 608
 Pant, Millie 289
 Parras-Gutierrez, Elisabet 140

- Parrella, Francesco 591
 Pavard, Bernard 551
 Pavon, Juan 523
 Payá, Luis 680
 Pazos R., Rodolfo A. 558
 Peregrín, Antonio 739
 Pinho, Joel 458
 Pinteá, Camelia-M. 273
 Pinzón, Cristian 46
 Pizzuti, Stefano 322
 Pop, Petrica C. 273
 Prieto, Abraham 633
- Rani, Deepti 289
 Rebollo, Miguel 515
 Redondo, Miguel A. 329
 Reinoso, Oscar 680
 Reyes, César 425
 Rivas, Víctor M. 140
 Rodríguez, Alejandra 196
 Rodríguez, Fco. Álvarez 543
 Rodríguez, Nibaldo 265
 Rodríguez, Pablo G. 425
 Rodríguez, Sara 38
 Rodríguez-Losada, Diego 441
- Saidouni, Djamel-Eddine 172
 Sáiz, Lourdes 188
 Salgado, Mauricio 523
 Salvi, Joaquim 70
 Sánchez, Gracia 730
 Sánchez, José F. 730
 Sánchez, Luciano 608
 Sánchez, Miguel A. 54
 Sánchez-Pérez, Juan M. 257
 Sánchez-Pi, Nayat 62
 Santamaría, Jose 599
 Santos, Olga C. 329
 Sanz, Yolanda 625
 Sedano, Javier 583
 Segrera, Saddys 458
 Simić, Dragan 314
 Simić, Svetlana 314
- Simić-Ivkov, Milana 314
 Sirvio, Konsta 204
 Slankamenac, Petar 314
 Sogorb, Javier 680
 Sottara, Davide 337
 Srivastava, Dinesh Kumar 289
 Sun, Xiao-nan 417
- Tahayori, Hooman 491
 Tapia, Dante I. 54
 Thangaraj, Radha 289
 Tian, Xin 103
- Vaquerizo García, Maria Belén 306
 Vega-Rodríguez, Miguel A. 257
 Vellido, Alfredo 392
 Venturini, Verónica 124
 Vescan, Andreea 164
 Viadero, Carlos Fernández 353
 Villar José Ramón 583
 Villaverde, Ivan 657
 Visconti, Andrea 491
- Wang, Kangping 30
 Wozniak, Michal 361
 Wu, Xindong 1
- Xiao, Bo 103
- Yang, Zheng Rong 696, 706
 Ye, Hong 281
 Yerpes, Ariadna 649
 Yin, Hujun 15, 714
 Yu, Danpin 103
- Zeng, Sanyou 103
 Zhang, Hong 417
 Zhang, Lei 103
 Zhao, Lei 417
 Zhao, Pengpeng 483
 Zhou, Chunguang 30
 Zhu, Fangming 113
 Zhukova, Sofya V. 408
 Zulueta, Ekaitz 673