

LNAI 6678

Emilio Corchado
Marek Kurzyński
Michał Woźniak (Eds.)

Hybrid Artificial Intelligent Systems

6th International Conference, HAIS 2011
Wroclaw, Poland, May 2011
Proceedings, Part I

1
Part I



Springer

Lecture Notes in Artificial Intelligence 6678
Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Emilio Corchado Marek Kurzyński
Michał Woźniak (Eds.)

Hybrid Artificial Intelligent Systems

6th International Conference, HAIS 2011
Wroclaw, Poland, May 23-25, 2011
Proceedings, Part I

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

University of Salamanca

Faculty of Biology

BISITE Research Group

Salamanca, Spain

E-mail: escorhado@usal.es

Marek Kurzyński

Michał Woźniak

Wrocław University of Technology

Faculty of Electronics

Department of Systems and Computer Networks

Wrocław, Poland

E-mail: {marek.kurzynski,michal.wozniak}@pwr.wroc.pl

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-21218-5

e-ISBN 978-3-642-21219-2

DOI 10.1007/978-3-642-21219-2

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2011

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

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

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

Printed on acid-free paper

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

Preface

Hybrid intelligent systems are becoming more and more popular due to their capabilities in handling many real-world complex problems, involving imprecision, uncertainty, vagueness and high dimensionality. Intelligent systems are pervasive in our society and growing in size and complexity at an astonishing pace, and the response to the coming challenges will come; as always it has the modularity and the ability to decompose problems and find the best partial solution. In this paradigm, hybrid intelligent systems are the natural approach to the problems, rather than the exceptional case. They provide us with the opportunity to use both our knowledge and raw data to solve problems in a more interesting and promising way. This multidisciplinary research field is in a continuous expansion within the artificial intelligence research community.

The 6th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2011) provided an interesting opportunity to present and discuss the latest theoretical advances and real-world applications in this multidisciplinary research field.

The volume of *Lecture Notes in Artificial Intelligence* (LNAI) includes accepted papers presented at HAIS 2011, which took place in Wroclaw University of Technology, Wroclaw, Poland, in May 2011.

Since the first edition held 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 agent and multi-agent systems, bioinformatics and bio-inspired models, fuzzy systems, artificial vision, artificial neural models, optimization models, compound and combined classification and alike.

HAIS 2011 received 241 technical submissions. After a rigorous peer-review process the International Program Committee selected 114 papers, which are published in these conference proceedings. In this edition emphasis was put to the organization of special sessions. Nine special sessions were organized on the following topics:

- Hybrid Intelligence System on Logistics and Intelligent Optimization
- Metaheuristics for Combinatorial Optimization and Modelling Complex Systems
- Hybrid Systems for Context-Based Information Fusion
- Methods of Classifier Fusion
- Intelligent Systems for Data Mining and Applications
- Systems, Man, and Cybernetics
- Hybrid Artificial Intelligence Systems in Management of Production Systems
- Hybrid Artificial Intelligent Systems for Medical Applications
- Hybrid Intelligent Approaches in Cooperative Multi-robot Systems

The editors would like to express their deep thanks to authors for their valuable submissions and all reviewers and special session organizers for their hard work. A thorough peer-review process is very important to maintain the high quality of a conference and the HAIS series of conferences would not exist without their help.

As a follow-up of the conference, we anticipate further publication of selected papers in special issues of the following journals:

- *Computational Intelligence*, Wiley-Blackwell
- *Expert Systems*, Wiley-Blackwell
- *Neurocomputing*, Elsevier
- *Journal of Applied Logic*, Elsevier

HAIS 2011 enjoyed outstanding keynote speeches by distinguished guest speakers:

- Ajith Abraham - Machine Intelligence Research Labs (USA)
- Francisco Herrera - University of Granada (Spain)
- Adam Krzyżak - Concordia University (Canada)
- Juliusz Lech Kulikowski - M. Nalecz Institute of Biocybernetics and Biomedical Engineering PAS (Poland)
- James Llinas - State University of New York at Buffalo (USA)
- B. John Oommen - Carleton University (Canada)
- Gerald Schaefer - Loughborough University (UK)

We would like to fully acknowledge support from the Wrocław University of Technology, especially from the Dean of the Faculty of Electronics and the Chairs of the Department of Systems and Computer Networks. The IEEE Systems, Man & Cybernetics Society, through its Spanish and Czech Republic chapters, the Spanish Association for Artificial Intelligence (AEPIA), MIR LABS and the International Federation for Computational Logic have also supported this event.

We would like to thank Alfred Hofmann and Anna Kramer from Springer for their help and collaboration during the publication process.

Last but not least we would like to give special thanks to the local organizing team (Robert Burduk, Kondrad Jackowski, Bartosz Krawczyk, Maciej Krysmann, Bartosz Kurlej, Piotr Sobolewski, Szymon Sztajer, Marcin Zmysłony, Andrzej Żołnierek) who did a great job.

May 2011

Emilio Corchado
Marek Kurzyński
Michał Woźniak

Organization

Honorary Chair

Tadeusz Więckowski	Rector, Wroclaw University of Technology (Poland)
Andrzej Kasprzak	Vice-Rector for Education, Wroclaw University of Technology (Poland)
Jan Zarzycki	Dean of Faculty of Electronics, Wroclaw University of Technology (Poland)
Marek Kurzyński	Wroclaw University of Technology (Poland)

General Chair

Emilio Corchado	University of Salamanca (Spain)
-----------------	---------------------------------

International Advisory Committee

Ajith Abraham	Machine Intelligence Research Labs (Europe)
Antonio Bahamonde	Spanish Association for Artificial Intelligence (AEPIA)
Carolina Blasco	Regional Government of Castilla y León (Spain)
Pedro M. Caballero	CARTIF (Spain)
Andre de Carvalho	University of São Paulo (Brazil)
Sung-Bae Cho	Yonsei University (Korea)
Juan M. Corchado	University of Salamanca (Spain)
José R. Dorronsoro	Autonomous University of Madrid (Spain)
Michael Gabbay	Kings College London (UK)
Ali A. Ghorbani	UNB (Canada)
Mark A. Girolami	University of Glasgow (Scotland)
Manuel Graña	University of the Basque Country (Spain)
Petro Gopych	Universal Power Systems USA-Ukraine LLC (Ukraine)
Jon G. Hall	The Open University (UK)
Francisco Herrera	University of Granada (Spain)
César Hervás-Martínez	University of Córdoba (Spain)
Tom Heskes	Radboud University Nijmegen (Holland)
Dusan Husek	Academy of Sciences of the Czech Republic (Czech Republic)
Lakhmi Jain	University of South Australia (Australia)
Samuel Kaski	Helsinki University of Technology (Finland)
Daniel A. Keim	University of Konstanz (Germany)

Isidro Laso	D.G. Information Society and Media (European Commission)
Amy Neustein	Linguistic Technology Systems (USA)
Marios Polycarpou	University of Cyprus (Cyprus)
Witold Pedrycz	University of Alberta (Canada)
Václav Snášel	VSB-Technical University of Ostrava (Czech Republic)
Xin Yao	University of Birmingham (UK)
Hujun Yin	University of Manchester (UK)
Michał Woźniak	Wrocław University of Technology (Poland)

Program Committee

Emilio Corchado	University of Salamanca (Spain) (PC Co-chair)
Michał Woźniak	Wrocław University of Technology (Poland) (PC Co-chair)
Agnar Aamodt	Norwegian University of Science and Technology (Norway)
Jesús Alcalá-Fernández	University of Granada (Spain)
Rafael Alcalá	University of Granada (Spain)
José Luis Álvarez	University of Huelva (Spain)
Davide Anguita	University of Genoa (Italy)
Bruno Apolloni	Università degli Studi di Milano (Italy)
Antonio Aráuzo-Azofra	University of Córdoba (Spain)
Estefanía Argente	University of Valencia (Spain)
Fidel Aznar	University of Alicante (Spain)
Jaume Bacardit	University of Nottingham (UK)
Antonio Bahamonde	University of Oviedo (Spain)
Javier Bajo	Universidad Pontificia de Salamanca (Spain)
John Beasley	Brunel University (UK)
Bruno Baroque	University of Burgos (Spain)
Joé Manuel Benítez	University of Granada (Spain)
Ester Bernadó	Universitat Ramon Llull (Spain)
Richard Blake	Norwegian University of Science and Technology (Norway)
Juan Botía	University of Murcia (Spain)
Vicente Botti	Universidad Politécnica de Valencia (Spain)
Robert Burduk	Wrocław University of Technology (Poland)
Gustavo Olague Caballero	CICESE (Mexico)
Jose Luis Calvo	University of Coruña (Spain)
José Ramón Cano	University of Jaén (Spain)
Davide Carneiro	Universidade do Minho (Portugal)
Cristóbal José Carmona	University of Jaén (Spain)
Leocadio González Casado	Universidad de Almería (Spain)
Blanca Cases	University of the Basque Country (Spain)
Oscar Castillo	Tijuana Institute of Technology (Mexico)

Paula María Castro Castro	Universidade da Coruña (Spain)
Jonathan Chan	King Mongkut's University of Technology Thonburi (Thailand)
Richard Chbeir	Bourgogne University (France)
Enhong Chen	University of Science and Technology of China (China)
Camelia Chira	University of Babes-Bolyai (Romania)
María Guijarro	University of Madrid (Spain)
Sung-Bae Cho	Yonsei University (Korea)
Darya Chyzyk	University of the Basque Country (Spain)
Juan Manuel Corchado	University of Salamanca (Spain)
Emilio Corchado	University of Salamanca (Spain)
Rafael Corchuelo	University of Seville (Spain)
Guionmar Corral	University Ramon Llull (Spain)
Raquel Cortina Parajon	University of Oviedo (Spain)
Ângelo Costa	Universidade do Minho (Portugal)
Carlos Cotta	University of Málaga (Spain)
José Alfredo F. Costa	Universidade Federal do Rio Grande do Norte (Brazil)
Juan José del Coz Velasco	Universidad de Oviedo (Spain)
Francisco Cuevas	CIO (Mexico)
Leticia Curiel	University of Burgos (Spain)
Alfredo Cuzzocrea	University of Calabria (Italy)
Keshav Dahal	University of Bradford (UK)
Theodoros Damoulas	Cornell University (UK)
Ernesto Damiani	University of Milan (Italy)
Bernard De Baets	Ghent University (Belgium)
Enrique de la Cal	University of Oviedo (Spain)
Javier de Lope Asiain	Universidad Politécnica de Madrid (Spain)
Marcilio de Souto	Universidade Federal do Rio Grande do Norte (Brazil)
María José del Jesús	University of Jaén (Spain)
Ricardo del Olmo	University of Burgos (Spain)
Joaquín Derrac	University of Granada (Spain)
Nicola Di Mauro	University of Bari (Italy)
António Dourado	University of Coimbra (Portugal)
Richard Duro	University of Coruña (Spain)
Susana Irene Díaz Rodríguez	University of Oviedo (Spain)
José Dorronsoro	Universidad Autónoma de Madrid (Spain)
Pietro Ducange	University of Pisa (Italy)
Talbi El-Ghazali	University of Lille (France)
Aboul Ella Hassanien	University of Cairo (Egypt)
Marc Esteva	Artificial Intelligence Research Institute (Spain)
Juan José Flores	University of Michoacana (Mexico)
Alberto Fernández	Universidad Rey Juan Carlos (Spain)

Alberto Fernández	University of Granada (Spain)
Elías Fernández-Combarro	
Álvarez	University of Oviedo (Spain)
Elsa Fernández	University of the Basque Country (Spain)
Nuno Ferreira	Instituto Politécnico de Coimbra (Portugal)
Juan Arturo Nolazco Flores	ITESM (Mexico)
Richard Freeman	Capgemini (Spain)
Rubén Fuentes	Universidad Complutense de Madrid (Spain)
Giorgio Fumera	University of Cagliari (Italy)
Bogdan Gabrys	Bournemouth University (UK)
João Gama	University of Porto (Portugal)
Matjaz Gams	Jozef Stefan Institute Ljubljana (Slovenia)
Rosario Girardi	Federal University of Maranhão (Brazil)
Jun Gao	Hefei University of Technology (China)
Tom Heskes	Radboud University Nijmegen (The Netherlands)
Isaías García	University of León (Spain)
José García	University of Alicante (Spain)
Salvador García	University of Jaén (Spain)
Neveen Ghali	Azhar University (Egypt)
Adriana Giret	Universidad Politécnica de Valencia (Spain)
Jorge Gómez Sanz	Universidad Complutense de Madrid (Spain)
Pedro González García	University of Jaén (Spain)
Petro Gopych	Universal Power Systems USA-Ukraine LLC (Ukraine)
Jose Luis Gordillo	ITESM (Mexico)
Juan Manuel Górriz	University of Granada (Spain)
Maite García-Sebastián	University of the Basque Country (Spain)
Manuel Graña	University of the Basque Country (Spain)
Maciej Grzenda	Warsaw University of Technology (Poland)
Arkadiusz Grzybowski	Wrocław University of Technology (Poland)
Jerzy Grzymala-Busse	University of Kansas (USA)
Anne Håkansson	Stockholm University (Sweden)
Saman Halgamuge	The University of Melbourne (Australia)
José Alberto Hernández	Universidad Autónoma del Estado de Morelos (Mexico)
Carmen Hernández	University of the Basque Country (Spain)
Francisco Herrera	University of Granada (Spain)
Álvaro Herrero	University of Burgos (Spain)
Sean Holden	University of Cambridge (UK)
Vasant Honavar	Iowa State University (USA)
Konrad Jackowski	Wrocław University of Technology (Poland)
Yaochu Jin	Honda Research Institute Europe (Germany)
Ulf Johansson	University of Borås (Sweden)
Ivan Jordanov	University of Portsmouth (UK)
Cliff Joslyn	Pacific Northwest National Laboratory (USA)

Vicente Julián Inglada	Universidad Politécnica de Valencia (Spain)
Juha Karhunen	Helsinki University of Technology (Finland)
Przemysław Kazienko	Wroclaw University of Technology (Poland)
Frank Klawonn	University of Applied Sciences
Andreas König	Braunschweig/Wolfenbüttel (Germany)
Mario Köppen	University of Kaiserslautern (Germany)
Rudolf Kruse	Kyushu Institute of Technology (Japan)
Bernadetta Kwintiana	Otto-von-Guericke-Universität Magdeburg
Dario Landa-Silva	(Germany)
Soo-Young Lee	Universität Stuttgart (Germany)
Lenka Lhotská	University of Nottingham (UK)
Hailin Liu	Brain Science Research Center (Korea)
Otoniel López Granado	Czech Technical University in Prague
Karimele López de Ipiña	(Czech Republic)
Oscar Luaces	Guangdong University of Technology (China)
Teresa Ludermir	Universidad Autónoma de Madrid (Spain)
Julián Luengo	University of the Basque Country (Spain)
Wenjian Luo	Universidad de Oviedo (Spain)
Núria Macià	Universidade Federal de Pernambuco (Brazil)
Kurosh Madani	University of Granada (Spain)
Ana Maria Madureira	University of Science and Technology of China
Roque Marin	(China)
Yannis Marinakis	Universitat Ramon Llull (Spain)
Jose Luis Marroquin	University of Paris-Est Creteil (France)
José Fco. Martínez-Trinidad	Instituto Politécnico do Porto (Portugal)
José Luis Martínez	University of Murcia (Spain)
Jacinto Mata	Technical University of Crete (Greece)
Giancarlo Mauri	CIMAT (Mexico)
Dawn Medlin	INAOE (Mexico)
David Meehan	University of Castilla La Mancha (Spain)
Gerardo M. Méndez	University of Huelva (Spain)
Abdel-Badeeh M. Salem	University of Milano-Bicocca (Italy)
Masoud Mohammadian	Appalachian State University (USA)
José Manuel Molina	Dublin Institute of Technology (Ireland)
Claudio Moraga	Instituto Tecnológico de Nuevo León (Mexico)
Marco Mora	Ain Shams University (Egypt)
Eduardo Morales	University of Canberra (Australia)
Ramón Moreno	University Carlos III of Madrid (Spain)
Juan Álvaro Muñoz Naranjo	European Centre for Soft Computing (Spain)
Dimitris Mountzis	Universidad Católica del Maule (Chile)
Susana Nascimento	INAOE (Mexico)
	Universidad del País Vasco (Spain)
	Universidad de Almería (Spain)
	University of Patras (Greece)
	Universidade Nova de Lisboa (Portugal)

Martí Navarro Llácer	Universidad Politécnica de Valencia (Spain)
Amy Neustein	Linguistic Technology Systems (USA)
Yusuke Nojima	Osaka Prefecture University (Japan)
Paulo Novais	Universidade do Minho (Portugal)
Alberto Ochoa-Zezzatti	Juarez City University/CIATEC (Mexico)
Albert Orriols	University Ramon Llull (Spain)
Rubé Ortiz	Universidad Rey Juan Carlos (Spain)
Vasile Palade	Oxford University (UK)
Stephan Pareigis	Hamburg University of Applied Sciences (Germany)
Witold Pedrycz	University of Alberta (Canada)
Elzbieta Pekalska	University of Manchester (UK)
Jorge Díez Peláez	Universidad de Oviedo (Spain)
Carlos Pereira	Universidade de Coimbra (Portugal)
Antonio Peregrín	University of Huelva (Spain)
Lina Petrakieva	Glasgow Caledonian University (UK)
Gloria Phillips-Wren	Loyola College in Maryland (USA)
Han Pingchou	Peking University (China)
Camelia Pintea	University of Babes-Bolyai (Romania)
Igor Podolak	Jagiellonian University (Poland)
Julio Ponce	Universidad Autónoma de Aguascalientes (Mexico)
Piotr Porwik	University of Silesia (Poland)
Khaled Ragab	King Faisal University (Saudi Arabia)
José Ranilla Pastor	University of Oviedo (Spain)
Javier Ramírez Pé	University of Granada (Spain)
Romain Raveaux	La Rochelle University (France)
Carlos Redondo Gil	University of León (Spain)
Bernadete Ribeiro	University of Coimbra (Portugal)
Mariano Rivera	CIMAT (Mexico)
Ramón Rizo	University of Alicante (Spain)
Peter Rockett	The University of Sheffield (UK)
Adolfo Rodríguez	University of León (Spain)
Rosa M. Rodríguez Maraña	University of León (Spain)
Katya Rodriguez-Vázquez	Universidad Nacional Autónoma de México (Mexico)
Adam Roman	Jagiellonian University (Poland)
Fabrice Rossi	TELECOM ParisTech (France)
António Ruano	University of Algarve (Portugal)
Ashraf Saad	Armstrong Atlantic State University (USA)
Khalid Saeed	AGH University of Science and Technology (Poland)
Ozgur Koray Sahingoz	Turkish Air Force Academy (Turkey)
Wei-Chiang Samuelson Hong	Oriental Institute of Technology (Taiwan)
Luciano Sánchez	University of Oviedo (Spain)

José Santamaría	University of Jaén (Spain)
Alexandre Savio	University of the Basque Country (Spain)
Fatima Sayuri Quezada	Universidad Autónoma de Aguascalientes (Mexico)
Gerald Schaefer	Aston University (UK)
Robert Schaefer	AGH University of Science and Technology (Poland)
Javier Sedano	University of Burgos (Spain)
Leila Shafti	Universidad Autónoma de Madrid (Spain)
Dragan Simic	Novi Sad Fair (Serbia)
Konstantinos Sirlantzis	University of Kent (UK)
Dominik Slezak	University of Regina (Canada)
Humberto Sossa	CIC-IPN (Mexico)
Cecilia Sönströd	University of Borås (Sweden)
Luis Enrique Sucar Succar	INAOE (Mexico)
Ying Tan	Peking University (China)
Ke Tang	University of Science and Technology of China (China)
Bogdan Trawiński	Wroclaw University of Technology (Poland)
Nikos Thomaidis	University of the Aegean (Greece)
Alicia Troncoso	Universidad Pablo de Olavide de Sevilla (Spain)
Eiji Uchino	Yamaguchi University (Japan)
Roberto Uribeetxeberria	Mondragon University (Spain)
Stefan Valcuha	Slovak University of Technology in Bratislava (Slovakia)
José Valls	University Carlos III of Madrid (Spain)
Miguel Ángel Veganzones	Universidad del País Vasco (Spain)
Sebastian Ventura	Universidad de Córdoba (Spain)
José Luis Verdegay	University of Granada (Spain)
Ivica Veza	University of Split (Croatia)
José Ramón Villar	University of Oviedo (Spain)
José Ramón Cano de Amo	University of Jaén (Spain)
Krzysztof Walkowiak	Wroclaw University of Technology (Poland)
Guoyin Wang	Chongqing University of Posts and Telecommunications (China)
Michał Wozniak	Wroclaw University of Technology (Poland)
Zhuoming Xu	Hohai University (China)
Ronald Yager	Iona College (USA)
Hujun Yin	The University of Manchester (UK)
Indre Zliobaite Constantin	Technical University of Crete (Greece)
Zopounidis	Brunel University (UK)
Huiyu Zhou	University of Genoa (Italy)
Rodolfo Zunino	Mondragon University (Spain)
Urko Zurutuza	

Special Session Committees

Hybrid Intelligent Systems on Logistics and Intelligent Optimization

Camelia Chira	Babes-Bolyai University (Romania)
Alberto Ochoa	Universidad Autónoma de Ciudad Juárez (Mexico)
Miguel Vargas	UOIT (Canada)
Felipe Padilla	ETS, Québec (Canada)
Julio Ponce	Aguascalientes University (Mexico)
Ricardo Aceves	UNAM (Mexico)
Luciana Buriol	UFRGS (Brazil)
Cristiano Castelfranchi	ISTC-CNR (Italy)
Cerasela Crisan	University of Bacau (Romania)
D. Dumitrescu	Babes-Bolyai University (Romania)
Anca Gog	Babes-Bolyai University (Romania)
Luana Hatsukimi	UNICAMP (Brazil)
Stella Heras	Universidad Politécnica de Valencia (Spain)
Samer Hassan	Surrey University (UK)
Barna Iantovics	Petru Maior University Targu-Mures (Romania)
Fabricio Olivetti de França	University of Campinas (Brazil)
Camelia-M. Pintea	Babes-Bolyai University (Romania)
Petrica Pop	North University Baia-Mare (Romania)
Dario Landa-Silva	University of Nottingham (UK)
Simoneé Suarent	Technical University (Mauritius)
José Ramón Villar	University of Oviedo (Spain)

Metaheuristics for Combinatorial Optimization and Modelling Complex Systems

Camelia Chira	University of Babes-Bolyai (Romania)
Enrique de la Cal	University of Oviedo (Spain)
José Ramón Villar	University of Oviedo (Spain)
Alba Berzosa	Instituto Tecnológico de Castilla y León (Spain)
André Carvalho	University of Sao Paulo (USP) at San Carlos (Brazil)
Camelia Chira	University of Babes-Bolyai (Romania)
Enrique de la Cal	University of Oviedo (Spain)
Anca Gog	Babes-Bolyai University (Romania)
Nima Hatami	University of Cagliari (Italy)
Dragos Horvath	Université de Strasbourg (France)
Eduardo Raul Hruschka	University of Sao Paulo (USP) at San Carlos (Brazil)
Oscar Ibañez	European Centre for Soft Computing (Spain)
David Iclanzan	Sapientia University (Romania)

Rodica Ioana Lung	Babes-Bolyai University (Romania)
Paula Mello	University of Bologna (Italy)
Gerardo M. Méndez	Instituto Tecnológico de Nuevo León, Mexico
Luis Oliveira	University of Oviedo (Spain)
Ana Palacios	University of Oviedo (Spain)
Camelia Pintea	University of Babes-Bolyai (Romania)
Adolfo Rodríguez	University of León (Spain)
Luciano Sánchez	University of Oviedo (Spain)
María del Rosario Suárez	University of Oviedo (Spain)
Javier Sedano	University of Burgos (Spain)
Carmen Vidaurre	Technical University of Berlin (Germany)
José Ramón Villar	University of Oviedo (Spain)
Jose Luis Calvo Rolle	University of A Coruña (Spain)
María Sierra	University of Oviedo (Spain)

Methods of Classifiers Fusion

Robert Burduk	Wroclaw University of Technology (Poland)
Emilio Corchado	University of Burgos (Spain)
José Alfredo Costa	Universidade Federal do Rio Grande do Norte (Brazil)
Giorgio Fumera	University of Cagliari (Italy)
Bogdan Gabrys	Bournemouth University (UK)
Álvaro Herrero	Universidad de Burgos (Spain)
Konrad Jackowski	Wroclaw University of Technology (Poland)
Przemyslaw Kazienko	Wroclaw University of Technology (Poland)
Elzbieta Pekalska	University of Manchester (UK)
Konstantinos Sirlantzis	University of Kent (UK)
Václav Snášel	VSB-Technical University of Ostrava (Czech Republic)
Jerzy Stefanowski	Poznan University of Technology (Poland)
Igor T. Podolak	Jagiellonian University (Poland)
Bogdan Trawinski	Wroclaw University of Technology (Poland)
Michał Wozniak	Wroclaw University of Technology (Poland)

Intelligent Systems for Data Mining and Applications

Alicia Troncoso Lora	Pablo de Olavide University of Seville (Spain)
Francisco Martínez Álvarez	Pablo de Olavide University of Seville (Spain)
Jesús S. Aguilar Ruiz	Pablo de Olavide University of Seville (Spain)
Marta Arias	Polytechnic University of Catalonia (Spain)
Manuel D. de la Iglesia	New York University (USA)
Héctor Pomares Cintas	University of Granada (Spain)
José C. Riquelme Santos	University of Seville (Spain)
Cristina Rubio Escudero	University of Seville (Spain)
Esteban Tabak	New York University (USA)

Systems, Man, Cybernetics by HAIS Workshop

Emilio Corchado	University of Salamanca (Spain)
Manuel Grana	University of the Basque Country (Spain)
Richard Duro	University of Coruna (Spain)
Juan M. Corchado	University of Salamanca (Spain)
Vicent Botti	Polytechnical University of Valencia (Spain)
Ramon Rizo	University of Alicante (Spain)
Juan Pavon	University Complutense of Madrid (Spain)
Jose Manuel Molina	University Carlos III of Madrid (Spain)
Francisco Herrera	University of Granada (Spain)
Cesar Hervas	University of Cordoba (Spain)
Sebastian Ventura	University of Cordoba (Spain)
Alvaro Herrero	University of Burgos (Spain)
Bruno Baroque	University of Burgos (Spain)
Javier Sedano	University of Burgos (Spain)
Sara Rodriguez	University of Salamanca (Spain)

Hybrid Artificial intelligence Systems in Management of Production Systems

Edward Chlebus	Wroclaw University of Technology (Poland)
Bożena Skolud	Silesian University of Technology (Poland)
Anna Burduk	Wroclaw University of Technology (Poland)
Jarosław Chrobot	Wroclaw University of Technology (Poland)
Stefan Valcuha	Slovak University of Technology in Bratislava (Slovakia)
Kamil Krot	Wroclaw University of Technology (Poland)
Michał Kuliberda	Wroclaw University of Technology (Poland)
Dimitris Mourtzis	University of Patras (Greece)
Mieczysław Jagodziński	Silesian University of Technology (Poland)
Jacek Czajka	Wroclaw University of Technology (Poland)
Arkadiusz Górska	Wroclaw University of Technology (Poland)
Arkadiusz Kowalski	Wroclaw University of Technology (Poland)
Ivica Veza	University of Split (Croatia)
Krzysztof Kalinowski	Silesian University of Technology (Poland)

Hybrid Intelligent Approaches in Cooperative Multi-robot Systems

Manuel Graña	Universidad del País Vasco (Spain)
Richard J. Duro	Universidade da Coruña (Spain)
Javier de Lope	Universidad Politécnica de Madrid (Spain)

Hybrid Artificial Intelligent Systems for Medical Application

Vicente Vera
Emilio Corchado

Universidad Complutense de Madrid (Spain)
University of Salamanca (Spain)

Organizing Committee

Robert Burduk (Chair)

Kondrad Jackowski

Bartosz Krawczyk

Maciej Krysmann

Bartosz Kurlej

Piotr Sobolewski

Szymon Sztajer

Katarzyna Tylkowska

Marcin Zmysłony

Andrzej Żołnierek

Table of Contents – Part I

Plenary

Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution	1
<i>A. Fernández, S. García, and F. Herrera</i>	
A New Tool for the Modeling of AI and Machine Learning Applications: Random Walk-Jump Processes.....	11
<i>Anis Yazidi, Ole-Christoffer Granmo, and B. John Oommen</i>	
Pattern Recognition Based on Similarity in Linear Semi-ordered Spaces	22
<i>Juliusz L. Kulikowski and Małgorzata Przytulska</i>	
Quo Vadis Hybrid Intelligent Systems: New Trends and Approaches (Abstract)	30
<i>Ajith Abraham</i>	
Reflections on Concepts of Employment for Modern Information Fusion and Artificial Intelligence Technologies: Situation Management, Decision Making under Varying Uncertainty and Ambiguity, Sequential Decision-Making, Learning, Prediction, and Trust (Abstract)	31
<i>James Llinas</i>	

General Track

A Genetic Algorithm Applied to a Main Sequence Stellar Model	32
<i>Gabriela de Oliveira Penna Tavares and Marco Aurelio Cavalcanti Pacheco</i>	
Using Artificial Intelligence Techniques for Strategy Generation in the <i>Commons Game</i>	43
<i>Petro Verkhogliad and B. John Oommen</i>	
Evaluation of Network Survivability Considering Degree of Separation	51
<i>Frank Yeong-Sung Lin, Hong-Hsu Yen, Pei-Yu Chen, and Ya-Fang Wen</i>	
Fuzzy Control of Trade-off between Exploration and Exploitation Properties of Evolutionary Algorithms	59
<i>Adam Slowik</i>	

Hybridization of Evolutionary Algorithm with Yule Walker Method to Design Minimal Phase Digital Filters with Arbitrary Amplitude Characteristics	67
<i>Adam Slowik</i>	
Automatic Identification Approach for Sea Surface Bubbles Detection	75
<i>Juan José Fuertes, Carlos M. Travieso, and J.B. Alonso</i>	
The Application of Artificial Intelligence Hybrid in Traffic Flow	83
<i>Ilija Tanackov, Vuk Bogdanović, Jovan Tepić, Siniša Sremac, and Nenad Ruškić</i>	
Diagnosis of Partial Discharge Using Self Organizing Maps and Hierarchical Clustering – An approach	91
<i>Rubén Jaramillo-Vacio, Alberto Ochoa-Zezzatti, S. Jöns, Sergio Ledezma-Orozco, and Camelia Chira</i>	
Bayesian Segmentation of Magnetic Resonance Images Using the α -Stable Distribution	99
<i>Diego Salas-Gonzalez, Matthias Schlögl, Juan M. Górriz, Javier Ramírez, and Elmar Lang</i>	
On-Line Valuation of Residential Premises with Evolving Fuzzy Models	107
<i>Edwin Lughofner, Bogdan Trawiński, Krzysztof Trawiński, and Tadeusz Lasota</i>	
Investigation of Genetic Algorithms with Self-adaptive Crossover, Mutation, and Selection	116
<i>Magdalena Smętek and Bogdan Trawiński</i>	
Hybrid Multi-agent System for Knowledge Management in Distributed Control System	124
<i>Dariusz Choinski, Mieczysław Metzger, and Witold Nocon</i>	
SVM with Bounds of Confidence and PLS for Quantifying the Effects of Acupuncture on Migraine Patients	132
<i>M. López, J.M. Górriz, J. Ramírez, D. Salas-Gonzalez, R. Chaves, and M. Gómez-Río</i>	
An Intelligent Automated Recognition System of Abnormal Structures in WCE Images	140
<i>Piotr Szczępiński, Artur Klepaczko, and Michał Strzelecki</i>	
Effective Diagnosis of Alzheimer's Disease by Means of Distance Metric Learning	148
<i>R. Chaves, J. Ramírez, J.M. Górriz, D. Salas-Gonzalez, M. López, I. Illán, F. Segovia, and A. Olivares</i>	

Risk Estimation for Hierarchical Classifier	156
<i>I.T. Podolak and A. Roman</i>	
Combining Meta-learning and Active Selection of Datasetoids for Algorithm Selection	164
<i>Ricardo B.C. Prudêncio, Carlos Soares, and Teresa B. Ludermir</i>	
A Parallel Genetic Programming Algorithm for Classification	172
<i>Alberto Cano, Amelia Zafra, and Sebastián Ventura</i>	
Evolutionary Algorithm for P2P Multicasting Network Design Problem	182
<i>Michał Wiśniewski and Krzysztof Walkowiak</i>	
A Focused Wave Front Algorithm for Mobile Robot Path Planning	190
<i>Anshika Pal, Ritu Tiwari, and Anupam Shukla</i>	
Evolving Temporal Fuzzy Association Rules from Quantitative Data with a Multi-Objective Evolutionary Algorithm	198
<i>Stephen G. Matthews, Mario A. Gongora, and Adrian A. Hopgood</i>	
Stereovision-Based Obstacle Avoidance Procedure for Autonomous Mobile Platforms	206
<i>Maciej Polańczyk, Agnieszka Owczarek, Michał Strzelecki, and Krzysztof Ślot</i>	
Detecting Unknown Attacks in Wireless Sensor Networks Using Clustering Techniques	214
<i>Z. Banković, J.M. Moya, J.C. Vallejo, and D. Fraga</i>	
A Hybrid System with Regression Trees in Steel-Making Process	222
<i>Miroslaw Kordos, Marcin Blachnik, Marcin Perzyk, Jacek Kozłowski, Orestes Bystrzycki, Mateusz Gródek, Adrian Byrdziak, and Zenon Motyka</i>	
Interval Type-2 Fuzzy Modelling and Simulated Annealing for Real-World Inventory Management	231
<i>Simon Miller, Mario Gongora, and Robert John</i>	
An Evidential Fusion Architecture for People Surveillance in Wide Open Areas	239
<i>M. Fornaciari, D. Sottara, A. Prati, P. Mello, and R. Cucchiara</i>	
Artificial Neural Networks Application in Software Testing Selection Method	247
<i>Kristina Smilgyte and Jovita Nenortaitė</i>	
Combining OWL Ontology and Schema Annotations in Metadata Management	255
<i>Tadeusz Pankowski</i>	
Benchmarking IBHM Method Using NN3 Competition Dataset	263
<i>Paweł Zawistowski and Jarosław Arabas</i>	

Face Tracking Using Adaptive Appearance Models and Convolutional Neural Network	271
<i>Boguslaw Rymut and Bogdan Kwolek</i>	
Genetic Selection of Subgraphs for Automatic Reasoning in Design Systems	280
<i>Barbara Strug</i>	
Controlling the Prediction Accuracy by Adjusting the Abstraction Levels	288
<i>Tomasz Łukaszewski, Joanna Józefowska, Agnieszka Lawrynowicz, Lukasz Józefowski, and Andrzej Lisiecki</i>	
Delta Analysis: A Hybrid Quantitative Approach for Measuring Discrepancies between Business Process Models	296
<i>Eren Esgin and Pinar Senkul</i>	
Analysis of Face Gestures for Human-Computer Interaction	305
<i>Jacek Rondio and Aleksandra Królik</i>	
Assessing Safety of Object Pushing Using the Principle of Reversibility	313
<i>Yuri Gavshin and Maarja Kruusmaa</i>	
Class Prediction in Microarray Studies Based on Activation of Pathways	321
<i>Henryk Maciejewski</i>	
A Hybrid Approach for ECG Classification Based on Particle Swarm Optimization and Support Vector Machine	329
<i>Dawid Kopiec and Jerzy Martyna</i>	
Fuzzy Modeling of Digital Products Pricing in the Virtual Marketplace	338
<i>Jarosław Jankowski, Jarosław Wątroboski, and Mateusz Piwowarski</i>	
An Algorithm Based on Genetic Fuzzy Systems for the Selection of Routes in Multi-Sink Wireless Sensor Networks	347
<i>Líliam B. Leal, Marcus Vinícius de S. Lemos, Raimir Holanda Filho, Ricardo A.L. Rabelo, and Fabio A.S. Borges</i>	
Hybrid Analytical and ANN-Based Modelling of Temperature Sensors Nonlinear Dynamic Properties	356
<i>Lidia Jackowska-Strumillo</i>	
An Improved Annealing Algorithm for Throughput Maximization in Static Overlay-Based Multicast Systems	364
<i>Michał Kucharzak and Krzysztof Walkowiak</i>	

An Implementation of Differential Evolution for Independent Tasks Scheduling on GPU	372
<i>Pavel Krömer, Jan Platoš, Václav Snášel, and Ajith Abraham</i>	
Collaborative Community Detection in Complex Networks	380
<i>Camelia Chira and Anca Gog</i>	
JCLEC Meets WEKA!	388
<i>A. Cano, J.M. Luna, J.L. Olmo, and S. Ventura</i>	
An Argumentation Framework for Supporting Agreements in Agent Societies Applied to Customer Support	396
<i>Jaume Jordán, Stella Heras, Soledad Valero, and Vicente Julián</i>	
Finger Vein Pattern Extraction Algorithm	404
<i>Michał Waluś, Jan Kosmala, and Khalid Saeed</i>	
An Exploratory Research on Text-Independent Speaker Recognition	412
<i>Mohammad Kheir Nammous, Adam Szczępański, and Khalid Saeed</i>	
Towards Automatic Image Annotation Supporting Document Understanding	420
<i>Urszula Markowska-Kaczmar, Paweł Minda, Krzysztof Ociepa, Dariusz Olszowy, and Roman Pawlikowski</i>	
A Computational Assessment of a Blood Vessel's Compliance: A Procedure Based on Computed Tomography Coronary Angiography	428
<i>Piotr Porwik, Maciej Sosnowski, Tomasz Wesolowski, and Krzysztof Wrobel</i>	
Visual System for Drivers' Eye Recognition.....	436
<i>Bogusław Cyganek and Sławomir Gruszczyński</i>	
A Hybrid Context-Aware Wearable System with Evolutionary Optimization and Selective Inference of Dynamic Bayesian Networks ...	444
<i>Jun-Ki Min and Sung-Bae Cho</i>	
Global/Local Hybrid Learning of Mixture-of-Experts from Labeled and Unlabeled Data	452
<i>Jong-Won Yoon and Sung-Bae Cho</i>	
Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer	460
<i>Young-Seol Lee and Sung-Bae Cho</i>	
Author Index	469

Table of Contents – Part II

Hybrid Intelligent System on Logistics and Intelligent Optimization

- Outlier Analysis for Plastic Card Fraud Detection a Hybridized and Multi-Objective Approach 1
Arturo Elías, Alberto Ochoa-Zezzatti, Alejandro Padilla, and Julio Ponce

- Comparative Analysis of Recombination Operators in Genetic Algorithms for the Travelling Salesman Problem 10
Anca Gog and Camelia Chira

- Multiple Local Searches to Balance Intensification and Diversification in a Memetic Algorithm for the Linear Ordering Problem 18
*Héctor Joaquín Fraire Huacuja,
Guadalupe Castilla Valdez, Claudia G. Gómez Santillan,
Juan Javier González Barbosa, Rodolfo A. Pazos R.,
Shulamith S. Bastiani Medina, and David Terán Villanueva*

- Enhancing Accuracy of Hybrid Packing Systems through General-Purpose Characterization 26
*Laura Cruz-Reyes, Claudia Gómez-Santillán, Satu Elisa Schaeffer,
Marcela Quiroz-Castellanos, Victor M. Alvarez-Hernández, and
Verónica Pérez-Rosas*

Metaheuristics for Combinatorial Optimization and Modelling Complex Systems

- Improving Classification Performance of BCIs by Using Stationary Common Spatial Patterns and Unsupervised Bias Adaptation 34
Wojciech Wojcikiewicz, Carmen Vidaurre, and Motoaki Kawanabe

- A Simple Proactive Provider Participation Technique in a Mesh-Based Peer-to-Peer Streaming Service 42
*Darío Padula, María Elisa Bertinat,
Franco Robledo Amoza, Pablo Rodríguez-Bocca, and
Pablo Romero*

- Modelling Non-stationarities in EEG Data with Robust Principal Component Analysis 51
Javier Pascual, Motoaki Kawanabe, and Carmen Vidaurre

Performance Evaluation of Road Traffic Control Using a Fuzzy Cellular Model	59
<i>Bartłomiej Płaczek</i>	
An Improved Heuristic for the Bandwidth Minimization Based on Genetic Programming	67
<i>P.C. Pop and O. Matei</i>	
About Inducing Simple Emergent Behavior in Large Cournot Games by Using Crowding Based Differential Evolution	75
<i>Rodica Ioana Lung</i>	
An Study of the Tree Generation Algorithms in Equation Based Model Learning with Low Quality Data	84
<i>Alba Berzosa, José R. Villar, Javier Sedano, Marco García-Tamargo, and Enrique de la Cal</i>	
Hybrid Systems for Context-Based Information Fusion	
RT-MLR: A Hybrid Framework for Context-Aware Systems	92
<i>Pablo Rangel, José G. de Carvalho Jr., Milton R. Ramirez, and Jano M. de Souza</i>	
DAFNE – A Distributed and Adaptive Fusion Engine	100
<i>Maarten Ditzel, Sebastiaan van den Broek, Patrick Hanckmann, and Miranda van Iersel</i>	
Context Representation and Fusion via Likelihood Masks for Target Tracking	110
<i>Lauro Snidaro, Ingrid Visentini, and Gian Luca Foresti</i>	
Adaptive Data Fusion for Air Traffic Control Surveillance	118
<i>Juan A. Besada, Guillermo Frontera, Ana M. Bernardos, and Gonzalo de Miguel</i>	
Dynamic Channel Model LMS Updating for RSS-Based Localization ...	127
<i>Paula Tarrío, Ana M. Bernardos, Xian Wang, and José R. Casar</i>	
Improving the Accuracy of Action Classification Using View-Dependent Context Information.....	136
<i>Rodrigo Cilla, Miguel A. Patricio, Antonio Berlanga, and Jos M. Molina</i>	
A General Purpose Context Reasoning Environment to Deal with Tracking Problems: An Ontology-Based Prototype	144
<i>Miguel A. Serrano, Miguel A. Patricio, Jesús García, and José M. Molina</i>	

Methods of Classifiers Fusion

Accuracy Updated Ensemble for Data Streams with Concept Drift	155
<i>Dariusz Brzeziński and Jerzy Stefanowski</i>	
Classifier Ensembles for Virtual Concept Drift – The DEnBoost Algorithm	164
<i>Kamil Bartocha and Igor T. Podolak</i>	
On Performance of DRSA-ANN Classifier	172
<i>Urszula Stańczyk</i>	
Performance Analysis of Fuzzy Aggregation Operations for Combining Classifiers for Natural Textures in Images	180
<i>Maria Guijarro, Gonzalo Pajares, P. Javier Herrera, and J.M. de la Cruz</i>	
A Generalization of Majority Voting Scheme for Medical Image Detectors	189
<i>Henrietta Toman, Laszlo Kovacs, Agnes Jonas, Lajos Hajdu, and Andras Hajdu</i>	
An Efficient Hybrid Classification Algorithm – An Example from Palliative Care	197
<i>Tor Gunnar Houeland and Agnar Aamodt</i>	
An Effective Feature Selection Algorithm Based on the Class Similarity Used with a SVM-RDA Classifier to Protein Fold Recognition	205
<i>Wiesław Chmielnicki and Katarzyna Stępor</i>	
Empirical Comparison of Resampling Methods Using Genetic Neural Networks for a Regression Problem	213
<i>Tadeusz Lasota, Zbigniew Telec, Grzegorz Trawiński, and Bogdan Trawiński</i>	
Structured Output Element Ordering in Boosting-Based Classification	221
<i>Tomasz Kajdanowicz and Przemysław Kazienko</i>	
Probabilistic Approach to the Dynamic Ensemble Selection Using Measures of Competence and Diversity of Base Classifiers	229
<i>Rafał Lysiak, Marek Kurzynski, and Tomasz Wołoszynski</i>	
Complexity and Multithreaded Implementation Analysis of One Class-Classifiers Fuzzy Combiner	237
<i>Tomasz Wilk and Michał Woźniak</i>	
Costs-Sensitive Classification in Multistage Classifier with Fuzzy Observations of Object Features	245
<i>Robert Burduk</i>	

Intelligent Systems for Data Mining and Applications

Fusion of Similarity Measures for Time Series Classification	253
<i>Krisztian Buza, Alexandros Nanopoulos, and Lars Schmidt-Thieme</i>	
Enhancing IPADE Algorithm with a Different Individual Codification	262
<i>Isaac Triguero, Salvador García, and Francisco Herrera</i>	
A Multi-objective Evolutionary Approach for Subgroup Discovery	271
<i>Victoria Pachón, Jacinto Mata, Juan Luis Domínguez, and Manuel J. Maña</i>	
Gene Regulatory Networks Validation Framework Based in KEGG	279
<i>Norberto Díaz-Díaz, Francisco Gómez-Vela, Domingo S. Rodríguez-Baena, and Jesús Aguilar-Ruiz</i>	
Computational Intelligence Techniques for Predicting Earthquakes	287
<i>F. Martínez-Álvarez, A. Troncoso, A. Morales-Estebar, and J.C. Riquelme</i>	
Reduct-Based Analysis of Decision Algorithms: Application in Computational Stylistics	295
<i>Urszula Stańczyk</i>	
Evolutionary Protein Contact Maps Prediction Based on Amino Acid Properties	303
<i>Alfonso E. Márquez Chamorro, Federico Divina, and Jesús S. Aguilar-Ruiz</i>	
A Comparative Study between Two Regression Methods on LiDAR Data: A Case Study	311
<i>Jorge García-Gutiérrez, Eduardo González-Ferreiro, Daniel Mateos-García, Jose C. Riquelme-Santos, and David Miranda</i>	
Analysis of Measures of Quantitative Association Rules	319
<i>M. Martínez-Ballesteros and J.C. Riquelme</i>	

Systems, Man, and Cybernetics by SOCO-Workshop

Supervised Rule Based Thermodynamic Cycles Design Technique	327
<i>Ramon Ferreiro Garcia and Jose Luis Calvo Rolle</i>	
Deformation Based Features for Alzheimer's Disease Detection with Linear SVM	336
<i>Alexandre Savio, Manuel Graña, Jorge Villanúa</i>	

A Hybrid System for Survival Analysis after EVAR Treatment of AAA	344
---	-----

Josu Maiora and Manuel Graña

Plenary

A Hybrid Intelligent System for Generic Decision for PID Controllers Design in Open-Loop	352
--	-----

José Luis Calvo-Rolle, Emilio Corchado, Amer Laham, and Ramón Ferreiro García

Clustering Ensemble for Spam Filtering	363
--	-----

Santiago Porras, Bruno Baruque, Belén Vaquerizo, and Emilio Corchado

Hybrid Artificial Intelligence Systems in Management of Production Systems

Rule-Based Expert System Dedicated for Technological Applications ...	373
---	-----

Edward Chlebus, Kamil Krot, and Michał Kuliberda

Concept of a Data Exchange Agent System for Automatic Construction of Simulation Models of Manufacturing Processes	381
--	-----

Edward Chlebus, Anna Burduk, and Arkadiusz Kowalski

Evaluation of the Risk in Production Systems with a Parallel Reliability Structure Taking into Account Its Acceptance Level	389
---	-----

Anna Burduk

Production Preparation and Order Verification Systems Integration Using Method Based on Data Transformation and Data Mapping	397
--	-----

Bozena Skolud and Damian Krenczyk

Object-Oriented Models in an Integration of CAD/CAPP/CAP Systems	405
--	-----

Cezary Grabowik and Krzysztof Kalinowski

Hybrid Artificial Intelligent Systems for Medical Applications

A Hybrid Artificial Intelligence System for Assistance in Remote Monitoring of Heart Patients	413
---	-----

Theodor Heinze, Robert Wierschke, Alexander Schacht, and Martin von Löwies

- Hybrid Patient Classification System in Nursing Logistics Activities 421
Dragan Simić, Dragana Milutinović, Svetlana Simić, and Vesna Suknaja

- An Approach of Soft Computing Applications in Clinical Neurology 429
Dragan Simić, Svetlana Simić, and Ilija Tanackov

Plenary

- A Hybrid System for Dental Milling Parameters Optimisation 437
Vicente Vera, Javier Sedano, Emilio Corchado, Raquel Redondo, Beatriz Hernando, Monica Camara, Amer Laham, and Alvaro Enrique Garcia

Hybrid Intelligent Approaches in Cooperative Multi-robot Systems

- A Hybrid Color Distance for Image Segmentation 447
R. Moreno, M. Graña, and A. d'Anjou

- Empirical Study of Q-Learning Based Elemental Hose Transport Control 455
Jose Manuel Lopez-Guede, Borja Fernandez-Gauna, Manuel Graña, and Ekaitz Zulueta

- Towards Concurrent Q-Learning on Linked Multi-Component Robotic Systems 463
Borja Fernandez-Gauna, Jose Manuel Lopez-Guede, and Manuel Graña

- Evolutionary Procedure for the Progressive Design of Controllers for Collective Behaviors 471
P. Caamaño, J.A. Becerra, F. Bellas, A. Prieto, and R.J. Duro

- Topos 2: Spiking Neural Networks for Bipedal Walking in Humanoid Robots 479
Pablo González-Nalda and Blanca Cases

- Author Index** 487

Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution

A. Fernández^{1,*}, S. García¹, and F. Herrera²

¹ Dept. of Computer Science, University of Jaén
Tel.:+34-953-213016; Fax:+34-953-212472

{alberto.fernandez,sglopez}@ujaen.es
² Dept. of Computer Science and A.I., University of Granada,
herrera@decsai.ugr.es

Abstract. Classifier learning with datasets which suffer from imbalanced class distributions is an important problem in data mining. This issue occurs when the number of examples representing one class is much lower than the ones of the other classes. Its presence in many real-world applications has brought along a growth of attention from researchers.

The aim of this work is to shortly review the main issues of this problem and to describe two common approaches for dealing with imbalance, namely sampling and cost sensitive learning. Additionally, we will pay special attention to some open problems, in particular we will carry out a discussion on the data intrinsic characteristics of the imbalanced classification problem which will help to follow new paths that can lead to the improvement of current models, namely size of the dataset, small disjuncts, the overlapping between the classes and the data fracture between training and test distribution.

Keywords: Imbalanced Datasets, Sampling, Cost Sensitive Learning, Small Disjuncts, Overlapping, Dataset Shift.

1 Introduction

In many applications, there exists a significant difference between the class prior rates, that is the probability a particular example belongs to a particular class. This situation is known as the class imbalance problem [12] and it is dominant in a high number of real problems including, but not limited to, telecommunications, WWW, finances, ecology, biology, medicine and so on; for which it is considered as one of the top problems in data mining [3]. Furthermore, it is worth to point out that the positive or minority class is usually the one that has the highest interest from the learning point of view and it also implies a great cost when it is not well classified [4].

The hitch with imbalanced datasets is that standard classification learning algorithms are often biased towards the majority classes and therefore there is a

* Corresponding author.

higher misclassification rate in the minority class instances. Therefore, throughout the last year, many solutions have been proposed to deal with this problem, which can be categorised into two major groups:

1. **Data sampling:** in which the training instances are modified in such a way as to produce a balanced class distribution that allow classifiers to perform in a similar manner to standard classification [5,6].
2. **Algorithmic modification:** this procedure is oriented towards the adaptation of base learning methods to be more attuned to class imbalance issues [7]. We must also stress in this case the use of cost-sensitive learning solutions, which basically assume higher misclassification costs with samples in the rare class and seek to minimise the high cost errors [8,9].

In this contribution, our initial goal is to develop a review on this type of methodologies, analysing the different approaches that have been traditionally applied and describing the main features of each one of them.

Additionally, most of the studies on the behavior of several standard classifiers in imbalance domains have shown that significant loss of performance is mainly due to skew of class distributions. However, there are also several investigations which also suggest that there are other factors that contribute to such performance degradation, for example, size of the dataset, small disjuncts, overlapping between classes and dataset shift among others [10,11,12,13,14].

According to the previous issues, we aim to carry out a discussion about the source where the difficulties for imbalanced classification emerge, focusing on the analysis of significant data intrinsic characteristics such as the ones previously mentioned. We must point out that some of these topics have recent studies associated, but that they still need to be addressed in detail in order to have models of quality in this classification scenario.

In order to do so, this contribution is organised as follows. First, Section 2 presents the problem of imbalanced datasets, introducing its features and the metrics employed in this context. Next, Section 3 describes the preprocessing and cost sensitive methodologies that have been proposed to deal with this problem. Section 4 is devoted to analyse and discuss some open problems on the topic. Finally, Section 5 summarises and concludes the work.

2 Imbalanced Datasets in Classification

In the classification problem field, the scenario of imbalanced datasets appears with high frequency. The main property of this type of classification problem is that the examples of one class outnumber examples of the other one [1,2]. The minority classes are usually the most important concepts to be learnt, since they represent rare cases [3] or because the data acquisition of these examples is costly [4].

Since most of the standard learning algorithms consider a balanced training set, this situation may cause the obtention of suboptimal classification models, i.e. a good coverage of the majority examples whereas the minority ones are

misclassified more frequently; therefore, those algorithms which obtains a good behavior in the framework of standard classification do not necessarily achieves the best performance for imbalanced data-sets [17]. There are several reasons behind this behaviour which are enumerated below:

1. The use of global performance measures for guiding the search process, such as standard accuracy rate, may benefit the covering of the majority examples.
2. Classification rules that predict the positive class are often highly specialised and thus their coverage is very low, hence they are discarded in favour of more general rules, i.e. those that predict the negative class.
3. It is always not easy to distinguish between noise examples and minority class examples and they can be completely ignored by the classifier.

In recent years, the imbalanced learning problem has received a high attention in the machine learning community. Specifically, regarding real world domains the importance of the imbalance learning problem is growing, since it is a recurring problem in many applications. As a few examples, we may find very high resolution airbourne imagery [18], remote-sensing [19], face recognition [20] and especially medical diagnosis [21][22].

We must also point out that in imbalanced domains the evaluation of the classifiers' performance must be carried out using specific metrics to take into account the class distribution. Since in this classification scenario we intend to achieve good quality results for both classes, one way to combine the individual measures of both the positive and negative classes, and to produce an evaluation criteria, is to use the Receiver Operating Characteristic (ROC) graphic [23]. This graphic allows to visualise the trade-off between the benefits (TP_{rate}) and costs (FP_{rate}), thus it evidences that any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC) [24] provides a single measure of a classifier's performance for evaluating which model is better on average. Other metrics of interest to be stressed in this area are the geometric mean of the true rates [25] and the F-measure [26].

3 Addressing Classification with Imbalanced Data: Preprocessing and Cost Sensitive Learning

A large number of approaches have been previously proposed to deal with the class-imbalance problem. These approaches can be categorised in two groups: the internal approaches that create new algorithms or modify existing ones to take the class-imbalance problem into consideration [25][27] and external approaches that preprocess the data in order to diminish the effect of their class imbalance [5][28]. Furthermore, cost-sensitive learning solutions incorporating both the data and algorithmic level approaches assume higher misclassification costs with samples in the minority class and seek to minimise the high cost errors [8][29].

Regarding this, in this section we first introduce the main features of preprocessing techniques and next, we describe the cost-sensitive learning approach.

3.1 Preprocessing Imbalanced Datasets: Resampling Techniques

In the specialised literature, we can find some papers about resampling techniques studying the effect of changing class distribution to deal with imbalanced datasets where it has been empirically proved that, applying a preprocessing step in order to balance the class distribution, is usually a positive solution [5,30,31]. The main advantage of these techniques is that they are independent of the underlying classifier.

Resampling techniques can be categorised into three groups: *undersampling methods*, which create a subset of the original dataset by eliminating instances (usually majority class instances); *oversampling methods*, which create a superset of the original dataset by replicating some instances or creating new instances from existing ones; and finally, *hybrids methods*, that combine both sampling approaches.

Among these categories, there exist several different proposals, from which the most simple ones are non heuristic methods such as random undersampling and random oversampling. In the first case, the major drawback is that it can discard potentially useful data, that could be important for the induction process. For random oversampling, several authors agree that this method can increase the likelihood of occurring overfitting, since it makes exact copies of existing instances.

According to the previous facts, more sophisticated methods have been proposed. Among them, the “Synthetic Minority Oversampling Technique” (SMOTE) [6] have become one of the most significant approaches in this area. In brief, its main idea is to create new minority class examples by interpolating several minority class instances that lie together for oversampling the training set.

Regarding undersampling techniques, the application of genetic algorithms for the correct identification of the most significant instance have shown to achieve very positive results [32,33]. Also, a training set selection can be carried out for enhancing the learning stage of several classification algorithms in the area of imbalanced data-sets [34].

Finally, some combination of preprocessing of instances with data cleaning techniques could lead to diminish the overlapping that is introduced from sampling methods. Some representative works in this area [5] include the condensed nearest neighbour rule and Tomek Links integration method, the neighbourhood cleaning rule based on the edited nearest neighbour (ENN) rule which removes examples that differ from two of its three nearest neighbours, and the integrations of SMOTE with ENN and SMOTE with Tomek links.

3.2 Cost-Sensitive Learning

Cost-sensitive learning takes into account the variable cost of a misclassification of the different classes [8,9]. In this case, a cost matrix codifies the penalties of classifying examples of one class as a different one. However, given a dataset, this matrix is not usually given [1,35]. Specifically, when dealing with imbalanced problems it is usually of most interest to recognise the positive instances rather

than the negative ones and therefore, the cost when mistaking a positive instance is higher than the cost of mistaking a negative one.

Three main general approaches have been proposed to deal with cost-sensitive problems:

1. Methods based on modifying the training data set. The most popular technique lies in resampling the original class distribution of the training data set according to the cost decision matrix by means of undersampling/oversampling, modifying decision thresholds or assigning instance weights [9,29].
2. Other methods change the learning process in order to build a cost-sensitive classifiers, for example, in the context of decision tree induction, the tree-building strategies are adapted to minimise the misclassification costs [36].
3. Methods based on the Bayes decision theory that assign instances to the class with minimum expected cost [7,8].

4 Analysing Open Problems Related to Intrinsic Data Characteristics in Imbalanced Classification

As it was stated in the introduction of this work, skewed class distribution does not hinder the learning task by itself [1,2], but usually a series of difficulties related with this problem turn up.

In this section, we aim to develop a discussion on the nature of the problem itself, emphasising several data intrinsic characteristics that do have a strong influence on imbalanced classification, in order to be able to address this problem in a more reasonable way.

First, we present the issues about the size of the dataset and the difficulties related to the apparition of small disjuncts in the data. Next, we focus on the class overlap and how it is extremely significant on imbalanced domains. Finally, we define the dataset shift problem and its relationship to imbalanced datasets classification.

4.1 Small Sample Size and Small Disjuncts

One problem that can arise in classification is a small sample size [32]. This issue is related to the “lack of information” where induction algorithms do not have enough data to make generalisations about the distribution of samples. This problem is increased in the presence of high dimensional data, i.e a large number of features.

The combination of imbalanced data and the small sample size problem presents a new challenge to the research community [38]. In this scenario, the minority class can be poorly represented and the knowledge model to learn this data space become too specific, leading to overfitting. Therefore, two datasets can not be considered to present the complexity with the same imbalance ratio (the ratio between the positive and negative instances [39]) but it is also significant how good do the training data represents the minority instances.

On the other hand, the existence of the imbalanced classes is closely related to the problem of small disjuncts. This situation occurs when the concepts are represented within small clusters, which arise as a direct result of underrepresented subconcepts [11][40]. Although those small disjuncts are implicit in most of the problems, the existence of these small disjuncts highly increases the complexity of the problem in the case of imbalance because it becomes hard to know whether these examples represent an actual subconcept or are merely attributed to noise [41].

4.2 Overlapping or Class Separability

The problem of overlapping between classes appears when a region of the data space contains a similar quantity of training data from each class. This lead to develop an inference with almost the same a priori probabilities in this overlapping area, which makes very hard or even impossible the distinction between the two classes. Indeed, any “linearly separable” problem can be solved by any simple classifier regardless of the class distribution.

There are several works which aim to study the relationship between overlapping and class imbalance. Specifically, in [12] we may find a study where the authors propose several experiments with synthetic datasets varying the imbalance ratio and the overlap existing between the two classes. Their conclusions stated that it is not the class probabilities the main responsible for the hinder in the classification performance, but instead the degree of overlapping between the classes.

Also, in [42] the authors developed a similar study with several algorithms in different situations of imbalance and overlap focusing in the the k-NN algorithm. In this case, the authors proposed two different frameworks: on the one hand, they try to find the relation when the imbalance ratio in the overlap region is similar to the overall imbalance ratio whereas on the other hand, they search for the relation when the imbalance ratio in the overlap region is inverse to the overall one (the positive class is locally denser than the negative class in the overlap region). They shown that when the overlapped data is not balanced, the IR in overlapping can be more important than the overlapping size. In addition, classifiers based on more global learning attain greater TP rates whereas more local learning models obtain better TN rates than the former.

More recent works [13] have extracted empirically some interesting findings with real world datasets. Specifically, the authors depicted the performance of the different datasets ordered according to different data complexity measures (including the IR) in order to search for some regions of interesting good or bad behaviour. They could not characterize any interesting behaviour according IR, but they do for example according the so called metric *F1* or *maximum Fishers discriminant ratio* [43], which measures the overlap of individual feature values.

A closely related issue is the how deep is the impact of noisy and borderline examples from the minority class on the classifier performance [44], and also its relationship with the use of focused re-sampling methods with respect to the simplest random and cluster oversampling.

As a final remark, a positive approach at the algorithm-level could consist in working with different granular levels, in a way that more general submodels of knowledge could cover the largest part of the problem space, whereas in more difficult areas, that is, boundary zones with a high degree of overlapping, we could use more specific discrimination functions in different paradigms of learning algorithms. In [45] the authors introduced a fuzzy system with hierarchical fuzzy partitions for managing specific regions, i.e. the most difficult areas.

4.3 Dataset Shift

The problem of dataset shift [46] is defined as the case where training and test data follow different distributions. This is a common problem that can affect all kind of classification problems, and it often appears due to sample selection bias issues. A mild degree of dataset shift is present in most real-world problems, but general classifiers are often capable of handling it without a severe performance loss.

However, the dataset shift issue is specially relevant when dealing with imbalanced classification, because in highly imbalanced domains, the minority class is particularly sensitive to singular classification errors, due to the typically low number of examples it presents [14]. In the most extreme cases, a single misclassified example of the minority class can create a significant drop in performance. There are two different potential approaches in the study of the effect and solution of dataset shift in imbalanced domains:

1. The first one focuses on intrinsic dataset shift, that is, the data of interest includes some degree of shift that is producing a relevant drop in performance. In this case, we may develop techniques to discover and measure the presence of dataset shift [47], but adapting them to focus on the minority class. Furthermore, we may design algorithms that are capable of working under dataset shift conditions, either by means of preprocessing techniques [48] or with ad hoc algorithms that are [49]. In both cases, we are not aware of any proposals in the literature that focus on the problem of imbalanced classification in the presence of dataset shift.
2. The second branch in terms of dataset shift in imbalanced classification is related to induced dataset shift. Most current state of the art research is validated through stratified cross-validation techniques, which are another potential source of shift in the machine learning process. A more suitable validation technique needs to be developed in order to avoid introducing dataset shift issues artificially.

5 Concluding Remarks

In this contribution we have reviewed the topic of classification with imbalanced datasets, focusing on two main issues: (1) to present the main approaches for dealing with this problem, namely preprocessing of instances and cost-sensitive

learning, and (2) to develop a throughout discussion on the data intrinsic characteristics of this scenario of data mining.

Specifically, we have pointed out that the imbalance ratio have a significant effect on the classifiers' performance, but that there other issues that must be taken into account such as small sample size, small disjuncts, class overlapping and dataset shift. Overcoming these problems can be the key for developing new approaches that improve the correct identification of both the minority and majority classes, and therefore we have stressed them as future trends of research for imbalanced datasets.

Acknowledgment

This work had been supported by the Spanish Ministry of Science and Technology under Project TIN2008-06681-C06-01.

References

1. Sun, Y., Wong, A.K.C., Kamel, M.S.: Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence* 23(4), 687–719 (2009)
2. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21(9), 1263–1284 (2009)
3. Yang, Q., Wu, X.: 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making* 5(4), 597–604 (2006)
4. Elkan, C.: The foundations of cost-sensitive learning. In: Proceedings of the 17th IEEE International Joint Conference on Artificial Intelligence (IJCAI 2001), pp. 973–978 (2001)
5. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behaviour of several methods for balancing machine learning training data. *SIGKDD Explorations* 6(1), 20–29 (2004)
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligent Research* 16, 321–357 (2002)
7. Zadrozny, B., Elkan, C.: Learning and making decisions when costs and probabilities are both unknown. In: Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining (KDD 2001), pp. 204–213 (2001)
8. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: KDD 1999, pp. 155–164 (1999)
9. Zadrozny, B., Langford, J., Abe, N.: Cost-sensitive learning by cost-proportionate example weighting. In: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), pp. 435–442 (2003)
10. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. *Intelligent Data Analysis Journal* 6(5), 429–450 (2002)
11. Weiss, G.M., Provost, F.J.: Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research* 19, 315–354 (2003)

12. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Class imbalances *versus* class overlapping: An analysis of a learning system behavior. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 312–321. Springer, Heidelberg (2004)
13. Luengo, J., Fernández, A., García, S., Herrera, F.: Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling. In: Soft Computing (in press 2011), doi:10.1007/s00500-010-0625-8
14. Moreno-Torres, J.G., Herrera, F.: A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction. In: 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010), pp. 501–506 (2010)
15. Weiss, G.M.: Mining with rarity: a unifying framework. SIGKDD Explorations 6(1), 7–19 (2004)
16. Weiss, G.M., Tian, Y.: Maximizing classifier utility when there are data acquisition and modeling costs. Data Mining and Knowledge Discovery 17(2), 253–282 (2008)
17. Fernandez, A., García, S., Luengo, J., Bernadó-Mansilla, E., Herrera, F.: Genetics-based machine learning for rule induction: State of the art, taxonomy and comparative study. IEEE Transactions on Evolutionary Computation 14(6), 913–941 (2010)
18. Chen, X., Fang, T., Huo, H., Li, D.: Graph-based feature selection for object-oriented classification in VHR airborne imagery. IEEE Transactions on Geoscience and Remote Sensing 49(1), 353–365 (2011)
19. Williams, D., Myers, V., Silvious, M.: Mine classification with imbalanced data. IEEE Geoscience and Remote Sensing Letters 6(3), 528–532 (2009)
20. Kwak, N.: Feature extraction for classification problems and its application to face recognition. Pattern Recognition 41(5), 1718–1734 (2008)
21. Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y., Baker, J.A., Tourassi, G.D.: Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Networks 21(2-3) (2008)
22. Peng, X., King, I.: Robust BMPM training based on second-order cone programming and its application in medical diagnosis. Neural Networks 21(2-3), 450–457 (2008)
23. Bradley, A.P.: The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition 30(7), 1145–1159 (1997)
24. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering 17(3), 299–310 (2005)
25. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition 36(3), 849–851 (2003)
26. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)
27. Ducange, P., Lazzerini, B., Marcelloni, F.: Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets. Soft Computing 14(7), 713–728 (2010)
28. Estabrooks, A., Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data sets. Computational Intelligence 20(1), 18–36 (2004)
29. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering 18(1), 63–77 (2006)
30. Fernández, A., García, S., del Jesus, M.J., Herrera, F.: A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. Fuzzy Sets and Systems 159(18), 2378–2398 (2008)

31. Fernández, A., del Jesus, M.J., Herrera, F.: On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets. *Information Sciences* 180(8), 1268–1291 (2010)
32. García, S., Cano, J., Herrera, F.: A memetic algorithm for evolutionary prototype selection: a scaling up approach. *Pattern Recognition* 41(8), 2693–2709 (2008)
33. García, S., Herrera, F.: Evolutionary under-sampling for classification with imbalanced data sets: Proposals and taxonomy. *Evolutionary Computation* 17(3), 275–306 (2009)
34. García, S., Fernández, A., Herrera, F.: Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. *Applied Soft Computing* 9, 1304–1314 (2009)
35. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* 40(12), 3358–3378 (2007)
36. Ling, C.X., Yang, Q., Wang, J., Zhang, S.: Decision trees with minimal costs. In: ICML (2004)
37. Raudys, S., Jain, A.: Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(3), 252–264 (1991)
38. Wasikowski, M., Chen, X.W.: Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on Knowledge and Data Engineering* 22(10), 1388–1400 (2010)
39. Orriols-Puig, A., Bernadó-Mansilla, E.: Evolutionary rule-based systems for imbalanced datasets. *Soft Computing* 13(3), 213–225 (2009)
40. Orriols-Puig, A., Bernadó-Mansilla, E., Goldberg, D.E., Sastry, K., Lanzi, P.L.: Facetwise analysis of XCS for problems with class imbalances. *IEEE Transactions on Evolutionary Computation* 13, 260–283 (2009)
41. Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. *ACM SIGKDD* 6(1), 40–49 (2004)
42. García, V., Mollineda, R., Sánchez, J.S.: On the k-NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis Applications* 11(3-4), 269–280 (2008)
43. Ho, T., Basu, M.: Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(3), 289–300 (2002)
44. Napierała, K., Stefanowski, J., Wilk, S.: Learning from imbalanced data in presence of noisy and borderline examples. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 158–167. Springer, Heidelberg (2010)
45. Fernández, A., del Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. *International Journal of Approximate Reasoning* 50(3), 561–577 (2009)
46. Quiñonero Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press, Cambridge (2009)
47. Cieslak, D.A., Chawla, N.V.: A framework for monitoring classifiers performance: when and why failure occurs? *Knowledge and Information Systems* 18(1), 83–108 (2009)
48. Moreno-Torres, J.G., Llorà, X., Goldberg, D.E., Bhargava, R.: Repairing Fractures between Data using Genetic Programming-based Feature Extraction: A Case Study in Cancer Diagnosis. *Information Sciences*, doi:10.1016/j.ins.2010.09.018 (in press)
49. Bickel, S., Brückner, M., Scheffer, T.: Discriminative learning under covariate shift. *Journal of Machine Learning Research* 10, 2137–2155 (2009)

A New Tool for the Modeling of AI and Machine Learning Applications: Random Walk-Jump Processes

Anis Yazidi¹, Ole-Christoffer Granmo¹, and B. John Oommen^{2,1}

¹ Dept. of ICT, University of Agder, Grimstad, Norway*

² School of Computer Science, Carleton University, Ottawa, Canada**

Abstract. There are numerous applications in Artificial Intelligence (AI) and Machine Learning (ML) where the criteria for decisions are based on testing procedures. The most common tools used in such random phenomena involve Random Walks (RWs). The theory of RWs and its applications have gained an increasing research interest since the start of the last century. [1]. In this context, we note that a RW is, usually, defined as a trajectory involving a series of successive random steps, which are, quite naturally, modeled using Markov Chains (MCs). MCs are probabilistic structures that possess the so-called “Markov property” – which implies that the next “state” of the walk depends on the current state and not on the entire past states (or history). This imparts to the structure practical consequential implications since it permits the modeler to predict how the chain will behave in the immediate and distant future, and to thus quantify its behavior.

Although Random Walks (RWs) with *single*-step transitions have been extensively studied for almost a century, problems involving the analysis of RWs that contain *interleaving* random steps and random “jumps” are intrinsically hard. In this paper, we consider the analysis of one such fascinating RW, where every step is paired with its counterpart random jump. Apart from this RW being conceptually interesting, it also has applications in the testing of entities (components or personnel), where the entity is never allowed to make more than a pre-specified number of *consecutive* failures. The paper contains the analysis of the chain, some fascinating limiting properties, and simulations that justify the analytic results. The generalization for a researcher to use the same strategy to know when an AI scheme should switch from “Exploration” to “Exploitation” is an extremely interesting avenue for future research.

As far as we know, the entire field of RWs with interleaving steps and jumps is novel, and we believe that this is a pioneering paper in this field, with vast potential in AI and ML.

* The first author gratefully acknowledges the financial support of the *Ericsson Research*, Aachen, Germany, and the third author is grateful for the partial support provided by NSERC, the Natural Sciences and Engineering Research Council of Canada.

** *Chancellor’s Professor; Fellow : IEEE and Fellow : IAPR.* The Author also holds an *Adjunct Professorship* with the Dept. of ICT, University of Agder, Norway.

1 Introduction

The theory of Random Walks (RWs) and its applications have gained an increasing research interest since the start of the last century. From the recorded literature, one perceives that the pioneering treatment of a one-dimensional RW was due to Karl Pearson in 1905 [1]. In this context, we note that a RW is, usually, defined as a trajectory involving a series of successive random steps, which are, quite naturally, modeled using Markov Chains (MCs). MCs are probabilistic structures that possess the so-called “Markov property” – which, informally speaking, implies that the next “state” of the walk depends on the current state and not on the entire past states (or history). The latter property is also referred to as the “memory lack” property, which imparts to the structure practical consequential implications since it permits the modeler to predict how the chain will behave in the immediate and distant future, and to thus quantify its behavior.

RWs have been utilized in a myriad of applications stemming from areas as diverse as biology, computer science, economics and physics. For instance, concrete examples of these applications in biology are epidemics models [2], the Wright-Fisher model, the Moran Model [3], and so on. RWs also arise in the modeling and analysis of queuing systems [4], ruin problems [5], risk theory [6], sequential analysis and learning theory [7]. In addition to the aforementioned *classical* application of RWs, recent applications include mobility models in mobile networks [8], collaborative recommendation [9], web search algorithms [10], and reliability theory for both software and hardware components [11].

RWs can be broadly classified in terms of their Markovian representations. Generally speaking, RWs are either ergodic or possess absorbing barriers. In the simplest case, the induced MC is ergodic, implying that sooner or later, each state will be visited (with probability 1), independent of the initial state. In such MCs, the limiting distribution of being in any state is independent of the corresponding initial distribution. This feature is desirable when the directives dictating the steps of the chain are a consequence of interacting with a non-stationary environment, allowing the walker to not get trapped into choosing any single state. Thus, before one starts the analysis of a MC, it is imperative that one understands the nature of the chain, i.e., if it is ergodic, which will determine whether or not it possesses a stationary distribution.

A RW can also possess absorbing barriers. In this case, the associated MC has a set of transient states which it will sooner or later never visit again, and it, thus, cannot be ergodic. When the walker reaches an absorbing barrier, it is “trapped”, and is destined to remain there forever. RWs with two absorbing barriers have also been applied to analyze problems akin to the two-choice bandit problems [12] and the gambler’s ruin problem [5], while their generalizations to chains with multiple absorbing barriers have their analogous extensions.

Although RWs are traditionally considered to be uni-dimensional (i.e., on the line), multi-dimensional RWs operate on the plane or in a higher dimensional space.

The most popularly studied RWs are those with single step transitions. The properties of such RWs have been extensively investigated in the literature. It is

pertinent to mention that the available results pertaining to RWs in which the chain can move from any state N to *non-neighboring* states $N + k$ or $N - k$ are scanty. Indeed, the analysis of the corresponding MCs is almost impossible in the most general case, *except when certain conditions like time reversibility can be invoked*. Finally, such chains fall completely outside the scope of the so-called family of “birth-and-death” processes, because even in these cases, the number of individuals does not drastically fall or increase in a single time unit (except in the case when one models catastrophes - but even here, MCs of this type are unreported).

Although Random Walks (RWs) with single-step transitions, such as the ruin problem, have been extensively studied for almost a century [13], problems involving the analysis of RWs containing *interleaving* random steps and random jumps are intrinsically hard. In this paper, we consider the analysis of one such fascinating RW, where every step is paired with its counterpart random jump. Apart from this RW being conceptually interesting, it also has applications in the testing of entities (components or personnel), where the entity is never allowed to make more than a pre-specified number of *consecutive* failures.

To motivate the problem, consider the scenario when we are given the task of testing an error-prone component. At every time step, the component is subject to failure, where the event of failure occurs with a certain probability, q . The corresponding probability of the component not failing¹ is p , where $p = 1 - q$. Further, like all real-life entities, the component can operate under two modes, either in the *Well-Functioning* mode, or in the *Mal-Functioning* mode. At a given time step, we aim to determine if the component is behaving well, i.e., in the *Well-Functioning* mode, or if it is in the *Mal-Functioning* mode, which are the two states of nature. It is not unreasonable to assume that both these hypotheses are mutually exclusive, implying that only one of these describes the state of the component at a given time step, thus excluding the alternative. Let us now consider a possible strategy for determining the appropriate hypothesis for the state of nature.

Suppose that the current maintained hypothesis conjectures that the component is in a *Mal-Functioning* mode. This hypothesis is undermined and systematically replaced by the hypothesis that the component is in its *Well-Functioning* mode if it succeeds to realize a certain number N_1 of successive recoveries (or successes). In the same vein, suppose that the current hypothesis conjectures that the component is in its *Well-Functioning* mode. This hypothesis, on the other hand, is invalidated and systematically replaced by the hypothesis that the component is in its *Mal-Functioning* mode if the component makes a certain number $N_2 + 1$ of successive failures. We shall show that such a hypothesis testing paradigm is most appropriately modeled by a RW in which the random steps and jumps are interleaving. To the best of our knowledge, such a modeling paradigm is novel! Further, the analysis of such a chain is unreported in the literature too.

¹ The latter quantity can also be perceived to be the probability of the component *recovering* from a failure, i.e., if it, indeed, had failed at the previous time instant.

The generalization for a researcher to use the same strategy to know when an AI scheme should switch from “Exploration” to “Exploitation” is an extremely interesting avenue for future research.

By way of nomenclature, throughout this paper, we shall refer to p as the “Reward Probability”, and to q as the “Penalty Probability”, where $p + q = 1$.

2 Problem Formulation

2.1 Specification of the State Space and Transitions

We consider the following RW with Jumps (RWJ) as depicted in Figure 1. Let $X(t)$ denote the index of the state of the walker at a discrete time instant ‘ t ’. The details of the RW can be catalogued as below:

1. First of all, observe that the state space of the RW contains $N_1 + N_2 + 1$ states.
2. The states whose indices are in the set $\{1, \dots, N_1\}$ are paired with their counterpart random jump to state 1.
3. The states whose indices fall in the range between the integers $\{N_1 + 1, \dots, N_1 + N_2\}$ are paired with *their* counterpart random jump to state $N_1 + 1$.
4. Finally, the state whose index is $N_1 + N_2 + 1$ is linked to both states 1 and $N_1 + 1$.
5. Essentially, whenever the walker is in a state $X(t) = i$ which belongs to the the set $\{1, \dots, N_1\}$, he has a chance, p , of advancing to the neighboring state $i + 1$, and a chance $q = 1 - p$ of performing a random jump to state 1. Similarly, whenever he is in a state $X(t) = i$ in the set $\{N_1 + 1, \dots, N_1 + N_2\}$, the walker has a chance, q , of advancing to the neighbor state $i + 1$, and a chance p of operating a random jump to state N_1 . However, whenever the walker is in state $N_1 + N_2 + 1$, he has a probability p of jumping to state $N_1 + 1$ and a probability q of jumping to state 1.
6. These rules describe the RWJ completely.

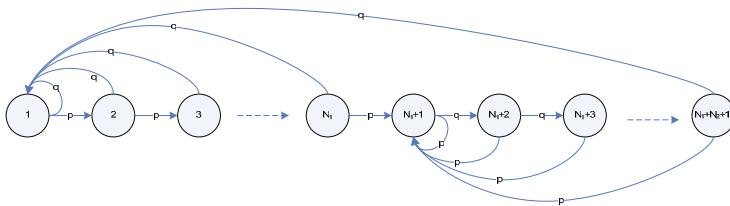


Fig. 1. The state transitions of the Random Walk with Jumps

The reader will observe a marginal asymmetry in the assignment of our states. Indeed, one could query: Why should do we operate with N_1 and $N_2 + 1$ states in the corresponding modes, instead of N_1 and N_2 respectively? Would it not have been “cleaner” to drop the extra state in the latter case, i.e., to use N_2 states

instead of $N_2 + 1$? The reason why we have allowed this asymmetry is because we have consciously intended to put emphasis on the so-called “acceptance state”, that counts as unity. Our position is that this is also a more philosophically-correct position – because the acceptance state is really one which has already obtained a success.

2.2 Application to Component Testing

As briefly alluded to earlier, from a philosophical point of view, the testing of a component can be modeled by the RWJ presented in Section 2.1. At each time step, the entity is either subject to a success or a failure, and is either supposed to be in the *Well-Functioning* or *Mal-Functioning* mode. From a high level perspective, a success “enforces” the hypothesis that the entity is *Well-Functioning* while simultaneously “weakening” the hypothesis that it is *Mal-Functioning*. On the other hand, a failure “enforces” the hypothesis that the entity is deteriorating, i.e *Mal-Functioning*, while “weakening” the hypothesis that it is *Well-Functioning*. It is worth noting that states whose indices are in the set $\{1, \dots, N_1\}$ serve to memorize the number of consecutive successes that have occurred so far. In other words, if the walker is in state i ($i \in \{1, \dots, N_1\}$), it implies that we can deduce that the walker has passed the test i consecutive times. Similarly, states whose indices are in the set $\{N_1 + 1, \dots, N_1 + N_2 + 1\}$ present an indication of the number of consecutive failures that have occurred. In this case, if the walker is in state $N_1 + i$ where $0 < i \leq N_2 + 1$, we can infer that the walker has made i consecutive failures so far.

We present the following mapping of the states of the RW $\{1, \dots, N_1 + N_2 + 1\}$ to the set of hypotheses $\{\text{Well-Functioning}, \text{Mal-Functioning}\}$ as follows. The mapping is divided into two parts:

Mal-Functioning States: We refer to the states $\{1, \dots, N_1\}$ as being the so-called *Mal-Functioning* states, because whenever the index $X(t)$ of the current state of the walker is in that set, we conjecture that “the hypothesis that the component is in its *Mal-Functioning* mode” is true. In this phase, the state transitions illustrated in the figure are such that any deviance from the hypothesis is modelled by a successful transition to the neighboring state, while a failure causes a jump back to state 1. Conversely, only a pure uninterrupted sequence of N_1 successes will allow the walker to pass into the set of *Well-Functioning* states.

Well-Functioning states: We refer to the states $\{N_1 + 1, \dots, N_1 + N_2 + 1\}$ as the *Well-Functioning* states, because when in this set of states, we conjecture the hypothesis that the component is in its *Well-Functioning* mode. More specifically, we refer to state $N_1 + 1$ as being an “acceptance” state because, informally speaking, whenever the walker is in that state, the conjectured hypothesis, that the component is *Well-Functioning*, has been confirmed with highest probability. In particular, within these states, the goal is to detect when the entity deteriorates, causing it to degrade into one of the *Mal-Functioning* states. These states can be perceived to be the “opposite”

of the *Mal-Functioning* states in the sense that an uninterrupted sequence of failures is required to “throw” the walker back into the *Mal-Functioning* mode, while a single success reconfirms the conjectured hypothesis that the component is functioning well, forcing the walker to return to the *Well-Functioning* state space.

We shall now present a detailed analysis of the above RWJ.

3 Theoretical Results

The analysis of the above-described RWJ is particularly difficult because the stationary (equilibrium) probabilities of being in any state is related to the stationary probabilities of *non-neighboring* states. In other words, it is not easy to derive a simple difference equation that relates the stationary probabilities of the neighboring states. To render the analysis more complex, we observe that the RW does not possess any time-reversibility properties either!

However, by studying the peculiar properties of the chain, we have succeeded in solving for the stationary probabilities, which, in our opinion, is far from trivial.

Theorem 1. *For the RWJ described by the Markov Chain given in Figure 1, P_1 , the probability of the walker being in the Mal-Functioning mode, is given by the following expression:*

$$P_1 = \frac{(1 - p^{N_1})q^{N_2}}{(1 - p^{N_1})q^{N_2} + p^{N_1-1}(1 - q^{N_2+1})}. \quad (1)$$

Similarly, P_2 , the probability of being in the Well-Functioning mode (or exiting from the Mal-Functioning mode) is:

$$P_2 = \frac{(1 - q^{N_2+1})p^{N_1-1}}{(1 - p^{N_1})q^{N_2} + p^{N_1-1}(1 - q^{N_2+1})}. \quad (2)$$

Proof: The proof is quite involved and is found in [14]. It is omitted here in the interest of brevity. \square

3.1 “Balanced Memory” Strategies

Although the results obtained above are, in one sense, pioneering, the question of understanding *how* the memory of the scheme should be assigned is interesting in its own right. To briefly address this, in this section, we consider the particular case where N_1 and $N_2 + 1$ are both equal to the same value, N . In this case, if $p = q = 1/2$, one can trivially confirm that $P_1 = P_2 = 1/2$, implying that the scheme is not biased towards any of the two modes, the *Mal-Functioning* or the *Well-Functioning* mode. In practice, employing a “Balanced Memory” strategy seems to be a reasonable choice since having equal memory depth (or number of states) for the *Mal-Functioning* and *Well-Functioning* modes eliminates any bias towards any of the conjectured hypotheses.

Theorem 2. For a “Balanced Memory” strategy in which $N_1 = N_2 + 1 = N$, the probability, P_1 , of being in the Mal-Functioning mode, approaches 0 as the memory depth N tends to infinity whenever $p > 0.5$. Formally, $\lim_{N \rightarrow \infty} P_1 = 0$.

The analogous result for the case when $p < 0.5$ follows.

Theorem 3. For a “Balanced Memory” strategy in which $N_1 = N_2 + 1 = N$, the probability, P_1 , of being in the Mal-Functioning mode, approaches unity as the memory depth N tends to infinity whenever $p < 0.5$. Formally, $\lim_{N \rightarrow \infty} P_1 = 1$.

Proof: The proofs of Theorem 2 and Theorem 3 are omitted here due to space limitations. \square

3.2 Symmetry Properties

The MC describing the RW with Jumps is described by its state occupation probabilities and the overall mode probabilities, P_1 and P_2 . It is trivial to obtain P_1 from P_2 and vice versa for the symmetric “Balanced Memory” case – one merely has to replace p by q and do some simple transformations. However, the RW also possesses a fascinating property when it concerns the underlying state occupation probabilities – the $\{\pi\}$ ’s themselves. Indeed, we shall derive one such interesting property of the scheme in Theorem 4, which specifies a rather straightforward (but non-obvious) method to deduce the equilibrium distribution of a “Balanced Memory” scheme possessing a reward probability p from the equilibrium distribution of the counterpart “Balanced Memory” scheme possessing a reward probability of $1 - p$.

Theorem 4. Let $\Pi = [\pi_1, \dots, \pi_{2N}]^T$ be the vector of steady state probabilities of a balanced scheme characterized by a reward probability p and penalty probability q . Let $\Pi' = [\pi'_1, \dots, \pi'_{2N}]^T$ be the vector of steady state probabilities of a balanced scheme possessing a reward probability $p' = 1 - p$ and penalty probability $q' = 1 - q$. Then Π' can be deduced from Π using the following transformation:

$$\pi'_k = \pi_{\sigma(k)} \text{ for } k \text{ with } 1 \leq k \leq 2N,$$

where σ is a circular permutation of the set $S = \{1, 2, \dots, 2N\}$ defined by

$$\sigma(k) = \begin{cases} 2N, & \text{if } k = N \\ (k + N)(\text{mod } 2N), & \text{Otherwise} \end{cases}$$

Proof: The proof is omitted due to space limitations and is in [14]. \square

4 Experimental Results

Apart from the above theoretical results, we have also rigorously tested the RWJ which we have studied in various experimental settings. In this section, we present some experimental results for cases where the RWJ has been simulated. The goal of the exercise was to understand the sensitivity of the MC to changes

in the memory size, the properties of P_1 as a function of the reward probability, the limiting (asymptotic) behavior of the walk, and the characteristics of the RW in non-stationary environments. Although the chain has been simulated for a variety of settings, in the interest of brevity, we present here only a few typical sets of results – essentially, to catalogue the overall conclusions of the investigation.

4.1 Sensitivity to the RWJ to Changes in the Memory Size

The first study that we undertook was to understand the characteristics of the chain for changes in the memory size. The results obtained have been recorded in Table I, which summarizes the performance of the MC, after convergence, for a wide range of reward probabilities, p , numbers of *Mal-Functioning* states, N_1 , and *Well-Functioning* states, $N_2 + 1$. The resulting performance is then reported in the table in terms of the asymptotic mode occupation probability P_1 , where P_1 has been estimated by averaging over 1,000 experiments, each consisting of 100,000 iterations.

Table 1. The effects of varying the memory size and the reward probability on P_1

(N_1, N_2)	$p = 0.9$	$p = 0.8$	$p = 0.5$	$p = 0.2$	$p = 0.1$
(1, 5)	9.9E-7	6.4E-5	0.015	0.262	0.531
(2, 5)	2.1E-6	1.4E-4	0.045	0.680	0.925
(3, 5)	3.3E-6	2.4E-4	0.1	0.916	0.992
(4, 5)	4.7E-6	3.6E-4	0.192	0.982	0.999
(5, 5)	6.2E-6	5.2E-4	0.329	0.996	0.999
(5, 4)	6.2E-5	2.6E-3	0.5	0.997	0.999
(5, 3)	6.2E-4	0.012	0.673	0.998	0.999
(5, 2)	6.2E-3	0.062	0.815	0.998	0.999
(5, 1)	0.059	0.254	0.911	0.999	0.999

From the experimental results we can conclude the following:

1. For any given values of N_1 and N_2 , the value of P_1 generally decreases monotonically with p . Thus, when N_1 and N_2 are 3 and 5 respectively, the value of P_1 increases from its lowest value of 3.3×10^{-6} for $p = 0.9$, to 0.916 for $p = 0.1$. This is as expected.
2. For any given value of p , if the value of N_2 is fixed, the value of P_1 generally increases monotonically with N_1 . Thus, when p is 0.1 and $N_2 = 5$, the value of P_1 increases from its lowest value of 0.531 for $N_1 = 1$, to 0.9996 for $N_1 = 5$. This too is as expected.
3. Finally, in order to observe the effect of the size of the memory, we consider the column in the Table I for a reward probability $p = 0.5$. Note that the configuration $(N_1, N_2) = (5, 4)$ corresponds to a balanced memory scheme.

If $N_2 + 1 > N_1$, the the walk is concentrated in the *Well-Functioning* states, which has the effect of minimizing P_1 . On the other hand, if $N_1 > N_2 + 1$, the the walk is concentrated in the *Mal-Functioning* states, which, in turn, has the effect of maximizing P_1 .

4.2 Investigation of the MCJ for Non-stationary Environments

In this final experiment, we were interested in understanding the characteristics of the chain when interacting with a non-stationary environment (i.e., when p changed with time). To do this, we modeled a non-stationary environment by altering the reward probability, p , governing the state transitions at every 100th time slot. Then, as before, P_1 was estimated by averaging over 1,000 experiments, each consisting of 100,000 iterations. In particular, the reward probabilities that we used in the experiments were drawn sequentially every 100 time instants from the reward vector $R = [0.2, 0.8, 0.5, 0.1, 0.7, 0.4, 0.9, 0.1]$. To be more specific, between time instants 0 and 100 the reward probability was equal to 0.2, between instants 100 and 200 the reward probability is equal to 0.8 and so on. In Figure 2, we have plotted the average value of P_1 over the set of experiments using a continuous line, and the target distribution of P_1 using a discontinuous (dashed) line, when $N = 3$, where the specific target distribution of P_1 was explicitly calculated using Theorem 1, in which we substituted p with the respective element of the reward vector, R , currently used. For example, between instants 0 and 100, since the reward probability was $p = 0.2$, the target distribution of P_1 was computed to be 0.9701. Similarly, between instants 100 and 200, since the reward probability was $p = 0.8$, the corresponding target distribution was seen to be $P_1 = 0.029$. In Figure 3, we have reported the results of re-doing the same experiment (as in Figure 2), however with a different memory $N = 5$. From Figures 2 and 3, we observe that the instantaneous value of P_1 tracks the target distribution drawn in discontinuous line in a in

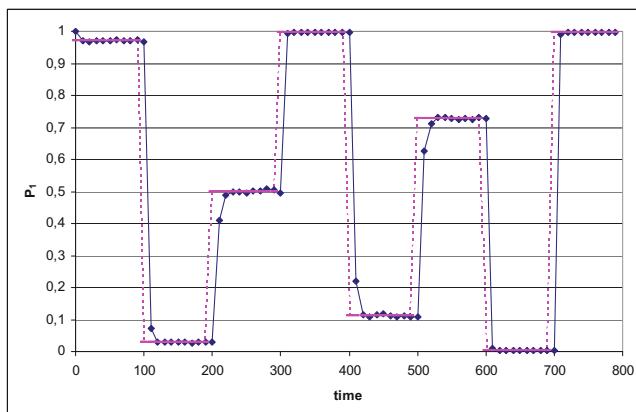


Fig. 2. Ability of the scheme to track the target distribution with $N = 3$

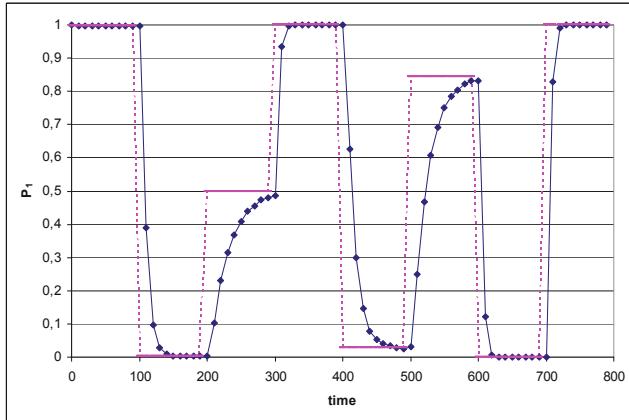


Fig. 3. Ability of the scheme to track the target distribution with $N = 5$

near-optimal manner, which we believe is quite fascinating! The use of this to track the time-varying testing of components is obvious!

5 Conclusions

In this paper, we have studied a novel Random Walk (RW) with interleaving steps and jumps, which has potential applications in the area of component testing. Although RWs with single-step transitions have been extensively studied for almost a century [13], problems involving the analysis of RWs that contain interleaving random steps and random “jumps” are intrinsically hard. In this paper, we have presented the theoretical properties of one such fascinating RW, where every step is paired with its counterpart random jump. As mentioned, the RW has applications in the testing of entities (components or personnel), because we can constrain the entity to never be allowed to make more than a pre-specified number of *consecutive* failures. We believe that the entire field of RWs with interleaving steps and jumps is novel, and we believe that this is a pioneering paper in this field.

References

1. Pearson, K.A.R.L.: The Problem of the Random Walk. *Nature* 72(1867), 342 (1905)
2. Berg, H.C.: *Random Walks in Biology*. Revised edn. Princeton University Press, Princeton (1993)
3. Nowak, M.A.: *Evolutionary Dynamics: Exploring the Equations of Life*. Belknap Press of Harvard University Press (September 2006)
4. Gross, D., Harris, C.M.: *Fundamentals of Queueing Theory* (Wiley Series in Probability and Statistics). Wiley Interscience, Hoboken (1998)
5. Takacs, L.: On the classical ruin problems. *Journal of the American Statistical Association* 64(327), 889–906 (1969)

6. Paulsen, J.: Ruin theory with compounding assets – a survey. *Insurance: Mathematics and Economics* 22(1), 3–16 (1998); Special issue on the interplay between insurance, finance and control
7. Bower, G.H.: A turning point in mathematical learning theory. *Psychological Review* 101(2), 290–300 (1994)
8. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communications & Mobile Computing (WCMC)*: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications 2(5), 483–502 (2002)
9. Fouss, F., Pirotte, A., Renders, J.-M., Saerens, M.: Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.* 19(3), 355–369 (2007)
10. Altman, A., Tennenholtz, M.: Ranking systems: the pagerank axioms. In: EC 2005: Proceedings of the 6th ACM Conference on Electronic Commerce, pp. 1–8. ACM, New York (2005)
11. Bishop, P.G., Pullen, F.D.: A random walk through software reliability theory, pp. 83–111 (1991)
12. Oommen, B.J.: Absorbing and ergodic discretized two-action learning automata. *IEEE Transactions on Systems, Man, and Cybernetics SMC-16*, 282–293 (1986)
13. Feller, W.: An Introduction to Probability Theory and Its Applications, 3rd edn., vol. 1. Wiley, Chichester (1968)
14. Yazidi, A., Grammo, O.C., Oommen, B.: On the analysis of a random walk-jump process with applications to testing, unabridged version of this paper (submitted for publication)

Pattern Recognition Based on Similarity in Linear Semi-ordered Spaces

Juliusz L. Kulikowski and Małgorzata Przytulska

M. Nalecz Institute of Biocybernetics and Biomedical Engineering PAS,

4 Ks. Trojdena Str., 02-109 Warsaw, Poland

{juliusz.kulikowski,malgorzata.przytulska}@ibib.waw.pl

Abstract. In the paper an approach to pattern recognition based on a notion of similarity in linear semi-ordered (Kantorovitsch) space is presented. It is compared with other approaches based on the metric distance and on angular dilation measures in observation spaces. Basic assumptions of the Kantorovitsch space are shortly presented. It is shown that finite reference sets for pattern recognition take on in Kantorovitsch space formal structures presented by connectivity graphs which facilitate finding the reference vectors for pattern recognition.

Keywords: pattern recognition, similarity measures, semi-ordering, Kantorovitsch space.

1 Introduction

Pattern recognition (*PR*) can roughly be defined as an art of assigning to individual objects proper names of classes (*patterns*) consisting of similar to them in an assumed sense objects. The notion of *similarity* in *PR* plays thus a basic role. In general, its interpretation depends on the type of the recognized objects' formal representation. It is usually assumed that the objects are elements of an universe U where the patterns $C^{(1)}, C^{(2)}, \dots, C^{(L)}$ as its subsets are defined. A great deal of pattern recognition methods is based not on a priori established but on gradually, step-by-step extended and iteratively delineated patterns. This is particularly important in the case of recognition of objects belonging to rarely observed patterns (e.g. of syndromes of rare diseases). Many of on artificial neural networks based learning systems as well as many other on learning sets based *PR* systems belong to this category. In such cases the patterns are defined as some, on a similarity measure based inductive extension of finite subsets of objects which formerly as the members of the corresponding patterns have been recognized. The similarity criteria as well as the corresponding patterns description on geometrical, functional, statistical, fuzzy sets, topological, formal-linguistic, set-algebraic and some other approaches can be based [2,3,4]. Practical usability of the above-mentioned methods of patterns characterization in different situations is different and none of them for *PR* purposes as the most effective one can be recommended. In general, if no ambiguity of the *PR* decisions is allowed, the patterns are assumed to be pair-wise disjoint and as such, they are considered as similarity classes

generated by a reciprocal, symmetrical and transitive, i.e. a *strong similarity relation* in the universe U . However, two basic types of objects' similarity should be distinguished:

- a/ Direct similarity of objects in the universe U ;
- b/ Indirect similarity of objects' representations in an auxiliary features' space W .

In the last case, it is assumed that there is given an univocal irreversible projection $U \rightarrow W$ such that projection of a family of similarity classes in U leads also to a partition of W into a family of pair-wise disjoint similarity classes. Hence, PR of objects representations in W is equivalent to PR of the original objects in U . Usually, W is assumed to be a multi-dimensional linear vector space; its elements w will be called objects representing *observations*. This makes possible to define a *similarity measure* as a function [6]:

$$\sigma: W \times W \rightarrow [0, \dots, 1] \quad (1)$$

satisfying the conditions:

$$\left. \begin{array}{l} a) \quad \sigma(w^{(i)}, w^{(i)}) \equiv 1, \\ b) \quad \sigma(w^{(i)}, w^{(j)}) \equiv \sigma(w^{(j)}, w^{(i)}), \\ c) \quad \sigma(w^{(i)}, w^{(j)}) \cdot \sigma(w^{(j)}, w^{(k)}) \leq \sigma(w^{(i)}, w^{(k)}) \end{array} \right\} \quad (2)$$

for any $w^{(i)}, w^{(j)}, w^{(k)} \in W$. On the basis of a given similarity measure a strong (reciprocal, symmetrical and transitive) similarity relation in W can be established. For this purpose it is necessary: 1st to fix a threshold δ , $0 < \delta \leq 1$, and 2nd to assume that *similar* are any and only such two observations $w^{(i)}, w^{(j)}$ that $\sigma(w^{(i)}, w^{(j)}) > \delta$. So-defined similarity relation generates a partition of the space W into a family of pair-wise disjoint and in the similarity sense integral similarity classes. Moreover, for a given similarity measure and a fixed natural odd number κ the following " κ -MSO" (κ Most Similar Objects) PR rule can be formulated:

If : 1st $N^{(1)}, N^{(2)}, \dots, N^{(L)}$, $1 < L$, is a family of pair-wise disjoint finite subsets of W given, respectively, as representatives of the similarity classes $C^{(1)}, C^{(2)}, \dots, C^{(L)}$ in the universe U ; 2nd there is given an observation $w^{(0)} \in W$; 3rd it has been chosen a set $\{w^{(1)}, w^{(2)}, \dots, w^{(\kappa)}\} \subseteq (N^{(1)} \cup N^{(2)} \cup \dots \cup N^{(L)})$ of κ the most similar to $w^{(0)}$ observations among which $l^{(1)}$ belong to $N^{(1)}$, $l^{(2)}$ belong to $N^{(2)}, \dots, l^{(L)}$ belong to $N^{(L)}$, and 4th $l^{(s)} = \max\{l^{(1)}, l^{(2)}, \dots, l^{(L)}\}$ then it should be decided: $w^{(0)}$ is a representative of $C^{(s)}$.

Two types of similarity measures based on: 1/ a metric distance $d(w^{(i)}, w^{(j)})$, 2/ an angular measure $\alpha(w^{(i)}, w^{(j)})$, are particularly useful in many practical applications. In the first case, the above-formulated κ -MSO rule becomes equivalent to the widely known k -NN (k Nearest Neighbor) rule [8]. In the case of an angular measure based similarity the κ -MSO approach can also be used. However, in both cases it arises a problem of PR rules *extensibility*. We call a PR rule *extensible* if inclusion, according to this rule, of an observation $w^{(0)}$ into a similarity class $N^{(s)}$ does not destroy the similarity class integrity. This PR rule's property is of particular importance if on the basis of it a self-organizing (non-supervised) PR system is to be constructed. It is easy to show that the commonly-known k -NN rule does not satisfy the

extensibility requirement. Really, if e.g., there are given in W two similarity classes: $N^{(1)}$ and $N^{(2)}$, maximum distance of elements belonging to the same similarity class should be not higher than 10 units, $N^{(1)}$ is represented by two observations whose mutual distance is equal to 6 units and the distance of a new observation $w^{(0)}$ to the nearest to it observation in $N^{(1)}$ is 5 units and this to the next nearest one in $N^{(2)}$ is 8 units then $w^{(0)}$ will be included into $N^{(1)}$. However, in such case the holding in a metric space “triangle inequality” guarantees only the distance between $w^{(0)}$ and the next observation in $N^{(1)}$ being not higher than $6 + 5 = 11$ units. Hence, it may happen that it is higher than 10 units. Therefore, an extension of $N^{(1)}$ by inclusion to it $w^{(0)}$ is not admissible despite a correct recognition of $w^{(0)}$ as an element more to $N^{(1)}$ than to $N^{(2)}$ similar.

It is easy to show that a κ -MSO on angular measure based rule is also not extensible. Moreover, in such case even the “triangular inequality” weakening the effect of disintegration of distance-based similarity classes does not hold. The aim of this paper is to present a modification of the PR method based on angular measure similarity so as to make it extensible. The concept is based on the notion and properties of linear vector space (*Kantorovitsch space, K-space* [5]) used as observation space. The structure of the paper is as follows: in Sec. 2 formal definition and selected properties of K -spaces are shortly presented. In Sec. 3 geometrical models of similarity classes in metric, unitary and K -space are compared in order to illustrate some of their practical aspects. Sec. 4 contains a description of PR algorithm and presents some results of its application to biomedical textures recognition. Conclusions are given in Sec. 5.

2 K-Space as an Observation Space

A K -space can be defined as a multidimensional linear vector space in which a rule of semi-ordering of its elements (vectors) has been introduced [1,5]. Space linearity means that the operations of: 1st addition of vectors, 2nd vectors multiplication by real numbers, and 3rd existence of a null vector θ in the given space have been assumed [9]. A difference of vectors $\Delta w = w'' - w'$ then can be defined as a sum $\Delta w = w'' + (-1)w'$. Semi-ordering of vectors in K -space is weaker than this of real numbers where for any pair x, y exactly one of the relationships: $x < y$, $x > y$ or $x = y$ holds. For any two vectors w', w'' in K -space one of the following relationships is admissible:

- $w' \prec w''$ (w' is lower than w''),
- $w' \succ w''$ (w' is higher than w''),
- $w' = w''$ (w' is equal to w''),
- $w' ? w''$ (w' is incomparable to w'').

First two relationships are anti-symmetrical, two next ones are symmetrical. For identification of one of the above-given relationships the difference $\Delta w = w'' - w'$ should be taken into consideration. It is assumed that any vector w in the K -space may be *positive*, *negative* or *neutral*, excepting the null-vector θ being both, positive and negative. All positive vectors form a convex cone K^+ whose vertex is θ . All vectors $(-1)w$ such that $w \in K^+$ are by definition *negative*; the set of all negative vectors form

a negative cone K^- , anti-symmetrical to K^+ . All vectors in the K -space belonging neither to K^+ nor to K^- are by definition neutral.

The relationship:

$$\left. \begin{array}{l} w' \prec w'' \text{ holds if } \Delta w = w'' - w' \in K^+, \\ w' \succ w'' \text{ holds if } \Delta w = w'' - w' \in K^-, \\ w' = w'' \text{ holds if } \Delta w = w'' - w' = \theta, \\ w' ? w'' \text{ holds otherwise.} \end{array} \right\} \quad (3)$$

Fig. 1 illustrates the cases of the difference Δw belonging (a) and not belonging (b) to the cone K^+ . It is noticeable that a sum of any positive vectors in K -space is also positive, b/ a sum of any negative vectors is negative, while c/ a sum of neutral vectors in various cases may be positive, negative or neutral. The rate of positive and negative vectors with respect to the neutral ones depends on the dilation of the positive cone K^+ .

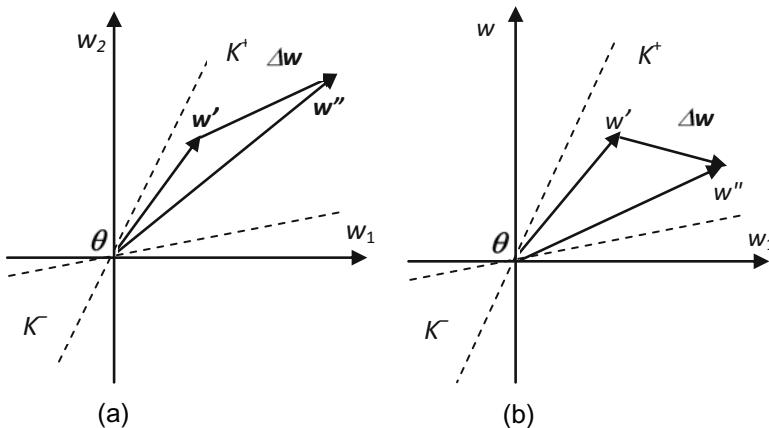


Fig. 1. Different relationships between vectors in K -space:
a) $w' \prec w''$, b) $w' ? w''$

For a given positive cone K^+ it can be established that two vectors w' , w'' are *similar* if either $\Delta w \in K^+$ or $-\Delta w \in K^+$. The larger is the dilation of $\Delta w \in K^+$ the more tolerant is the similarity criterion. There are many ways the positive cone can be established. The simplest one consists in fixing an axis vector c and a vertex angle γ satisfying the following conditions:

1. The norm (length) of the axis vector c is $\|c\| = 1$;
2. The angle β between c and an unity-vector $I = [1, 1, \dots, 1]$ is not greater than $\pi/2$;
3. The direction of c prefers vector components of higher importance in the given application problem;
4. The vertex angular dilation γ is such that $\beta + \gamma < \pi/2$ and it corresponds to the desired similarity tolerance.

Each finite set N of vectors in a K -space can be represented by its *connectivity graph* $G = [N, S]$ such that the elements of N are represented by the nodes while S is a set of

arcs s_{ij} assigned to the ordered pairs of vectors $[w^{(i)}, w^{(j)}]$ of N such that $w^{(i)} \prec w^{(j)}$ and there is no other vector $w^{(k)}$ in N such that $w^{(i)} \prec w^{(k)} \prec w^{(j)}$. The connectivity graph G may consist of one or more *graph components*. A graph component is a sub-graph based on a maximal subset $N' \subset N$ of nodes such that any pair of its nodes can be linked by a chain of arcs. The graph component is called *weakly compact* if among the chains linking the pairs of its nodes there exist some chains consisting of not uniformly directed arcs; otherwise, the graph component is called *compact*.

3 Geometrical Models of Similarity Classes

Choosing of a formal type of observation space W influences on the geometrical form of similarity classes. This is illustrated in Fig. 2 where three types of observation spaces: a) metric space, b) unitary space, c) K -space are presented.

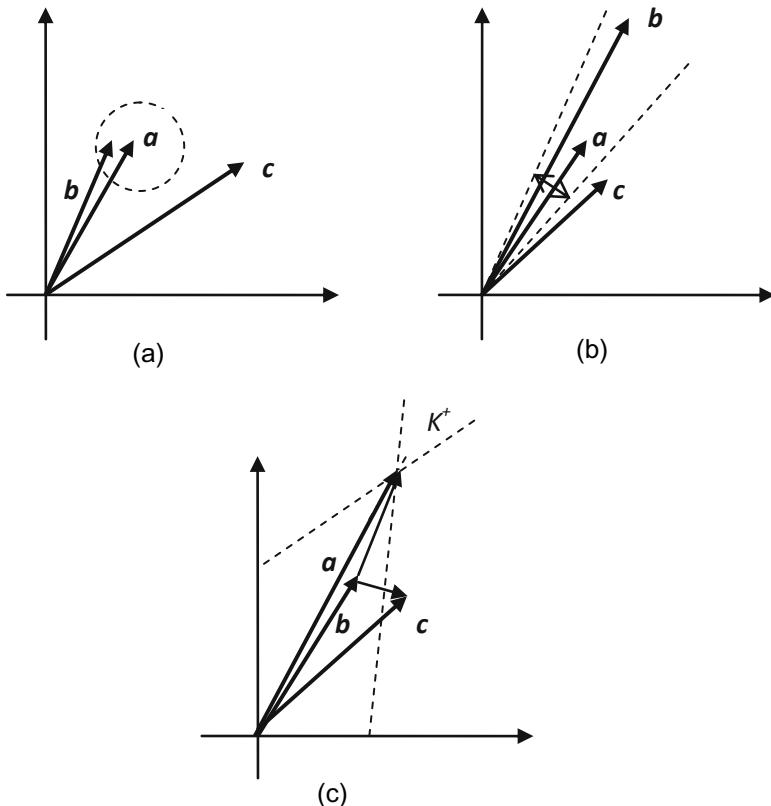


Fig. 2. Similarity and dissimilarity of vectors in different spaces:
a) metric space, b) unitary space, c) K -space.

In all cases three observation vectors: a, b, c are shown. In metric space all vectors assumed to be similar to a belong to a sphere surrounding a with a fixed radius.

In unitary space all vectors similar to \mathbf{a} are declined from it by an angle of limited dilation. In K -space similar to \mathbf{a} are vectors different from \mathbf{a} by a positive or negative vector. In Fig. 2 similar are the pairs \mathbf{a} and \mathbf{b} while dissimilar are the pairs \mathbf{a} and \mathbf{c} as well as \mathbf{b} and \mathbf{c} . The above-mentioned differences have in PR the following practical consequences:

1. Similarity based on metric distance models is sensible to observation vectors scaling while this on angular measure (unitary or K -space) based is scale-invariant.
2. PR robustness to additive noise depends on the area behind the similarity areas; it is higher in unitary and in K -space than in metric space based models.
3. PR methods based on angular measure in unitary space or on K -space based models are more robust to multiplicative distortions than those based on metric distance based models.
4. Let $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \dots, \mathbf{w}^{(r)}$ be a series of vectors linearly ordered so that a distance δ (metric or angular) between any two consecutive vectors is limited by a value δ_0 . We consider this series as candidates to be used as a reference set of observations outlining a similarity class C of observations whose distance is not greater than δ^* , $\delta_0 < \delta^*$. Then the maximal distance between the extreme vectors $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(r)}$ satisfies the inequality:

$$\max \delta_{\mathbf{w}^{(1)}, \mathbf{w}^{(r)}} \leq (r - 1) \cdot \delta_0. \quad (4)$$

Hence, it may happen that the maximal distance surpasses the value δ^* established by the definition of the similarity class. Therefore, the last observations in the series cannot be included into the reference set of the similarity class C without additional similarity tests.

In K -space the situation is different. Let the series of observations be such that the following relationships hold:

$$\mathbf{w}^{(1)} \prec \mathbf{w}^{(2)} \prec \dots \prec \mathbf{w}^{(r)}. \quad (5)$$

Let \mathbf{w} be a new observation such that for a certain pair $\mathbf{w}^{(i)}, \mathbf{w}^{(i+1)}$ of observation in the series it is

$$\mathbf{w}^{(i)} \prec \mathbf{w} \prec \mathbf{w}^{(i+1)}. \quad (6)$$

Then it can easily be shown that

$$\mathbf{w}^{(1)} \prec \mathbf{w}^{(2)} \prec \dots \prec \mathbf{w}^{(i)} \prec \mathbf{w} \prec \mathbf{w}^{(i+1)} \prec \dots \prec \mathbf{w}^{(r)}. \quad (7)$$

Therefore, under some conditions a new vector \mathbf{w} can be included into the reference set without necessity of proving its similarity to all elements of this set.

4 Pattern Recognition in K -Spaces

The κ -MSO rule of PR can be modified so as to be used to the observations represented in the K -space. For this purpose it is assumed that for each pattern $C^{(1)}, C^{(2)}, \dots, C^{(L)}$ in the K -space W reference sets $N^{(1)}, N^{(2)}, \dots, N^{(L)}$ consisting of correctly in the corresponding patterns recognized observations are given. The reference sets are finite and such that the following conditions are satisfied:

1. For any pair of vectors w' , w'' belonging to two different reference sets it is $w' \prec w''$;
2. Each reference set $N^{(\lambda)}$ consists of vectors which can be represented by a weakly compact connectivity graph $G^{(\lambda)}$.

Example of a reference set $N^{(\lambda)}$ is shown in Fig. 3. It consists of 7 vectors among which the below-given relationships have been established:

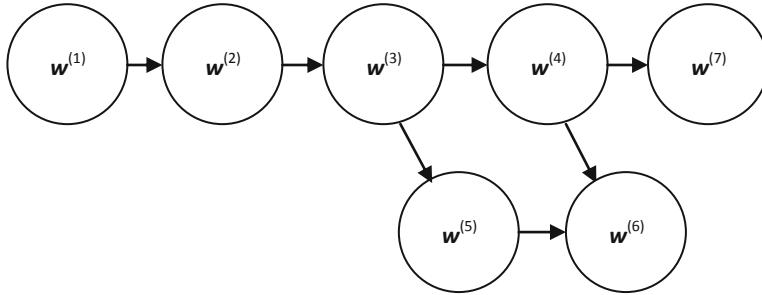


Fig. 3. Example of a reference set represented by its connectivity graph

$$w^{(1)} \prec w^{(2)} \prec w^{(3)} \prec w^{(4)} \prec w^{(7)}, w^{(1)} \prec w^{(2)} \prec w^{(3)} \prec w^{(5)} \prec w^{(6)},$$

$$w^{(1)} \prec w^{(2)} \prec w^{(3)} \prec w^{(4)} \prec w^{(6)}, w^{(4)} ? w^{(5)}, w^{(5)} ? w^{(7)}, w^{(6)} ? w^{(7)},$$

It can be noticed that in the given reference set there is one minimal ($w^{(1)}$) node and two maximal ($w^{(6)}$, $w^{(7)}$) nodes. Let w be a new observation to be recognized. Then, the following steps should be performed:

1. Neglect all patterns classes such that w is neither lower than any maximal element of the corresponding reference set nor higher than any minimal element of the reference set. If all patterns have been neglected then it means that no similarity between w and the reference sets in the sense of K -space based on a given positive cone K^+ can be established. In such case the cone K^+ should be widened or the observation w should be rejected.
2. For the rest of reference sets do as follows:
 - 2.1. Going along the paths from minimal to maximal elements in the connectivity graphs of the reference sets try to find, by taking into account the distance $|w - w^*|$, the closest to w vectors w^* such that $w^* \prec w$ or $w \prec w^*$.
 - 2.2. Having the closest to w vectors found continue the *PR* procedure according to the κ -MSO rule described in Sec. 1.

Let us remark that in the above-described case (point 2.1) a full review of the paths is in most cases unnecessary, because if a w^* has been found such that $w^* \prec w$ then for any other w^{**} such that $w^* \prec w^{**}$ it will be $|w - w^*| < |w - w^{**}|$. Therefore, in this case testing distance between w and the rest of nodes on the given path is not necessary. Moreover, if it has been found for a given w it exists in the graph a node

w^* such that even $w^* \prec w$ or $w^* \succ w$ then w can be included into the connectivity graph as its additional node without destroying its weak compactness. This means that the reference sets of patterns can easily be extended by inclusion new-recognized observations.

5 Conclusions

PR methods using reference sets representing the patterns by their correctly recognized observations are based on the concept of similarity. In the case of similarity of observation vectors interpretation as being distanced by positive or negative vectors in a K -space with adequately chosen positive cone K^+ extension of the reference sets of patterns by inclusion of new-recognized observations is easier than in other *PR* methods, based on distance measure or on angular distance. Despite this interesting property of the *PR* method based on concept of semi-ordering of vectors in the K -space lot of problems still should be explained. The method has been preliminarily tested with encouraging results on selected examples of discrimination of biomedical textures [7]. However, the experiments are still continued and their results will be published in a forthcoming paper.

Acknowledgements

This work was partially supported by the Ministry of Science and Higher Education of Poland, Project nr N N518 4211 33.

References

1. Akilov, G.P., Kukateladze, S.S.: Ordered vector spaces (in Russian) Nauka, Novosibirsk (1978)
2. Devijver, P.A., Kittler, J.: Pattern recognition: a statistical approach. Prentice Hall, London (1982)
3. Duda, O.R., Hart, P.E., Stork, D.G.: Pattern classification. John Wiley & Sons, New York (1986)
4. Fu, K.S.: Syntactic methods in pattern recognition. Acad. Press, New York (1974)
5. Kantorovich, L.V., Vulich, B.Z., Pinsker, A.G.: Functional analysis in semi-ordered spaces (in Russian). GITTL, Moscow (1959)
6. Kulikowski, J.L.: Pattern Recognition Based on Ambiguous Indications of Experts. In: Kurzyński, M. (ed.) Komputerowe Systemy Rozpoznawania KOSYR 2001, pp. 15–22. Wyd. Politechniki Wrocławskiej, Wrocław (2001)
7. Przytulska, M., Kulikowski, J.L., et al.: Technical Report Nr N N518 4211 33. Methods of computer analysis of radiological images aimed at pathological lesions assessment in selected inner body organs (in Polish). IBBE PAS, Warsaw (1910)
8. Vapnik, V.N.: Statistical learning theory. John Wiley & Sons, New York (1998)
9. Van der Waerden, B.L.: Algebra I. Springer, Heidelberg (1971)

Quo Vadis Hybrid Intelligent Systems: New Trends and Approaches

Ajith Abracham

Norwegian University of Science and Technology,
Trondheim, Norway
ajith.abraham@q2s.ntnu.no

Abstract. We are blessed with the sophisticated technological artifacts and the real world problems are getting more complicated. The emerging need for advanced Hybrid Intelligent Systems (HIS) is currently motivating important research and development work. The integration of different learning/adaptation and optimization techniques, to overcome individual limitations and achieve synergetic effects through hybridization of several intelligent algorithms, has in recent years contributed to a large number of new intelligent systems architectures. This talk presents a quick review of some of the generic hybrid architectures, which have evolved in the HIS community. We further attempt to discuss the importance of these architectures with an emphasis on the best practices for selection and combination of intelligent methods. Several application examples will be presented to demonstrate how such systems could be used for solving real world problems.

Reflections on Concepts of Employment for Modern Information Fusion and Artificial Intelligence Technologies: Situation Management, Decision Making under Varying Uncertainty and Ambiguity, Sequential Decision-Making, Learning, Prediction, and Trust

James Llinas

Research Professor (Emeritus), Executive Director
Center for Multisource Information Fusion
State University of New York at Buffalo
Buffalo, New York, USA
llinas@buffalo.edu

Abstract. Information Fusion (IF) is fundamentally an estimation process that attempts to automatically form a best approximated state of an unknown true world situation, typically from both observational and contextual data. To the degree possible, this process and its algorithms and methods also employ any deductive knowledge that model the evolutionary dynamics of the focal elements of interest in this world. Artificial Intelligence (AI) technologies are often directed to similar goals, and employ similar informational and knowledge components. For many modern problems of interest, there are factors that result in observational data whose quality is unknown, and for which the a priori deductive knowledge models are non-existent or weak. Moreover, even for conventional IF applications where uncertainties of interest are known or estimable, the Concepts of Employment that involve sequential estimation and decision-making dynamics have not been very well studied and integrated into the frameworks of understanding for the use of such IF capability. This talk will review a number of interrelated topics that bear on the thinking of how IF technologies will be used in these stressful and critical environments. It will review a previously-proposed overarching Situation Management process model, the modern (and controversial) literature on decision-making under severe uncertainty, aspects and implications of sequential operations on decision-making, as well as Learning and Prediction dynamics as they bear on IF applications. Some remarks will also be included on the dynamics of human trust in automated systems, a topic under current study at the Center for Multisource Information Fusion at Buffalo.

A Genetic Algorithm Applied to a Main Sequence Stellar Model

Gabriela de Oliveira Penna Tavares and Marco Aurelio Cavalcanti Pacheco

ICA – Pontifical Catholic University of Rio de Janeiro, Brazil
`{gtavares,marco}@ele.puc-rio.br`

Abstract. The purpose of this work is to determine some structural properties of main sequence stars through the utilization of a genetic algorithm (GA) and observational data. Genetic algorithms (GAs) are a Computational Intelligence technique inspired by Charles Darwin's theory of evolution, used to optimize the solution of problems for which there are many variables, complex mathematical modeling or a large search space. Here, we apply this technique to approximate certain aspects of stellar structure that cannot be determined through observation: the mass, radius, core density, core temperature and core pressure of a main sequence star. In order to achieve this, we use an observational value for the star's luminosity (energy flux released on the surface). Alternatively, an observational value for the star's surface temperature can be used. A mathematical model for the star's internal structure is needed to evaluate the adequacy of each solution proposed by the algorithm.

Keywords: Computational Intelligence, genetic algorithm, main sequence, star structure, stellar model.

1 Introduction

A star is a celestial body that is bound by its own gravity and that releases energy through thermonuclear processes. Because of the ongoing nuclear reactions happening in its interior, a star is constantly changing its chemical composition and structure. A simple stellar model is a simplification of the stellar structure that considers a star to be a static system. This means that the changes in composition are not taken under consideration and that variations in structural parameters are only considered along the radius of the star. Some of the parameters used to define the structure of a star are its total mass, radius, luminosity, core and surface temperature, core pressure and density. These parameters provide us with enough information about the star to model its behavior and its evolution. Other parameters include the star's chemical composition, which will not be studied in this work.

The product of this work is a hybrid intelligent system consisting of an evolutionary computation algorithm combined with a traditional mathematical model. The use of a GA in the context of stellar modeling is intended to facilitate and accelerate the convergence of solutions where the search space is intrinsically characterized by extremely large and also extremely small numbers (the mass of a star, for instance, is in the order of 10 to the power of 30, while a constant used to determine the nuclear

rate released per mass unit is in the order of 10 to the power of -15). The stochastic nature of GAs ensures that a population's subsequent state is always determined by both its previous state and by a random element, which can be very efficient in covering large search spaces.

Another difficulty in finding solutions for problems in Astronomy is that the equations involved are usually quite complex, with many variables and restrictions. GAs have been proven efficient in solving this type of problem through avoiding convergence to local optima in complex functions; the combination of mutation and crossover used in GAs has the ability to move the population away from local optima where a traditional hill climbing algorithm might get stuck in.

2 Genetic Algorithm

A genetic algorithm (GA) is a type of stochastic method that uses the concept of evolution to optimize the solution for a given problem. Each set of potential solutions for the problem is represented by a group of individuals or population. Each individual of a population is called a chromosome and contains genes that represent the characteristics or values to be optimized. Chromosomes can be encoded as binary strings, real number strings, or other type of encoding. The initial population can be obtained through generating random numbers for each gene; this population will evolve in such a way that every new generation provides better results in solving the original problem.

The steps of a GA cycle are: creation of the original population, fitness evaluation, reproduction among the fittest individuals, crossover and mutation and creation of the next population. Crossover is the exchange of genetic material among different individuals, while mutation is a random modification applied to the genetic material of an individual. Crossover and mutation are called operators of the GA.

Some techniques used to improve the convergence of the GA are: normalization, elitism and steady state. Linear normalization is used on the population's fitness evaluation and increases selection pressure on the best individuals. Elitism is a technique that keeps the best individual of a generation onto the next generation, in order

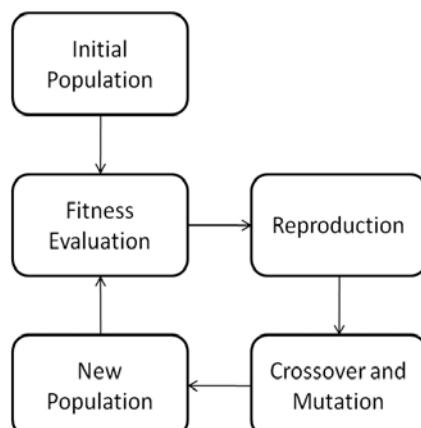


Fig. 1. A graph showing the five main steps of a genetic algorithm

to guarantee that the best individual in the new generation is at least as good as the one obtained in the previous one. Steady state is a reproduction technique that involves only partial substitution of the individuals of a population, keeping the fittest ones with the same purpose as elitism.

The rates for the application of each operator are parameters of the algorithm and must be adjusted in order to provide the best results. Fitness evaluation is a test that evaluates how close to the optimal solution an individual is. This test is actually an evaluation function that varies according to the problem. The fittest individuals will get better chances at reproducing and thus generating the individuals that will constitute the next generation. This can be simulated through a lottery where the fittest individuals have a higher weight associated to them, so that they get a better chance of being picked for reproduction. By doing this, we can guarantee that each generation usually has fitter individuals than the one that preceded it. The algorithm terminates when a certain maximum number of individuals has been evaluated or when a satisfactory level of fitness has been reached for the population.

In this work, the chromosome used in the GA is formed by six different genes represented by real numbers: m_{\max} (total mass), r_{\max} (total radius), k (opacity coefficient), u (variable related to the mean atomic mass and the average number of free electrons per nucleon), q_0 and n (auxiliary variables used in the calculation of the nuclear rate released per mass unit). The evaluation function is a measure of the error obtained when applying these values to the stellar model equations and comparing the results to the star's observational luminosity, which is the energy flux released at the surface. Two different values are obtained for the luminosity through the equations; the total error e can be calculated as the average of the errors found for the two luminosity values, as follows:

$$e = \frac{\sqrt{(F_1 - L)^2} + \sqrt{(F_2 - L)^2}}{2} . \quad (1)$$

where L is the observational luminosity and F_1 and F_2 are the luminosity values obtained through the equations.

Other parameters for the GA, such as crossover, mutation and steady state rates have been adjusted during tests and the results shall be analyzed in the case study.

3 Stellar Model

A main sequence star is a dwarf star that has reached a stage of hydrogen fusion in its nucleus. Stars spend most of their lives in this stage, transforming hydrogen into helium through the process of thermonuclear fusion. Nearly 90% of the stars we see from Earth are main sequence stars, including our Sun. This work uses a stellar model for which we assume the star's composition to be uniform throughout its body. This model can be used for any star for which the composition can be considered uniform, but here we focus on the results for main sequence stars, since they allow further simplifications: radiation pressure is considered negligible and opacity is considered constant.

The original stellar model used is a set of four ordinary differential equations and one non-differential equation, as follows:

$$\frac{dm}{dr} = 4\pi r^2 \rho ; \quad (2)$$

$$\frac{dP}{dr} = \frac{\rho G m}{r^2} ; \quad (3)$$

$$\frac{dF}{dr} = 4\pi r^2 \rho^2 q_0 T^n ; \quad (4)$$

$$\frac{dT}{dr} = \frac{-3}{4ac} \frac{k\rho}{T^3} \frac{F}{4\pi r^2} ; \quad (5)$$

$$P = \frac{R\rho T}{u} . \quad (6)$$

where the first is the continuity equation, the second is the hydrostatic equilibrium equation, the third is the thermal equilibrium equation, the fourth is the radiative transfer equation and the fifth is the equation for the total pressure of a gas. In the equations above, m represents the mass, ρ is the density, P is the pressure, T is the temperature and F is the energy flux, all of which vary along the radius of the star. The constants used are the gravitational constant (G), the constant for an ideal gas (R), and the radiation constant (a).

Solving the differential equations above would require a numerical method such as the Newton-Raphson method or the Runge-Kutta method. However, to avoid dealing with numerical solutions, we can extract dimensionless differential equations from the original ones, keeping only the relation between the parameters of interest. In order to do this, we first define a dimensionless variable x as the fractional radius of the star and define m , ρ , P , T and F as functions of x :

$$x = \frac{r}{r_{max}} ; \quad (7)$$

$$m = f_1(x)m_{max} ; \quad (8)$$

$$P = f_2(x)P_{max} ; \quad (9)$$

$$\rho = f_3(x)\rho_{max} ; \quad (10)$$

$$T = f_4(x)T_{max} ; \quad (11)$$

$$F = f_5(x)F_{max} . \quad (12)$$

where the subscript “max” indicates the maximum value for that variable. The maximum radius, mass and energy flux occur at the surface, while the maximum pressure, density and temperature occur at the core.

Now, the stellar structure equations can be rewritten and separated into two set of equations:

$$\frac{df_1}{dx} = 4\pi x^2 f_3 ; \quad (13)$$

$$\frac{df_2}{dx} = \frac{-f_1 f_3}{x^2} ; \quad (14)$$

$$f_3 = \frac{f_2}{f_4} ; \quad (15)$$

$$\frac{df_4}{dx} = \frac{-3}{4} \frac{f_3}{f_4^3} \frac{f_5}{4\pi x^2} ; \quad (16)$$

$$\frac{df_5}{dx} = 4\pi x^2 f_3^2 f_4^n ; \quad (17)$$

$$\rho_{max} = \frac{m_{max}}{r_{max}^3} ; \quad (18)$$

$$P_{max} = \frac{Gm_{max}^2}{r_{max}^4}; \quad (19)$$

$$T_{max} = \frac{uP_{max}}{R\rho_{max}}; \quad (20)$$

$$F_{max} = \frac{ac}{k} \frac{T_{max}^4 r_{max}^4}{m_{max}}; \quad (21)$$

$$F_{max} = q_0 \rho_{max} T_{max}^n m_{max}. \quad (22)$$

In each proposed solution, the GA determines values for the following variables for a given star: total mass, total radius, k (opacity coefficient), u (variable that accounts for ion and electron pressure), and two auxiliary variables, q_0 and n, used in the calculation of the nuclear rate released per mass unit. These last four variables are commonly approximated through extensive calculations. For the purpose of this work, they will be considered constant throughout the body of the star, even though that is not the case in reality. This is an approximation that allows us to create a simple model for star structure and avoid long calculations while still obtaining valid results.

We can use equations 18 to 22 in the GA's fitness evaluation. Once the algorithm determines an individual, its values will replace the variables in these equations, solving the whole system. Then, the two different values obtained for F can be compared to the observational luminosity through the error equation previously stated.

4 Case Study

The genetic algorithm was implemented in C# language with the use of GACOM, a component library for genetic algorithms developed at the ICA lab at PUC-Rio. Tests were then performed for different main sequence stars. Here we analyze the results obtained for the Sun and for Sirius (component A). These choices are based on the stars' proximity to Earth, which allows for more reliable data on their structural properties. This data will be used to measure the error obtained in the algorithm's results.

Below are the parameters used in all experiments performed. These parameters were determined empirically for obtaining best results. It is important to observe that an individual with fitness zero is an ideal solution for the problem being studied, therefore the evolution process is interrupted when the fitness for the best individual in a population converges to zero.

Parameters common to all experiments:

Population size – 100

Steady State rate – 0.5

Mutation rate – varying from 0.2 to 0.8, with a linear increase rate

Crossover rate – varying from 0.7 to 0.2, with a linear decrease rate

4.1 The Sun (Test I)

Number of experiments – 15

Number of generations in each experiment – 1,000

Total of individuals evaluated in each experiment – 100,000

Values obtained:

Luminosity – 3.846e+26 W

Mass – 1.9757e+30 kg
 Core temperature – 2.2343e+6 K
 Core density – 24.3557 kg/m³
 Radius – 43.289e+8 m
 Core pressure – 7.4173e+11 Pa

Error obtained in comparison to actual data:
 Luminosity - 0%
 Mass – 0.6%
 Core temperature – 85.7%
 Core density – 99.9%
 Radius - 521%
 Core pressure – (no actual data found)

The algorithm managed to obtain error zero in regard to the luminosity, meaning the population reached the best possible fitness. However, since the luminosity was the only piece of observational data used as a target value for the fitness function, the error increased as it spread through the system of equations. This is probably due to the existence of multiple solutions to the system of equations when the only requirement is that the luminosity matches the actual value. The mass value obtained through the GA is a good approximation, but the values for radius, density, temperature and pressure are quite unreliable.

Below are the plotted results obtained for best individual's fitness in each population, average fitness of each population and offline for each population, which is the fitness average of the best individuals found until the current population. The x axis shows the generation index and the y axis, presented in a logarithmic scale, shows the fitness values. Both the best individual and offline curves reached value zero near the end of the evolution process. The peaks shown in the population average curve are due to the random element introduced in every new population.

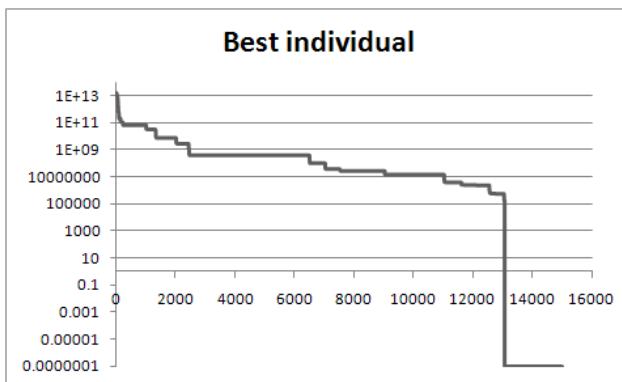


Fig. 2. Plot showing the best individual's fitness for each population in test I. The horizontal axis shows the generation index, reaching a total of 15,000 generations in the end (15 experiments with 1,000 generations each). The vertical axis is on a logarithmic scale in order to facilitate visualization. It can be concluded from the plot that the best fitness converges to zero on the 13th experiment performed.

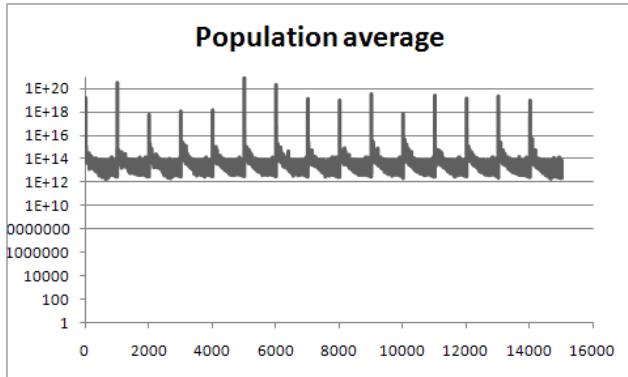


Fig. 3. Plot showing the average fitness for each population in test I. The horizontal axis shows the generation index, reaching a total of 15,000 generations in the end (15 experiments with 1,000 generations each). The vertical axis is on a logarithmic scale in order to facilitate visualization. The cyclic nature of this plot is due to high mutation rates used in the beginning of each experiment, which causes a peak in fitness with every new experiment. It can be concluded that the average fitness for the population converges near the end of each experiment, but is not improved with the use of multiple experiments.

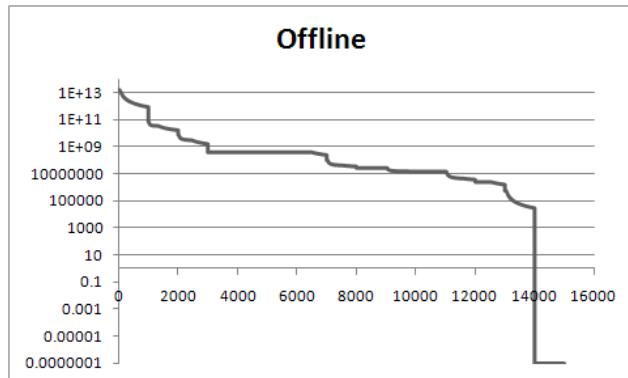


Fig. 4. Plot showing the offline for each population in test I. The offline is the fitness average of the best individuals found until the current population. The horizontal axis shows the generation index, reaching a total of 15,000 generations in the end (15 experiments with 1,000 generations each). The vertical axis is on a logarithmic scale in order to facilitate visualization. It can be concluded from the plot that the best fitness converges to zero on the 14th experiment performed.

4.2 The Sun (Test II)

In order to evaluate the algorithm's performance when another target is available for the fitness evaluation function, the core temperature value was provided as target in addition to the luminosity. Also, the following equation for the energy flux was dropped:

$$F_{max} = q_0 \rho_{max} T_{max}^n m_{max} . \quad (23)$$

This means that the chromosome had only four genes in this test: m_{\max} , r_{\max} , k and u. The modified fitness evaluation function used was:

$$e = \frac{\sqrt{(F-L)^2 + \sqrt{(T-Temp)^2}}}{2} . \quad (24)$$

where L is the observational luminosity, Temp is the target core temperature, and F and T are the values obtained for energy flux and temperature through the equations. The new results obtained are shown below.

Number of experiments – 30

Number of generations in each experiment - 100

Total of individuals evaluated in each experiment – 10,000

Values obtained:

Luminosity – 3.846e+26 W

Mass – 1.9743e+30 kg

Core temperature – 1.57e+7 K

Core density – 17,218 kg/m³

Radius – 4.8582e+8 m

Core pressure – 4.6689e+15 Pa

Error obtained in comparison to actual data:

Luminosity - 0%

Mass – 0.7%

Core temperature – 0%

Core density – 89.3%

Radius – 30.2%

Core pressure – (no actual data found)

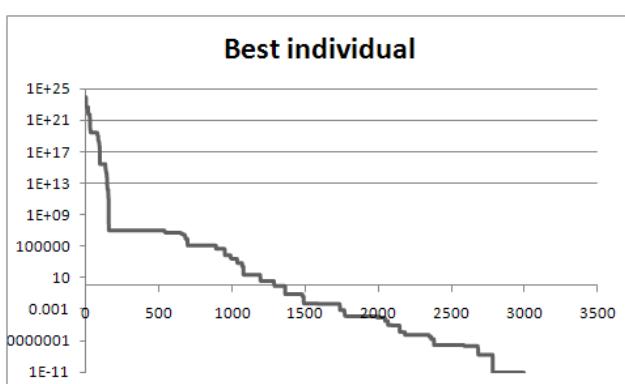


Fig. 5. Plot showing the best individual's fitness for each population in test II. The horizontal axis shows the generation index, reaching a total of 3,000 generations in the end (30 experiments with 100 generations each). The vertical axis is on a logarithmic scale in order to facilitate visualization. It can be concluded from the plot that the best fitness converges to zero on the 28th experiment performed.

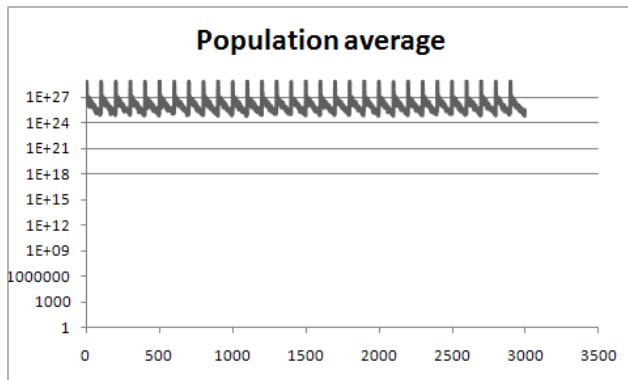


Fig. 6. Plot showing the average fitness for each population in test II. The horizontal axis shows the generation index, reaching a total of 3,000 generations in the end (30 experiments with 100 generations each). The vertical axis is on a logarithmic scale in order to facilitate visualization. The cyclic nature of this plot is due to high mutation rates used in the beginning of each experiment, which causes a peak in fitness with every new experiment. It can be concluded that the average fitness for the population converges near the end of each experiment, but is not improved with the use of multiple experiments.

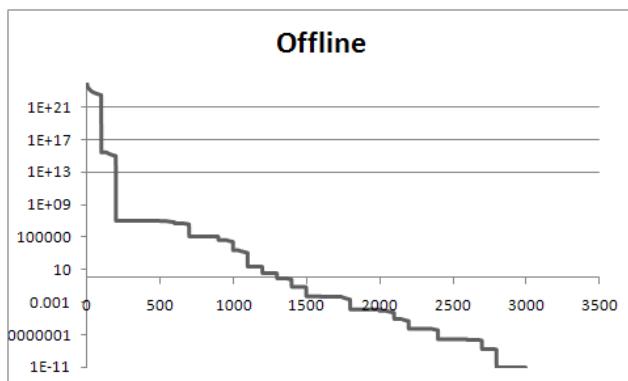


Fig. 7. Plot showing the offline for each population in test II. The offline is the fitness average of the best individuals found until the current population. The horizontal axis shows the generation index, reaching a total of 3,000 generations in the end (30 experiments with 100 generations each). The vertical axis is on a logarithmic scale in order to facilitate visualization. It can be concluded from the plot that the best fitness converges to zero on the 28th experiment performed.

The error obtained for luminosity and core temperature was zero, meaning that the GA found a solution with fitness zero for the equations. The error for total mass value was only slightly bigger, while the errors for radius and core density were considerably smaller. Even though there is no actual data for the core pressure to be used in comparison, the result obtained in this second experiment seems more reasonable.

4.3 Sirius, Component A

Sirius' component A is the brightest star seen in the night sky. It is a main sequence star with about twice the mass of the Sun and much greater luminosity.

Number of experiments – 30

Number of generations in each experiment - 100

Total of individuals evaluated in each experiment – 10,000

Values obtained:

Luminosity – 9.7688e+23 W

Mass – 6.1628e+30 kg

Core temperature – 1.8e+7 K

Core density – 1540 kg/m³

Radius – 1.587e+9 m

Core pressure – 3.9919e+14 Pa

Error obtained in comparison to actual data:

Luminosity - 0%

Mass – 53.3%

Radius – 33.2%

Core density - (no actual data found)

Core temperature - 0%

Core pressure - (no actual data found)

The configuration used for this experiment was the same as the one used in the last one, with only one equation for the energy flux and both luminosity and surface temperature used as targets in the fitness function. However, since no data was found for the star's core temperature, the value used was an approximation.

5 Conclusions

It can be concluded from the results shown above that even though the algorithm has been proven very efficient, the results obtained for the stellar structure still have room for improvement. The use of another piece of observational data, such as the value for the star's surface temperature, could considerably decrease the error in the results. The surface temperature of a star can be determined from the star's absorption spectrum. If the core temperature could be inferred from the surface temperature, then we would have one more target to be used in the GA's fitness function.

Some of the methods suggested for obtaining the star's core temperature from its surface temperature are: solving the differential equations in the stellar model or using a MultiLayer Perceptron neural network, thus creating an evolutionary computation-neural network hybrid intelligent system. The first would require a numerical method and the results would be equations for the mass, density, temperature, pressure and energy flux profiles of the star. This means that we would be able to obtain the value for any of these parameters at any distance from the center of the star, as long as the maximum value was known. The neural network, on the other hand, would require a large database of stars' characteristics, including their core temperature, so that the network could learn how to infer this value for other stars.

Another approach that could potentially improve the results obtained by the GA is to restrict the values obtained for core density and core pressure. This could be achieved by setting a minimum and a maximum value for these variables, and changing the algorithm so that any solutions that didn't meet these requirements would be discarded. This way, the GA would be able to avoid using solutions for the equations that didn't make sense in the star modeling context.

In conclusion, the GA has provided some interesting results in finding the parameters for the stellar model used. When the GA is paired with more available observational data and other techniques, good approximations can be obtained for the star's structural properties and also for coefficients such as k , u , q_0 and n , which are usually approximated through experiments.

6 Related Work

Genetic algorithms are used as an optimization tool in many different research areas, such as Biology, Economics, Petroleum Engineering, Electronics, among many others. They have also been applied to problems in Astrophysics and Astronomy. For instance, Paul Charbonneau developed a study of galaxies' rotation curves in the paper "Genetic Algorithms in Astronomy and Astrophysics" [7]. The paper "Spectral analysis of early-type stars using a genetic algorithm based fitting method" [8], by M. R. Mokiem et al, is another example.

The paper "Stellar Structure Modeling using a Parallel Genetic Algorithm for Objective Global Optimization" [9], by Travis S. Metcalfe and Paul Charbonneau, uses a GA in the context of white dwarf asteroseismology. They use a parallel implementation of a GA to model the star structure of white dwarf stars, using as observational data their oscillation frequency.

References

- 1 Priplnik, D.: An Introduction to the Theory of Stellar Structure and Evolution. Cambridge University Press, Cambridge (2010)
- 2 Phillips, A.C.: The Physics of Stars. Wiley, Chichester (2010)
- 3 Tayler, R.J.: Stars: their Structure and Evolution. Cambridge University Press, Cambridge (1994)
- 4 Freeman, R.A., Kauffman, W.J.: Universe. W.H. Freeman, New York (2008)
- 5 Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)
- 6 Corchado, E., Abraham, A., de Carvalho, A.C.: Hybrid Intelligent Algorithms and Applications. Information Science (2010)
- 7 Charbonneau, P.: Genetic Algorithms in Astronomy and Astrophysics. ApJS 101, 309 (1995)
- 8 Mokiem, M.R., De Koter, A., Puls, J., Herrero, A., Najarro, F., Villamariz, M.R.: Spectral analysis of early-type stars using a genetic algorithm based fitting method. A&A 441(2), 711–733 (2005)
- 9 Metcalfe, T.S., Charbonneau, P.: Stellar Structure Modeling using a Parallel Genetic Algorithm for Objective Global Optimization. JCP 185(1), 176–193 (2003)

Using Artificial Intelligence Techniques for Strategy Generation in the *Commons Game*

Petro Verkhogliad¹ and B. John Oommen²

¹ School of Computer Science, Carleton University, Ottawa, Canada
pverkhog@scs.carleton.ca

² Chancellor's Professor, Fellow: IEEE and Fellow: IAPR. School of Computer Science, Carleton University, Ottawa, Canada. Also Adjunct Professor with the University of Agder in Grimstad, Norway
oommen@scs.carleton.ca

Abstract. In this paper, we consider the use of artificial intelligence techniques to aid in discovery of winning strategies for the *Commons Game* (CG).

The game represents a common scenario in which multiple parties share the use of a self-replenishing resource. The resource deteriorates quickly if used indiscriminately. If used responsibly, however, the resource thrives. We consider the scenario one player uses hill climbing or particle swarm optimization to select the course of action, while the remaining $N - 1$ players use a fixed probability vector. We show that hill climbing and particle swarm optimization consistently generate winning strategies.

Keywords: Intelligent Game Playing, Commons Game, AI-based Resource Management.

1 Introduction

Artificial and computational intelligence (AI/CI) techniques have been used for decades to aid in the analysis and solution of games. In this work, we consider the *Commons Game* (CG) [17], which is a non-trivial N -person non-zero-sum game. We show that winning strategies (against opponents playing random strategies) for the *Commons Game* can be found through the application of either hill climbing (HC) or particle swarm optimization (PSO).

The *Commons Game* represents a social dilemma where the individually-selfish behavior of the members in the “society” leads to detrimental collective outcome. The simplest and most well known dilemma game is *Prisoner’s Dilemma* [4].

The *Commons Game* attempts to realistically model the interaction between a self-replenishing resource and a number of consumers of that resource. The interaction is modeled as an N player competition. Each player’s aim is to maximize the number of points they earn.

At every turn of the game, a player can take one of five actions: use the resource responsibly, use the resource indiscriminately, abstain from resource use, punish other players, reward other players.

As can be seen from the above description, the *Commons Game* can be used to model almost any resource-sharing scenario. Examples range from networking resource sharing to natural resource extraction via automated agents. This broad applicability is the one of the primary motivators for this study.

Although no results on strategy discovery *via* AI/CI methods have been published, the CG has received some attention from the research community. In [5], Baba presented two games that bear significant similarity to the game studied here. Kirts *et al.* [13] have conducted simulations, in a similar game, in order to increase awareness of the complexity associated with the decision-making and cooperation phases. In [7,6,11] Handa and Baba have shown that CG can be made more interesting for human players by altering the rules of the game to those discovered by a combination of genetic algorithms and neural networks. Brunovsky [8], on the other hand, has characterized the equilibria in the original “Tragedy of the Commons” setting. Furthermore, Faysse [10] has presented another variation of the commons dilemma game, as well as results of experiments conducted in common-pool resource sharing settings.

2 Description of the Commons Game

The *Commons Game* designed by Powers *et al.* [17,16] can be played by groups of 6 to 50 players. All of the actions in the game are mapped onto five “playing cards”: green, red, yellow, black and orange. The description of the cards and their respective point allocations¹ are given below.

The green card symbolizes indiscriminate use of the commons. Playing it yields the maximum reward, R_g , unless another player uses the black (punishment) card in the same turn, in which case it yields -20 points.

The red card represents careful or cooperative use of the commons. The reward for red card plays, R_r , depends on the total number of red cards played in the current turn, N_r . Red card players also benefit when others play the orange card. Any player using the red card cannot be penalized during the same turn.

The yellow card represents abstention from the use of the commons. Each yellow card play receives 6 points regardless of the state of the environment or the number of players in the game.

The black card can be used to punish players who abuse the environment by playing green. It yields $-N_p/N_b$, where N_b is the number of black cards played in that round, and N_p is the number of players.

The orange card can be used to encourage players who use the resource responsibly. It grants +10 points to the red card players. The orange card yields a score of $-N_p/N_o$, where N_o is the number of orange cards played during the round, and N_p is the number of players.

¹ Although we consider specific numeric score values as defined in the original manual [17], the principles presented here work even if one changes the values so as to preserve the main properties of the game.

The state of the environment also contributes to determining the reward received by green and red card players. The states range from -8 to +8. At the start of the game, the environment is at state 0.

The depletion and replenishment of the environment are modeled using a marker, m , such that $m \in [0, 180]$. At the end of every turn, the marker is updated using Eq. (1), where m_{t+1} is the marker value in the next turn, N_g is the number of green cards played in the current turn, S_t is the current state number, $I(S_t)$ is the replenishment value in the given state, and t is the current turn number.

$$m_{t+1} = \begin{cases} m_t - N_g + I(S_t) & \text{if } t \bmod 6 = 0 \\ m_t - N_g & \text{if } t \bmod 6 \neq 0 \end{cases} \quad (1)$$

The value of the marker is used to determine the state of the environment in the next turn as shown in Eq. (2):

$$S_{t+1} = \begin{cases} 0 & \text{if } 80 \leq m_t \leq 100 \\ \frac{m_t - 90}{10} & \text{if } m_t < 80 \text{ or } m_t > 100 \end{cases} \quad (2)$$

3 Methods

The focus of our work is to find winning strategy vectors consisting of the probabilities for playing a given card, $P = \{p_{green}, p_{red}, p_{yellow}, p_{orange}, p_{black}\}$, such that $\sum_{p_i \in P} p_i = 1$. We propose to do this by using hill climbing (HC) and particle swarm optimization (PSO).

3.1 Hill Climbing

Hill climbing (HC) is one of the earliest and most well-known optimization techniques. In general, HC starts with either a man-made or a random solution and attempts to find the optimum by sampling the area around it, and choosing the “best” value as determined by a heuristic function. Simple HC selects the first solution that is better than the current one. Steepest ascent/descent HC compares all neighbouring solutions and selects the best one. Finally, stochastic HC selects a random neighbour, provided its heuristic value is greater than the preset threshold.

Due to the complexity of the CG, we have opted to use a variation of the classical HC, inspired by approaches in reinforcement learning [18] and learning automata [15]. Learning automata algorithms attempt to find the best action (from a predefined set of actions) in a stochastic environment by using an action probability vector. The optimal action is found by updating the action probability vector using a well defined updating scheme.²

² There is a variety of action probability updating schemes. For lack of space they are not discussed here.

There are some key differences, however, between the version of HC used in this work and learning automata algorithms. One, at every time step learning automata select and use the action with the highest probability of reward. Two, only some of the learning automata employ a testing step and only as part of the initialization routine. The HC algorithm used here does not select the action which currently has the highest probability of reward. Rather, by generating candidate solutions it attempts to find an action with highest future reward. As such this version of the HC algorithm is closer to the algorithms described in [23] than classical HC.

At the start of the game the HC is initialized with the following parameters:

\mathcal{P} : A random solution

λ : The learning rate parameter

T : The number of turns used for learning

$f(P_t)$: A fitness function, where P_t is the vector of card probabilities at turn t .

Once initialized, the HC module is added to a game with $N - 1$ other players which employ a random action-selection strategy. In our case $N = 8$. At each turn, the HC player updates its probability vector, P , in an attempt to increase its overall reward. Five candidate vectors are created (one for each color) by choosing a card, p_i , and increasing its probability while decreasing the probabilities of the other cards. While Eq. (3) shows the updating function of the non-target cards, Eq. (4) specifies the updating function of the target card, *after* the others have been updated. In these equations, $p_i(t)$ is the current probability of selecting a card i , $p_i(t + 1)$ is the probability value that will be used in the next turn, and $i, j \in \{\text{green, red, yellow, orange, black}\}$ such that $i \neq j$.

$$p_i(t + 1) = \lambda p_i(t) . \quad (3)$$

$$p_j(t + 1) = p_j(t) + (1 - \sum_{i \in P} p_i(t + 1)) . \quad (4)$$

Each of the five candidate vectors are then tested by playing T turns of the game. The vector which returns the highest fitness value, i.e., the sum of scores over the T turns, is selected as the current best solution. The process is then repeated for K iterations.

The strategies discovered by this HC scheme are discussed in the following sections.

3.2 Particle Swarm Optimization

Particle swarm optimization (PSO) was originally developed by Kennedy and Eberhart [12]. The algorithm is based upon the flocking behavior of fish/birds, and has been successfully applied to many optimization and gaming problems [10, 9, 14]. PSO relies on a set of n -dimensional particles traveling through the solution space. Each particle's position represents a possible solution. Particles track their own best solution, velocity, and the global best solution.

Similar to HC, each particle uses a set of updating equations to direct itself and the rest of the swarm toward the optimum. The updating equation for a particle's position, \vec{x}_i , is shown in Eq. (5), where \vec{v}_i is the particle's velocity. The updating equation for the latter, is shown in Eq. (6), where g and \hat{g} are, respectively, the particle's and the swarm's “best” solutions.

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t) . \quad (5)$$

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_1 j(t)[g_{ij}(t) - x_{ij}(t)] + c_2 r_2 j(t)[\hat{g}_{ij}(t) - x_{ij}(t)] . \quad (6)$$

In the above equations, x_{ij} and v_{ij} are the j^{th} components of $\vec{x}_i(t)$ and $\vec{v}_i(t)$ respectively. Further, c_1 and c_2 are the acceleration coefficients, and ω is the inertia weight. Finally, r_1 and $r_2 \in U(0, 1)^n$. It has been shown that under the following conditions:

$$\omega > \frac{1}{2}(c_1 + c_2) - 1; \text{ and } 0 < \omega < 1 . \quad (7)$$

convergence to a stable equilibrium point is guaranteed.

The structure of the particle neighbourhood used in our work is commonly referred to as *gbest*. This configuration was intentionally selected as this is the first time it has been used in relation to CG. More complex approaches, lattice or Von Neumann, are reserved for future work.

Each particle in the swarm is initialized with a random position vector which represents a game strategy vector. The PSO player is then added to a game with $N - 1$ other players which employ a random action-selection strategy. In our case $N = 8$. At every game turn, the fitness of every particle is evaluated by using its position to play T turns of a training instance of the game. The fitness value is the sum of the rewards received over the T turns. Similar to the HC approach, the fitness function used does not incorporate any information about the game, except for player's score over the T turns. If the new fitness value is higher than any previously seen value, the new strategy becomes the particle's “best” solution. The individual “best” solutions are then compared to determine the global “best” solution. This continues for the rest of the game.

4 Results and Discussion

In order to explore the set of possible strategies for the *Commons Game*, experiments using both, HC and PSO, were performed in the following list of state ranges, $\mathcal{S} = \{[+8, -8], [+8, +8], [+8, +4], [+4, 0], [0, -4], [-4, -8]\}$. 100 games were played for every state range S_i , with games of 1,000 turns. As previously mentioned, only one “intelligent” player was used in each game. The remaining $N - 1$ players used a random action-selection strategy. Tables 1³, 3 and 2 show the scores and strategies generated by the HC and PSO players respectively⁴.

³ In this table the subscripts *as* and *sd* are “average score” and “standard deviation”, respectively.

⁴ Average scores shown in the tables were calculated as the sum of scores for 100 games divided by 100.

Table 1. Average score per player type

State Range	Random _{as}	Random _{sd}	HC _{as}	HC _{sd}	PSO _{sd}	PSO _{sd}
+8, +8	25,555	2,043.9	59,566	1,033.36	78,950	1,614.67
+8, +4	23,350	1,679.4	66,809	1,192.01	69,460	778.4
+4, 0	11,570	854.1	35,793	492.4	37,287	563.4
0, -4	-1,382	439.1	6,103	38.2	5,937	36.6
-4, -8	-3,129	193.6	5,993	2.26	5,951	32.4

Table 2. Summary of Hill Climbing Strategies

State Range	Red	Green	Yellow	Orange	Black
+8, -8	0.0000	0.0000	0.9998	0.0001	0.0000
+8, +8	0.6528	0.2089	0.0529	0.0325	0.0526
+8, +4	0.9982	0.0004	0.0004	0.0004	0.0004
+4, 0	0.9029	0.0083	0.0031	0.0120	0.0734
0, -4	0.5059	0.0000	0.4940	0.0000	0.0000
-4, -8	0.0001	0.0000	0.9994	0.0000	0.0000

Table 3. Summary of PSO Strategies

State Range	Red	Green	Yellow	Orange	Black
+8, -8	0.0017	0.0014	0.9958	0.0008	0.0001
+8, +8	0.8179	0.1604	0.0100	0.0062	0.0053
+8, +4	0.9891	0.0071	0.0015	0.0009	0.0012
+4, 0	0.9314	0.0300	0.0303	0.0004	0.007
0, -4	0.2026	0.0060	0.7903	0.0008	0.0000
-4, -8	0.0018	0.0016	0.9955	0.0008	0.0001

[+8, +8] State Range

In this state, red and green players reap the maximum possible rewards. Both HC and PSO allocate most of the probability into the red and green cards. HC is more aggressive. However, this resource overuse results in frequent punishment. PSO prefers red card plays, which can not be punished. This accounts for the difference in scores. The difference is statistically significant with $p < 0.05$.

It should be noted that the strategy discovered by the PSO player is very similar to the one considered by Powers [17] as the cooperative game solution.

[+8, +4] State Range

The “intelligent” players develop equivalent strategies. Neither player allocates any probability weight into playing green cards. Instead, almost all of the probability weight is placed into red card plays. The difference of payoffs

for this range in comparison to the $[+8, +8]$ range accounts for the different probability allocation.

[+4, 0] State Range

This range, as well as the one immediately below, are of particular interest since they include the starting state, 0. Red plays are, once again, the most dominant for both, HC and PSO, schemes. This can be attributed to the fact that the red play yields a positive reward while being nonpunishable.

Unlike in the $[+8, +4]$ state, the difference in the scores between the HC and PSO strategies is statistically significant with $p < 0.05$.

[0, -4] State Range

Here the strategies chosen by both of the players are quite different. PSO primarily abstains from resource use. HC, on the other hand, alternates between responsible use and abstention. This difference accounts for the better performance of the HC player. The difference in scores is statistically significant with $p < 0.05$.

[-4, -8] State Range

This state range represents a degraded commons. A previously unseen strategy of complete abstention arises here. It should also be noted that the strategy developed by the HC performs better with statistical significance, $p < 0.05$.

4.1 Discussion

Overall conclusions of the HC and PSO strategies are discussed below.

One way to interpret the discovered strategies is that they are self-reliant. Both algorithms play nonpunishable cards with positive rewards. The developed strategies, in fact, can be viewed those that recognize the volatility of the opponents' actions.

Also interesting is the fact that a single component of the strategy vector is preferred (when compared to all of the other possibilities) in the majority of the strategies. This suggests that in each state there is a single best response to any action by the random player. This also hints at the "existence" of possible equilibrium strategies.

Finally, we observe that the choice of the fitness function and the strategy representation play an important role in the above. Neither the HC nor the PSO players take into account the state of the game or the previous moves of their opponents. Indeed, in this configuration the players seem to discover those strategies that maximize the score independent of the behavior of the other players.

5 Conclusion

In this work we have investigated the use of artificial/computational intelligence techniques in order to generate winning strategies for an N -player, imperfect-information non-zero-sum game. More specifically, we have shown that general

purpose optimization techniques (such as hill climbing (HC) and particle swarm optimization (PSO)) can be used successfully in the confines of a complex N -person game such as the *Commons Game*.

References

1. Abdelbar, A.M., Ragab, S., Mitri, S.: Applying co-evolutionary particle swarm optimization to the egyptian board game seega. In: ASPGP 2003. pp. 9–15 (2003)
2. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13–15), 2729–2730 (2009)
3. Abraham, A., Köppen, M., Franke, K. (eds.): Design and application of hybrid intelligent systems. Amsterdam, IOS Press (2003)
4. Axelrod, R., Hamilton, W.: The evolution of cooperation. Science 211(4489), 1390 (1981)
5. Baba, N.: The commons game by microcomputer. Simulation & Gaming 15, 487–492 (1984)
6. Baba, N.: Utilization of genetic algorithms in order to make game playing much more exciting. In: KES 1999, pp. 473–476 (1999)
7. Baba, N., Nagasawa, K., Handa, H.: Utilization of Soft Computing Techniques for Making Environmental Games More Exciting –Toward an Effective Utilization of the COMMONS GAME. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 411–417. Springer, Heidelberg (2008)
8. Brunovský, P.: The commons game. Ekonomicky Casopis 55(8), 811–814 (2007)
9. Conradie, J., Engelbrecht, A.P.: Training bao game-playing agents using coevolutionary particle swarm optimization. In: CIG 2006, pp. 67–74 (2006)
10. Faysse, N.: Coping with the tragedy of the commons: Game structure and design of rules. Journal of Economic Surveys 19(2), 239–261 (2005)
11. Handa, H., Baba, N.: Evolutionary computations for designing game rules of the commons game. In: CIG 2007, pp. 334–339 (2007)
12. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: ICNN 1995, vol. 4, pp. 1942–1948 (1995)
13. Kirts, C.A., Tumeo, M.A., Sinz, J.M.: The commons game: Its instructional value when used in a natural resources management context. Simulation & Gaming 22(1), 5–18 (1991)
14. Laskari, E.C., Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization for minimax problems. In: CEC 2002, pp. 1582–1587 (2002)
15. Narendra, K.S., Thathachar, M.A.L.: Learning Automata: An Introduction. Prentice-Hall, Inc., Upper Saddle River (1989)
16. Powers, R.B., Boyle, W.: Generalization from a commons-dilemma game: The effects of a fine option, information, and communication on cooperation and defection. Simulation & Gaming 14(3), 253–274 (1983)
17. Powers, R.B., Duss, R.E., Norton, R.S.: THE COMMONS GAME Manual (1977)
18. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2004)

Evaluation of Network Survivability Considering Degree of Disconnectivity

Frank Yeong-Sung Lin, Hong-Hsu Yen, Pei-Yu Chen*, and Ya-Fang Wen

Department of Information Management, National Taiwan University

Taipei, Taiwan, R.O.C.

{yslin,d96006,r94048}@im.ntu.edu.tw,
hhyeh@cc.shu.edu.tw

Abstract. It is impossible for a system or network to keep completely safe with the possibility of many new threats occurring at any moment. To analyze and solve these kinds of problems, this paper presents a mathematical programming problem, which adopts a novel metric called Degree of Disconnectivity (DOD) to evaluate the damage level and survivability of a network. To evaluate and analyze the robustness of a network for network operators, this problem is modeled as a mathematical programming problem. Here, an attacker applies his limited attack power intelligently to the targeted network. The objective of the attacker is to compromise nodes, resulting in disconnections of O-D pairs, to ensure that the proposed degree of disconnectivity metric reaches a given level. A Lagrangean relaxation-based algorithm is adopted to solve the proposed problem.

Keywords: Degree of Disconnectivity, Lagrangean Relaxation, Network Attack, Optimization Problem, Resource Allocation.

1 Introduction

Increased reliance on the Internet has made information systems, computers and servers connected to the network more vulnerable to attacks. There are a variety of security threats on the Internet, and the number of new threats is growing rapidly [1]. The result of these threats may include system attacks or failures which may significantly reduce the capability of the communication network to efficiently deliver service to users.[1]. Therefore, the ability for a system or network to maintain a certain level of performance under the presence of security incidents is more important than to prevent a system or network from threats. This concept is called survivability, which is widely used in many networks [2, 3].

Network survivability is one of the most important issues in the planning and operation of networks and security systems, mainly when the threats are related with DoS attacks [4]. There has been a substantial amount of research related to network

* Corresponding author.

survivability solved by using mathematical programming approaches, such as an optimization problem as presented in [5] to solve network connectivity and flow interdiction problems, and, the problem of designing a network that is able to survive under intentional attacks is examined in [6]. Nevertheless, how to assess the survivability of a network under or after attacks is a continuing problem for information security researchers and experts.

2 Problem Formulation and Notations

2.1 Problem Description

Because an attacker's resources, i.e., time, money, and man power, are limited, only part of a network can be compromised. Therefore, the resources must be fully utilized so that the attacker can cause the maximum harm to the target network. Although in [7], the author proposed two extreme survivability metrics by considering the numbers of the O-D pairs that are connected or disconnected to measure the network survivability, it is too strict to comply with the real case. Based on [7], we have developed a more flexible metric for network survivability. The proposed survivability metric called the *degree of disconnectivity* (DOD) is defined as S , which assesses the average damage level of a network; it can also be called the *degree of segmentation*, *degree of segregation*, or *degree of separation*.

The DOD metric in this paper is defined as S , shown in Equation 1. S is evaluated on the disconnected numbers of O-D pairs among all O-D pairs, which can be generated as the residual index of the networks. t_{wi} is 1, while node i on an O-D pair w is dysfunctional. The transmission cost of dysfunctional node is M , otherwise it is ε . The greater the value of S , the more the network is damaged.

$$S = \frac{\sum_{w \in W} \sum_{i \in V} t_{wi} c_i}{C_2^N \times M}. \quad (1)$$

2.2 Problem Formulation

In this section, the serial of attack actions considering the survivability of a network is modeled as an optimization problem, in which the objective is to minimize the total attack cost from an attacker's perspective, such that the given critical O-D pair is disconnected and the survivability is over the given threshold resulting in the inability of the network to survive. Note that the network discussed here is at the AS level. Here, both the attacker and the defender have complete information about the targeted network topology and the budget allocation is assumed.

The above problem is formulated as a maximization mathematical model as follows. For simplicity, since the targeted network is at the AS level, the attacker cannot simply attack any node directly. The notations used in this paper and problem formulation is defined in Table 1.

Table 1. Given Parameters and Decision Variables

Given parameter		Description
Notation		
V	Index set of nodes	
W	Index set of OD pair	
P_w	Set of all candidate paths of an OD pair w , where $w \in W$	
M	Large enough number of processing cost that indicates a node has been compromised	
ε	Small enough number of processing cost that indicates a node is functional	
δ_{pi}	Indicator function, 1 if node i is on path p , 0 otherwise, where $i \in V$ and $p \in P_w$	
\hat{a}_i	The threshold of attack cost leading to a successful node attack	
S	The threshold of a network crash, which is the average damage level of all O-D pairs	
R_w	The weight of O-D pair w , where $w \in W$	
\hat{a}_i	The threshold of attack cost leading to a successful node attack	

Decision variable		Description
Notation		
x_p	1 if path p is chosen, 0 otherwise, where $p \in P_w$	
y_i	1 if node i is compromised by attacker, 0 otherwise (where $i \in V$)	
t_{wi}	1 if node i is used by O-D pair w , 0 otherwise, where $i \in V$ and $w \in W$	
c_i	Processing cost of node i , which is ε if i is functional, M if i is compromised by attacker, where $i \in V$	

The problem is then formulated as the following problem:

Objective function:

$$\min_{y_i} \sum_{i \in V} y_i \hat{a}_i , \quad (\text{IP 1})$$

Subject to:

$$c_i = y_i M + (1 - y_i) \varepsilon , \quad \forall i \in V \quad (\text{IP 1.1})$$

$$\sum_{i \in V} t_{wi} c_i \leq \sum_{i \in V} \delta_{pi} c_i , \quad \forall p \in P_w , \quad \forall w \in W \quad (\text{IP 1.2})$$

$$\sum_{p \in P_w} x_p \delta_{pi} = t_{wi} , \quad \forall i \in V , \quad \forall w \in W \quad (\text{IP 1.3})$$

$$S \leq \frac{\sum_{w \in W} \sum_{i \in V} t_{wi} c_i}{|W| \times M} \quad (\text{IP 1.4})$$

$$\sum_{p \in P_w} x_p = 1 , \quad \forall w \in W \quad (\text{IP 1.5})$$

$$x_p = 0 \text{ or } 1 , \quad \forall p \in P_w , \quad \forall w \in W \quad (\text{IP 1.6})$$

$$y_i = 0 \text{ or } 1 , \quad \forall i \in V \quad (\text{IP 1.7})$$

$$t_{wi} = 0 \text{ or } 1 , \quad \forall i \in V , \quad \forall w \in W . \quad (\text{IP 1.8})$$

The objective of the formulation is to minimize the total attack cost by of the attacker by deciding which node to compromise. Constraint (IP 1.1) describes the definition of the transmission cost of node i , which is ϵ if node i is functional, and M if node i is compromised. Constraint (IP 1.2) requires that the selected path for an O-D pair w should be the minimal cost path. Constraint (IP 1.3) denotes the relationship between t_{wi} and $x_p \delta_{pi}$. To simplify the problem-solving procedure, the auxiliary set of decision variables t_{wi} is replaced by the sum of all $x_p \delta_{pi}$. (IP 1.1) to (IP 1.3) jointly require that, when a node is chosen for attack, there must be exactly one path from the attacker's initial position, s , to that node, and each node on the path must have been compromised. These constraints are jointly described as the continuity constraints. And constraint (IP 1.4) determines if a target network has been compromised, the DOD metrics must be larger than the given threshold. Constraints (IP 1.5) and (IP 1.6) jointly entail that only one of the candidate paths of an OD pair w can be selected. Last, constraints (IP 1.6) to (IP 1.8) impose binary restrictions on decision variables.

3 Solution Approach

3.1 Solution Approach for Solving the Problem of (IP 1)

3.1.1 Lagrangean Relaxation

Lagrangean Relaxation (LR) [8] is an optimization method that can be applied to linear and integer programming, combinatorial optimization, and non-linear programming [9]. In this paper, a Lagrangean relaxation-based algorithm is thus proposed, in conjunction with the subgradient method, to solve (IP 1).

By applying this method [8] with a vector of Lagrangean multipliers u^1, u^2, u^3 , and u^4 , the model into the following Lagrangean Relaxation problem (LR 1) is transformed. In this case, Constraints (1-1) to (1-4) are relaxed. To achieve better results, a Lagrangean relaxation procedure is adopted. By definition, u^1, u^2, u^3 , and u^4 are the vectors of $\{u_i^1\}, \{u_{wp}^2\}, \{u_{wi}^3\}, \{u^4\}$ respectively. (LR 1) is decomposed into three independent and easily solvable optimization subproblems with respect to decision variables x_p, y_i , and t_{wi}, c_i , and the respective subproblems can thus be optimally solved.

Subproblem 1 (related to decision variable x_p)

$$Z_{Sub1}(u^3) = \min \sum_{w \in W} \sum_{i \in V} \sum_{p \in P_w} u_{wi}^3 \delta_{pi} x_p, \quad (\text{Sub 1})$$

$$\sum_{p \in P_w} x_p = 1, \forall w \in W \quad (\text{LR 1})$$

$$x_p = 0 \text{ or } 1, \forall p \in P_w, w \in W. \quad (\text{LR 2})$$

To reduce the complexity, subproblem 1 is decomposed into $|W|$ problems, which are all independent shortest path problems. The value of x_p for each O-D pair w is individually determined. Hence, u_{wi}^3 can be viewed as the cost of node i on O-D pair w . Dijkstra's algorithm is adopted to obtain x_p for each O-D pair w . The time complexity

of Dijkstra's algorithm is $O(|V|^2)$, where $|V|$ is the number of nodes; therefore, the time complexity of subproblem 1 is $O(|W| \times |V|)$.

Subproblem 2 (related to decision variable y_i)

$$\begin{aligned} Z_{Sub2}(u^1) &= \min \sum_{i \in V} y_i \hat{a}_i(b_i) + \sum_{i \in V} u_i^1 y_i \epsilon + \sum_{i \in V} u_i^1 y_i (-M) + u_i^1 \epsilon \\ &= \min \sum_{i \in V} [\hat{a}_i(b_i) + u_i^1 \epsilon + u_i^1 (-M)] y_i + u_i^1 \epsilon, \end{aligned} \quad (\text{Sub 2})$$

$$y_i = 0 \text{ or } 1, \forall i \in V. \quad (\text{LR 3})$$

To solve subproblem 2 optimally, this problem can also be decomposed into $|V|$ individual problems. The value of decision variable y_i is determined by its coefficient, whose value is $\hat{a}_i(b_i) + u_i^1 \epsilon + u_i^1 (-M)$. In order to minimize subproblem 2, if this coefficient is positive, y_i is set as zero; otherwise it is one. The time complexity of subproblem 2 is $O(|V|)$.

Subproblem 3 (related to decision variables t_{wi}, c_i)

$$\begin{aligned} Z_{Sub3}(u^1, u^2, u^3, u^4) &= \min \sum_{i \in V} u_i^1 c_i + \sum_{w \in W} \sum_{p \in P_w} u_{wp}^2 \sum_{i \in V} t_{wi} c_i + \sum_{w \in W} \sum_{p \in P_w} u_{wp}^2 \sum_{i \in V} (-\delta_{pi} c_i) + \\ &\quad \sum_{w \in W} \sum_{i \in V} u_{wi}^3 (-t_{wi}) + u^4 (-\sum_{w \in W} R_w \sum_{i \in V} t_{wi} c_i) + u^4 S |W|M \\ &= \min \sum_{i \in V} \left\{ u_i^1 - \sum_{w \in W} \sum_{p \in P_w} u_{wp}^2 \delta_{pi} + \sum_{w \in W} \left((\sum_{p \in P_w} u_{wp}^2) - u^4 R_w \right) t_{wi} \right\} c_i - \sum_{w \in W} u_{wi}^3 t_{wi} \Bigg\} + u^4 S |W|M, \end{aligned} \quad (\text{Sub 3})$$

$$t_{wi} = 0 \text{ or } 1 \quad (\text{LR 4})$$

$$c_i = M \text{ or } \epsilon, \forall i \in V. \quad (\text{LR 5})$$

To optimally solve subproblem 3, it is further decomposed it into $|V|$ independent subproblems. However, since each decision variable t_{wi} and c_i in (LR 4) and (LR 5) have only two kinds of value, the exhaustive search is applied here to find the optimal objective function value among the four combinations of t_{wi} and c_i . The time complexity of subproblem 3 is $O(|V| \times |W|)$.

These relaxed problems are solved optimally to get a lower bound for the primal problem. After solving (LR 1), the resulting bounds are taken as the initial bounds in the next stage. Three stage heuristics are adopted to derive feasible solutions to the primal problem, and the Subgradient method is used to update the Lagrangean multipliers to obtain a better bound.

3.1.2 Getting Primal Feasible Solutions

To obtain the primal feasible solutions of (IP 1), the solutions obtained from (LR) are considered. By using the Lagrangean Relaxation method and the Subgradient method, a theoretical lower bound on the primal objective function value and ample hints for

getting primal feasible solutions are obtained. However, as some critical and difficult constraints are relaxed to obtain the (LR) problem, the solutions may not be valid for the primal problem. Thus, there is the need to develop heuristics to tune the values of the decision variables so that primal feasible solutions can be obtained. As a result, a heuristic is adopted to improve this situation. In this heuristic, each solution to (LR) is adjusted to a feasible solution to (IP 1).

The concept of this heuristic arises from the attacker's strategy. Given that the node was traversed several times, the attacker would have higher possibility of attacking it. Hence, the compromised nodes are separated in the *Attack-Bucket*, while the rest nodes are in the *Safety-Bucket*. First, select nodes from the Safety-Bucket to transfer to the Attacked-Bucket. Then adjust the nodes transferred to the Attacked-Bucket from the Safety-Bucket. Along this manner, a heuristic for getting a primal feasible solution is developed. The time complexity for this heuristics is $O(|V|)$.

4 Computational Experiments

4.1 Experiment Environment

The proposed algorithms for the DOD model are coded in Visual C++ and run on a PC with an INTEL™ Core2 CPU 6400 2.13 GHz CPU. Two types of network topology, grid and scale-free networks, as attack targets are demonstrated here. To determine which budget allocation policy is more effective under different cases, two initial budget allocation policies are designed—uniform and degree-based. The former distributes the defense budget evenly to all nodes in the network, while the latter allocates budget to each node according to the percentage of a node's degree.

4.2 Experiment Result of the Problem of (IP 1)

To compare attack behavior under different scenarios, we use the attacker's attack cost to evaluate the degree to which the attacker's objective is achieved. The greater the attack cost, the more robust of the network. As Fig. 1 shows, the robustness of grid networks of these two budget allocations, uniform and degree-based, is quite similar. Inasmuch as the property of grid networks is fair to each node within networks, the same tendency can be found with these two budget allocations.

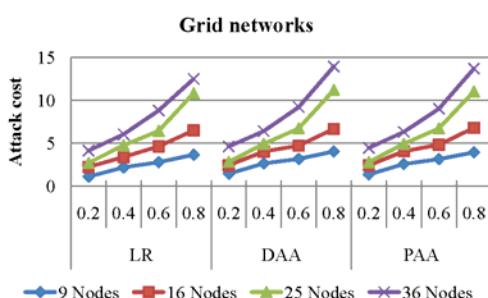


Fig. 1. Attack cost of grid networks under different budget allocations

For Fig. 2, under scale-free networks, comparing these two budget allocations, the uniform type is more vulnerable than degree-based. The regulation is an undesigned coincidence with Figure 4-4 and Figure 4-5. The uniform budget allocation, which treats each node equally, fails to reflect the discrepancy between the nodes, and results in an insecure network.

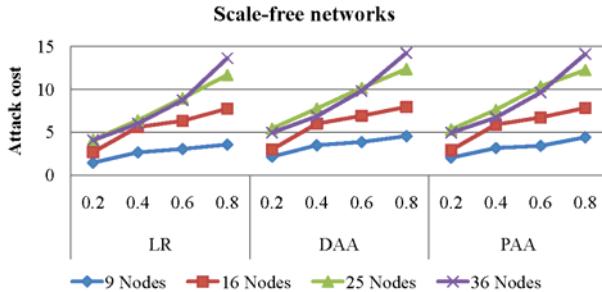


Fig. 2. Attack cost of scale-free Networks under different budget allocation

From Fig. 1 and Fig. 2, the increasing average damage level apparently gave rise to attack costs under diverse topologies and budget allocations. It is unmistakably clear that the proposed DOD metric is reflected in the residual survivable nodes of the networks. By disbursing more resources, the attackers achieve a higher average damage level to the target networks.

5 Conclusion

In this paper, we use attack and defense scenarios considering the DOD metric to describe attacker and defender behavior of networks. According to attacker's objective, the robustness of the networks is evaluated by the defender. The lesser the attack cost, the worse the survivability. The key contribution of this paper is that the attack and defense scenarios are successfully modeled as a well-formulated mathematical model, which is then optimally solved by the proposed heuristic. With this mathematical technique, we can resolve the complex problems based on the optimized methodology.

The novel network DOD metric is another contribution of this paper. The metric reflects the aim of an attacker to separate the target network into pieces. This metric enables the indication of the damage of the residual networks. Finally, we have also examined different network topologies and observed their robustness to malicious and intelligent attacks. The experiment results show that the degree-based defense budget allocation strategy is the most robust.

In this paper, we adopt a DOD metric in the computational experiments. The current research considers two attack strategies under given topology, but it would be more comprehensive if attacker behavior is more vivid. If the attacker's apparent objective is to compromise a network, he will try his best to damage the target

network, for example, by spending different budgets according to the importance of a node. This more complex attacker behavior, therefore, should be considered in further research.

Acknowledgments. This research was supported by the National Science Council of Taiwan, Republic of China, under grant NSC-99-2221-E-002-132.

References

1. Peters, S.: 2009 CSI Computer Crime and Security Survey, Computer Security Institute (December 2009)
2. Ellison, R.J., Fisher, D.A., Linger, R.C., Lipson, H.F., Longstaff, T., Mead, N.R.: Survivable Network Systems: An Emerging Discipline. Technical Report CMU/SEI-97-TR-013 (November 1997)
3. Garg, M., Smith, J.C.: Models and Algorithms for the Design of Survivable Multicommodity Flow Networks with General Failure Scenarios. *Omega* 36(6), 1057–1071 (December 2008)
4. Pinzón, C., De Paz, J.F., Zato, C., Pérez, J.: Protecting Web Services against DoS Attacks: A Case-Based Reasoning Approach. In: Graña Romay, M., Corchado, E., García Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 229–236. Springer, Heidelberg (2010)
5. Murray, A.T., Matisziw, T.C., Grubecic, T.H.: Critical Network Infrastructure Analysis: Interdiction and System Flow. *Journal of Geographical Systems* 9(2), 103–117 (June 2007)
6. Smith, J.C., Lim, C., Sudargho, F.: Survivable Network Design Under Optimal and Heuristic Interdiction Scenarios. *Journal of Global Optimization* 38(2), 181–199 (June 2007)
7. Lin, Y.S., Tsang, P.H., Chen, C.H., Tseng, C.L., Lin, Y.L.: Evaluation of Network Robustness for Given Defense Resource Allocation Strategies. In: 1st International Conference on Availability, Reliability and Security, pp. 182–189 (April 2006)
8. Fisher, M.L.: An Applications Oriented Guide to Lagrangian Relaxation. *Interfaces* 15(2), 10–21 (April 1985)

Fuzzy Control of Trade-Off between Exploration and Exploitation Properties of Evolutionary Algorithms

Adam Slowik

Department of Electronics and Computer Science, Koszalin University of Technology,
Sniadeckich 2 Street, 75-453 Koszalin, Poland
aslowik@ie.tu.koszalin.pl

Abstract. In this paper novel selection method based on hybridization of fuzzy logic and previously elaborated mix selection is presented. Proposed method is based on fuzzy control of trade-off between exploration and exploitation properties of evolutionary algorithm. Two factors are considered in proposed selection method: the first is a number of generations of evolutionary algorithm, and second is a population diversity. Due to these two factors, we control the trade-off between global and local search of solution space, and also due to the fuzzy control the proposed method is more "immune" in falling into the trap of local extremum. The results obtained using proposed method are compared with the results obtained using other selection methods like: roulette selection, elitist selection, fan selection, tournament selection, deterministic selection, truncation selection, and mix selection. The comparison is performed using test functions chosen from literature. The results obtained using proposed selection method are better in many cases than results obtained using other selection techniques.

1 Introduction

In paper [1], Back has shown that the selection processes can control the level of exploration and exploitation of solution space by varying selection pressure. Higher selection pressure pushes the search towards larger exploitation and lower selection pressure urges the search towards larger exploration [1, 2]. Therefore, one of the most important elements of genetic or evolutionary algorithm operation is a selection phase during which new populations of potential solutions (individuals) are created. The selection pressure is critical in designing a selection mechanism and has been widely studied in area of evolutionary algorithms. As an example we can mention the papers [3, 4]. Although selection has been studied for several decades, tuning of the selection pressure is still difficult, and not as easy to parameterize as other factors (i.e. population size), and remains an important open problem in evolutionary algorithm research [5, 6].

In literature, we can find many different kinds of selection techniques. Among them we can mention: roulette selection [11], elitist selection [12], deterministic

selection [13], tournament selection [14], truncation selection [15], or fan selection [16]. Many authors try to compare existing selection methods; for example in paper [17] a comparison of elitist selection [12], rank selection [19] and tournament selection [20] is presented. This comparison is performed with regard to selection intensity and selection variance.

In this paper, a fuzzy mix selection based on steering between exploration and exploitation properties of evolutionary algorithms is presented. The fuzzy mix selection is based on mix selection [7] which was elaborated earlier. In the proposed selection the exploration property and exploitation property are controlled using fuzzy control of α parameter (see section 2 and section 3). We have a hybrid artificial intelligent system [23, 24] between fuzzy logic and evolutionary algorithm. The proposed method has been tested using test functions chosen from literature. The results obtained using fuzzy mix selection are compared with results obtained using other existing selection methods.

2 Fuzzy Mix Selection

The proposed fuzzy mix selection (FMIX) method is based on mix selection method [7]. In FMIX method, the fuzzy logic is used to control of α parameter ($\alpha \in [-1; 1]$) values. Using the α parameter the values of relative fitness $r\text{fitness}$ for particular individuals are computed as follows:

- for the best individual (in the case when $\alpha \geq 0$)

$$r\text{fitness}'_{max} = r\text{fitness}_{max} + (1 - r\text{fitness}_{max}) \cdot \alpha \quad (1)$$

- for others individuals (in the case when $\alpha \geq 0$)

$$r\text{fitness}' = r\text{fitness} \cdot \left(\frac{r\text{fitness}_{max} - r\text{fitness}'_{max}}{\sum_{i=1}^M r\text{fitness}_i - r\text{fitness}_{max}} + 1 \right) \quad (2)$$

- for all individuals (in the case when $\alpha < 0$)

$$r\text{fitness}' = r\text{fitness} + \left(r\text{fitness} - \frac{1}{M} \right) \cdot \alpha \quad (3)$$

where: $r\text{fitness}'_{max}$ -new relative fitness of the best individual; $r\text{fitness}_{max}$ -old relative fitness of the best individual; α -scaling factor $\alpha \in [-1; 1]$; $r\text{fitness}'$ -new relative fitness of chosen individual; $r\text{fitness}$ -old relative fitness of chosen individual; M -number of individuals in population.

However the problem is, how the value of α parameter should be changed? In paper [7], the value of α increases linearly, and this means the exploitation property becomes stronger and stronger during running of the algorithm. But this may not work if the search is trapped in a local optimal solution (premature convergence). In this case selection pressure should be decreased, or even it is necessary to restart the algorithm, rather than to increase the selection pressure. Therefore, the α parameter value should be tuned adaptively during algorithm iterations progress. In this paper we proposed a fuzzy logic control in order to obtain adaptive changes of the α parameter value.

3 Fuzzy Logic Control of Value of α Parameter

This proposal consider two aspects describing the state of the evolutionary process: population diversity (PD) and generation percentage (GP) already performed (similarly as in paper [8]). The PD and GP are the input variables. The value of α parameter is an output value. The input linguistic variables PD and GP possess five linguistic values: low (L), low-medium (LM), medium (M), high-medium (HM) and high (H). These values are defined in the range $[0; 1]$, and are graphically presented in Figure 1a (linguistic variable PD) and Figure 1b (linguistic variable GP).

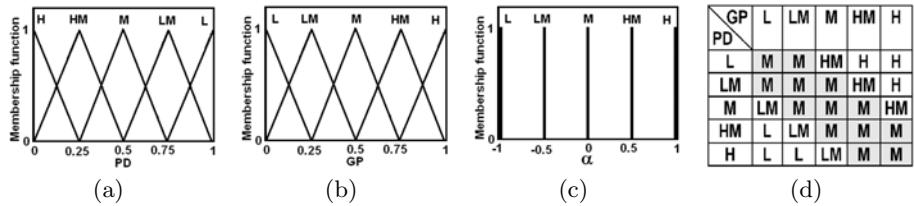


Fig. 1. Graphical representation of fuzzy sets which represent: the input linguistic value PD (a), the input linguistic value GP (b), the output linguistic value α (c); Fuzzy rules for controlling of α parameter value (d)

The output linguistic variable α also possesses five linguistic values such as: low (L), low-medium (LM), medium (M), high-medium (HM) and high (H). These values are defined in the range $[-1; 1]$, and are graphically presented in Figure 1c.

If we study the problem of adaptive control of α parameter we can consider as follows: at the early stage of the evolutionary process, it is useful to reduce greediness and to increase the perturbation factor to explore the search space as much as possible. As the process evolves, exploration needs to be reduced gradually, while increasing our emphasis on exploitation [8]. The rule base that reflect this concept is shown in Figure 1d. Its structure resembles that of a typical fuzzy PI controller, with the switching line shown by the M diagonal. The fuzzy rules are similar as in paper [8].

The value GP ($GP \in [0; 1]$) is computed as follows:

$$GP = \frac{\text{Iteration}}{T_{max}} \quad (4)$$

where: Iteration is a current number of iteration of the algorithm, T_{max} —maximal number of iteration of the algorithm.

Of course, the value of T_{max} parameter should be determined in advance, and usually it is unknown and dependent on the problem. In other words, it needs a careful tuning to get the value of T_{max} . In order to solve a problem connected with determination of T_{max} value, we can estimate a value of T_{max} parameter using method presented in the paper [9].

The value PD is computed as follows based on paper [10]:

$$PD = 1 - \sum_{i=1}^n \sum_{j=1}^M (x_{i,j} - c_i)^2; c_i = \frac{\sum_{j=1}^M x_{i,j}}{M} \quad (5)$$

where: $x_{i,j}$ is a value of i-th decision variable in the j-th individual, M is a number of individuals in population, n is a number of variables in the function being optimized.

In the equation (5), in order to fulfill design assumption of the fuzzy system (the value PD must be equal to 1 when the population diversity is equal to 0), the value of double sum is subtracted from 1.

Additionally, we have normalized the value of PD (in the range $[0; 1]$) in order to obtain the full range of variability for linguistic value PD . The normalization has been dynamically performed after each iteration of the algorithm:

$$PD_{norm,i} = \frac{PD_i - PD_{min}}{PD_{max} - PD_{min}} \quad (6)$$

where: $PD_{norm,i}$ is a normalized value of linguistic variable PD which is inserted into the input of proposed fuzzy controller in i-th iteration of the algorithm, PD_{min} is a lowest value of PD form iteration already performed, PD_{max} is the highest value of PD form iteration already performed, PD_i is a value of PD obtained for i-th iteration of the algorithm.

Additionally, it is worth to notice, that the dynamical normalization is not computationally expensive and the fuzzy controller has been used as a table controller, because the computational complexity for table fuzzy logic controller is very small.

4 Assumed Test Functions

Following test functions chosen from literature [21, 22] have been taken to quality verification of proposed method (GM represents the global minimal value, n represents number of variables in optimized function). These functions are widely used as benchmarks to test the quality of evolutionary algorithms. For all test functions minimization problems are considered.

- De Jong function F1

$$\sum_{i=1}^n x_i^2; -100 \leq x_i \leq 100; GM=0 \text{ in } (x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0); n = 30$$

- Ackley function F2

$$20 - 20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n x_i^2}\right) + \exp(1) - \exp\left(\frac{1}{n} \cdot \sum_{i=1}^n \cos(2 \cdot \pi \cdot x_i)\right); \\ -100 \leq x_i \leq 100; GM=0 \text{ in } (x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0); n = 30$$

- Griewank function F3

$$\frac{1}{4000} \cdot \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\ -600 \leq x_i \leq 600; GM=0 \text{ in } (x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0); n = 30$$

- Rastrigin function F4

$$10 \cdot n + \sum_{i=1}^n (x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i)) \\ -500 \leq x_i \leq 500; GM=0 \text{ in } (x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0); n = 20$$

- Schwefel function F5

$$418.9828872724339 \cdot n - \sum_{i=1}^n \left(x_i \cdot \sin \left(\sqrt{|x_i|} \right) \right)$$

$-500 \leq x_i \leq 500$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (420.96874636, \dots, 420.96874636)$; $n = 30$

- High Conditioned Elliptic function F6

$$\sum_{i=1}^n \left(10^6 \right)^{\frac{i-1}{n-1}} \cdot x_i^2 ; -5 \leq x_i \leq 5; \text{GM}=0 \text{ in } (x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0); n = 30$$

- Non-Continuous Rastrigin function F7

$$\sum_{i=1}^n \left(y_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot y_i) + 10 \right); y_i = \begin{cases} x_i, & \text{when } |x_i| < 0.5 \\ \text{round}(2 \cdot x_i) / 2, & \text{when } |x_i| \geq 0.5 \end{cases}$$

$-500 \leq x_i \leq 500$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$; $n = 30$

- Non-Continuous Expanded Schaffer function F8

$$F(y_1, y_2) + F(y_2, y_3) + \dots + F(y_{n-1}, y_n) + F(y_n, y_1); F(x, y) = 0.5 + \frac{\left(\sin^2 \left(\sqrt{x^2 + y^2} \right) - 0.5 \right)}{(1 + 0.001 \cdot (x^2 + y^2))^2}$$

$$y_i = \begin{cases} x_i, & \text{when } |x_i| < 0.5 \\ \text{round}(2 \cdot x_i) / 2, & \text{when } |x_i| \geq 0.5 \end{cases}$$

$-500 \leq x_i \leq 500$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$; $n = 30$

- Rotated Expanded Schaffer function F9

$$F(x_1, x_2) + F(x_2, x_3) + \dots + F(x_{n-1}, x_n) + F(x_n, x_1); F(x, y) = 0.5 + \frac{\left(\sin^2 \left(\sqrt{x^2 + y^2} \right) - 0.5 \right)}{(1 + 0.001 \cdot (x^2 + y^2))^2}$$

$-500 \leq x_i \leq 500$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$; $n = 30$

- De Jong function F10

$$\sum_{i=1}^n i \cdot x_i^4; -100 \leq x_i \leq 100; \text{GM}=0 \text{ in } (x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0); n = 30$$

- Bohachevsky function F11

$$\sum_{i=1}^n \left(x_i^2 + 2 \cdot x_{i+1}^2 - 0.3 \cdot \cos(3 \cdot \pi \cdot x_i) - 0.4 \cdot \cos(4 \cdot \pi \cdot x_{i+1}) + 0.7 \right)$$

$-15 \leq x_i \leq 15$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$; $n = 30$

- Rosenbrock function F12

$$\sum_{i=1}^n \left(100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right)$$

$-5 \leq x_i \leq 5$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$; $n = 30$

- Scaled Rastrigin function F13

$$10 \cdot n + \sum_{i=1}^n \left(\left(10^{\frac{i-1}{n-1}} \cdot x_i \right)^2 - 10 \cdot \cos \left(2 \cdot \pi \cdot 10^{\frac{i-1}{n-1}} \cdot x_i \right) \right)$$

$-5 \leq x_i \leq 5$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$; $n = 30$

- Skew Rastrigin function F14

$$10 \cdot n + \sum_{i=1}^n \left(y_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot y_i) \right); y_i = \begin{cases} 10 \cdot x_i, & \text{when } x_i > 0 \\ x_i, & \text{otherwise} \end{cases}$$

$-5 \leq x_i \leq 5$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$; $n = 30$

- Schaffer function F15

$$\sum_{i=1}^{n-1} \left(x_i^2 + x_{i+1}^2 \right)^{0.25} \cdot \left[\sin^2 \left(50 \cdot (x_i^2 + x_{i+1}^2)^{0.1} \right) + 1 \right]$$

$-100 \leq x_i \leq 100$; GM=0 in $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$; $n = 30$

5 Description of Experiments

The experiments were performed using test functions presented in section four. In evolutionary algorithm, the following parameters were assumed: individuals were coded as a real-number strings (each gene represents one variable in optimized function), $T_{max}=1000$, $M=50$, probability of crossover = 0.7, probability of mutation = $\frac{1}{n}$. The individuals in population were created randomly. Simple one-point crossover [11] was taken as a crossover operator. The crossover operators depend on cutting of two randomly chosen individuals from population

in one randomly chosen point, and then the cut fragments of chromosomes are exchanged between them. The non-uniform mutation [11, 20] was taken as a mutation operator (for mutation operator the level of inhomogeneity was equal to 2). This operator is responsible for precise tuning up of the evolutionary algorithm to potential solution. During the operation of the evolutionary algorithm only selection operator is changed. The parameter $a=0.3$ is taken for fan selection (identically as in paper [16]). The size of tournament group equal to 2 is assumed for tournament selection. Truncation threshold equal to 0.5 is assumed for truncation selection. In experiments, the evolutionary algorithm was executed 25 times for each test function. In Table 1, the best values of test functions obtained after 25-fold repetition of evolutionary algorithm with different kind of selection methods are presented. The symbols in Table 1 are as follows: SM-chosen selection method, RO-roulette selection, EL-elitist selection, FAN-fan selection, TOU-tournament selection, DET-deterministic selection, TRU-truncation selection, MIX-mix selection, FMIX-fuzzy mix selection.

Table 1. Average values of the best results obtained after 25-fold repetition of evolutionary algorithm for each selection method

SM	F1	F2	F3	F4	F5
RO	376.68±122.89	4.93±0.42	4.32±0.81	4091±1339	1127.20±250.21
EL	169.65±48.61	3.58±0.35	2.50±0.33	1584±491	284.81±133.07
FAN	1.22±0.47	0.32±0.07	0.79±0.15	69.50±10.68	3.13±1.47
TOU	90.06±21.83	3.65±0.24	1.78±0.22	886.23±289.39	407.07±105.02
DET	156.58±32.40	3.74±0.31	2.46±0.30	1729±376.32	381.11±182.32
TRU	55.00±14.35	2.97±0.30	1.47±0.11	508.76±112.41	170.67±94.24
MIX	0.38±0.11	0.20±0.05	0.48±0.11	42.13±9.89	114.80±97.55
FMIX	0.24±0.08	0.13±0.02	0.41±0.12	38.21±7.65	0.67±0.26
SM	F6	F7	F8	F9	F10
RO	53079±17711	9984±2736	9.70±0.77	9.78±0.83	2395497±1126483
EL	6702±3503	4790±1204	6.86±0.56	6.81±0.71	355215±187048
FAN	36.60±30.38	119.22±27.50	3.29±0.56	3.14±0.65	4.56±6.91
TOU	586.28±145.49	2516±704	7.61±0.60	7.74±0.68	46086±22653
DET	6985±2134	4104±923.95	7.33±0.75	7.54±0.69	360566±139065
TRU	265.83±123.37	1552±313	6.70±0.64	6.60±0.57	9521±5487
MIX	19.00±10.85	64.19±13.36	2.79±0.51	2.85±0.51	0.51±0.61
FMIX	13.85±6.25	48.18±10.79	2.79±0.46	2.37±0.35	0.18±0.16
SM	F11	F12	F13	F14	F15
RO	41.75±6.65	7811±2077	129.75±18.49	118.77±16.57	63.14±5.83
EL	26.85±4.03	2603±1067	99.81±12.57	93.37±15.83	52.15±4.85
FAN	1.15±0.45	119.68±59.46	8.25±2.59	5.33±1.79	21.99±2.92
TOU	18.76±1.77	545.96±153.23	66.89±8.51	53.14±8.98	50.14±5.71
DET	27.38±3.66	2155±739.71	86.52±11.79	78.39±9.73	47.11±4.94
TRU	13.38±2.33	80.73±31.90	55.68±10.51	42.32±6.84	42.93±5.32
MIX	0.36±0.10	96.55±37.09	5.01±1.48	3.45±1.59	17.82±2.71
FMIX	0.21±0.05	107.12±44.26	3.10±1.19	1.11±0.72	15.97±1.96

It can be seen from Table 1, that in many cases results obtained using fuzzy mix selection are better than results obtained using other selection methods. In order to check an importance of results presented in Table 1, t-Student statistical test (with 48 degrees of freedom) has been performed for all combinations between results obtained using fuzzy mix selection and results obtained using other selection methods. The obtained results are not presented in this paper with respect to space limitation. But in almost all cases, the results obtained using proposed method are statistically important (with 95% degree of trust). Also, it is important to notice, that in relation to mix selection, the results obtained using fuzzy mix selection are statistically important in 12 cases (on 15 possible).

6 Conclusions

The introduction of fuzzy logic to control of α parameter values makes easier the using of mix selection (we do not need to tune a α parameter value experimentally); also, the results obtained using fuzzy mix selection are better than results obtained using other methods (especially using mix selection without fuzzy logic). The fuzzy control of value of α parameter is based not only on algorithm iterations (as in mix selection), but also on population diversity. Due to these two factors (number of generations and value of population diversity) the proposed fuzzy mix selection is more "immune" to "fall" into the trap of local minimum (i.e. premature convergence problem). Also, the results obtained using fuzzy mix selection are statistically important (their importance has been confirmed using statistical t-Student test). Additionally, the fuzzy mix selection can be used in combinatorial problems.

References

1. Bäck, T.: Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. In: Proc. 1st IEEE Conf. on Evolutionary Computing, pp. 57–62 (1994)
2. Liu, S.-H., Mernik, M., Bryant, B.R.: Entropy-Driven Parameter Control for Evolutionary Algorithms. *Informatica* 31, 41–50 (2007)
3. Motoki, T.: Calculating the expected loss of diversity of selection schemes. *Evolutionary Computation* 10(4), 397–422 (2002)
4. Winkler, S., Affenzeller, M., Wagner, S.: Offspring selection and its effects on genetic propagation in genetic programming based system identification. *Cybernetics and Systems* 2, 549–554 (2008)
5. Xie, H., Zhang, M.: Tuning Selection Pressure in Tournament Selection, Technical Report Series, School of Engineering and Computer Science, Victoria University of Wellington, New Zealand (2009)
6. Jong, K.D.: Parameter setting in eas: a 30 year perspective. In: Parameter Setting in Evolutionary Algorithms, pp. 1–18. Springer, Heidelberg (2007)
7. Slowik, A.: Steering of Balance between Exploration and Exploitation Properties of Evolutionary Algorithms - Mix Selection. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS, vol. 6114, pp. 213–220. Springer, Heidelberg (2010)

8. Xue, F., Sanderson, A.C., Bonissone, P., Graves, R.J.: Fuzzy Logic Controlled Multi-Objective Differential Evolution. In: The 14th IEEE International Conference on Fuzzy Systems, pp. 720–725 (2005)
9. Codreanu, I.: A Procedure based on the ANOVA Method for Estimating the Maximum Number of Generations for Optimization Genetic Algorithms. In: International Semiconductor Conference, CAS 2007, pp. 497–500 (2007)
10. Morrison, R.W., De Jong, K.A.: Measurement of Population Diversity. In: Collet, P., Fonlupt, C., Hao, J.-K., Lutton, E., Schoenauer, M. (eds.) EA 2001. LNCS, vol. 2310, pp. 1047–1074. Springer, Heidelberg (2002)
11. Michalewicz, Z.: *Genetic algorithms + data structures = evolution programs*. Springer, Heidelberg (1992)
12. Zen, S., Zhou Yang, C.T.: Comparison of steady state and elitist selection genetic algorithms. In: Shi Zen, C.T. (ed.) Proc. of 2004 Int. Conf. on Intelligent Mechatronics and Automation, pp. 495–499 (2004)
13. Takaaki, N., Takahiko, K., Keiichiro, Y.: Deterministic Genetic Algorithm. Papers of Technical Meeting on Industrial Instrumentation and Control, pp. 33–36, IEEE Japan (2003)
14. Bickle, T., Thiele, L.: A Comparison of Selection Schemes used in Genetic Algorithms. In: Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, TIK Report, 2nd edn., vol. 11 (December 1995)
15. Muhlenbein, H., Schlierkamp-voosen, D.: Predictive Models for the Breeder Genetic Algorithm. *Evolutionary Computation* 1(1), 2549 (1993)
16. Slowik, A., Bialko, M.: Modified Version of Roulette Selection for Evolution Algorithms – The Fan Selection. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 474–479. Springer, Heidelberg (2004)
17. Bickle, T., Thiele, L.: A Comparison of Selection Schemes used in Evolutionary Algorithms. *Evolutionary Computation* 4(4), 361–394 (1996)
18. Miller, B.L., Goldberg, D.E.: Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems* 9, 193–212 (1995)
19. Bäck, T., Hoffmeister, F.: Extended Selection Mechanisms in Genetic Algorithms. In: Belew, R.K., Booker, L.B. (eds.) Proc. of the 4th Int. Conference on Genetic Algorithms, vol. 1991, pp. 92–99. Morgan Kaufmann Publishers, San Mateo (1991)
20. Zhaoa, X., Gaob, X.-S., Hu, Z.-C.: Evolutionary programming based on non-uniform mutation. *Applied Mathematics and Computation* 192(1), 1–11 (2007)
21. 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 Number 2005005 (Kanpur Genetic Algorithms Laboratory, IIT Kanpur) (May 2005)
22. Hansen, N., Kern, S.: Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiño, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 282–291. Springer, Heidelberg (2004)
23. Derrac, J., García, S., Herrera, F.: A first study on the use of coevolutionary algorithms for instance and feature selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
24. Corchado, E., Abraham, A., Ferreira de Carvalho, A.C.P.L.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)

Hybridization of Evolutionary Algorithm with Yule Walker Method to Design Minimal Phase Digital Filters with Arbitrary Amplitude Characteristics

Adam Slowik

Department of Electronics and Computer Science, Koszalin University of Technology,
Sniadeckich 2 Street, 75-453 Koszalin, Poland
aslowik@ie.tu.koszalin.pl

Abstract. In this paper a method of design minimal phase digital filters with arbitrary amplitude characteristics is presented. The filters designed using proposed method can be directly implemented in the hardware. Filter coefficients obtained using proposed method are immune on rounding error. Because these coefficients are created in required bit word length (based on numerical format which is used in given DSP system). The proposed method is based on connection of evolutionary algorithm and Yule Walker method. Due to this hybridization, digital filters are designed faster (than using only evolutionary algorithm) and possess much better properties (then using only Yule Walker method). Digital filters obtained using proposed method are ready to be implemented in DSP system without any additional errors. The four minimal phase digital filters (with coefficients in Q.15 format) and with arbitrary amplitude characteristics are designed using proposed method. The results obtained using proposed method are compared with results obtained using other techniques taken from literature.

1 Introduction

During IIR (Infinite Impulse Response) digital filters design, it is important to design the minimal phase filters. These filters have two main advantages: reduced filter length and minimal group delay. Minimal phase digital filters generally require fewer computations and less memory than linear phase filters [1, 2]. The main goals during minimal phase digital filters design are assurance of filter stability, that is location of all zeros of filter transmittance in unitary circle in the z plane, and fulfilment of design assumptions connected with the shape of amplitude characteristics. In order to obtain assumed characteristics we can use one of existing approximations such as: Butterworth, Chebyshev or Cauer during the design process. But, the problem is complicated in the case, when designed filter should have a non-standard amplitude characteristics [3]. The non-standard amplitude characteristics are widely used in different kind of amplitude equalizers. Therefore, in this case, the standard approximations are useless and we must

use one of optimization techniques. The one of possible methods to design minimal phase IIR digital filters with arbitrary amplitude characteristics is the Yule Walker (YW) method [4, 5]. This method designs recursive IIR digital filters using a least-squares fit (often Yule Walker method can find a local optimum) to a specified frequency response [4, 5]. But the problem is complicated when we want to use a designed filter in programmable fixed-point DSP processors that are used for real-world applications. Then the filter coefficients obtained using Yule Walker algorithm must be scaled to the range $[-1; 1]$ and next, these coefficients must be quantized to the fixed-point numbers. In real world applications the 16-bit fixed-point (Q.15) format is commonly used in most 16-bit fixed-point DSP processors, such as the TMS320C5000 [6, 7]. Q.15 format represents numbers in the range of -1 to $1 - 2^{-15}$ using a sign bit and 15 fractional bits with 2 complement format [8]. When the coefficients (for IIR digital filter) obtained using Yule Walker algorithm will be transformed to the Q.15 format, and applied in DSP processor, the shape of amplitude characteristics of designed filter probably will be changed, because IIR filters are very sensitive to variation of values of filter coefficients [9]. Also, in literature, we can find some evolutionary algorithms [12, 13], which are used in design of digital filters with finite bit word length [10] or where genetics algorithms are used to design of IIR digital filters with minimal phase (for example [11]). But the main disadvantage of evolutionary design of digital filters is a high computational time. The time of design of digital filters using evolutionary methods is much longer than using Yule Walker method. Therefore in this paper, we want to hybridize (hybrid method are more often used in practice [14-17]) a evolutionary method with Yule Walker method. Due to this hybridization, in proposed digital filters design method, the computational time should be shortened (in the case of evolutionary method) and the obtained filter coefficients should be ready for implementation in DSP processor without any additional errors (in the case of Yule Walker method). As a test of proposed method, the four 16-bit minimal phase IIR digital filters with amplitude characteristics: linearly growing, and linearly falling, and non-linearly growing, and non-linearly falling were designed. The method described in this paper is named HDFD (*Hybrid Digital Filter Design*).

2 IIR Digital Filters

The transmittance of IIR filters in z domain can be described using following equation:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_n \cdot z^{-n}}{a_0 - (a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n})} \quad (1)$$

In most cases the value of a_0 coefficient is equal to 1. But in this paper we assume that value a_0 is a parameter, because if the filter coefficients obtained using Yule Walker method will be scaled and quantized then the value of a_0 is often not equal to 1. The main goal of the design algorithm of digital filters is to find a such a set of filter coefficients a_i, b_i ($i \in [0; n]$, where n is a filter order) in order

assure stability of designed filter, minimal phase of the filter, and fulfill all design assumptions. However, if we want to obtain a digital filter, which will be resistive on rounding errors, the filter coefficients must take exactly determined values dependent on number of bits used in representation of each filter coefficient. In the case, when the number of bits which are used to represent the filter coefficient is equal to nb , then the $M=nb-1$ bits are allowable to realization of its a value (one bit is taken as a sign bit). Therefore, the digital filter coefficients can take the values from the following domain D (in fixed-point format Q.M):

$$D = \left[\frac{(-1) \cdot 2^M}{2^M}; \frac{2^M - 1}{2^M} \right] \quad (2)$$

In the 2's complement fractional representation, an nb bit binary word can represent 2^{nb} equally spaced numbers from $\frac{(-1) \cdot 2^M}{2^M} = -1$ to $\frac{2^M - 1}{2^M} = 1 - 2^{-M}$ (see equation 2).

The binary word BW which consists of nb bits (bw_i): $BW=bw_M, bw_{M-1}, bw_{M-2}, \dots, bw_2, bw_1, bw_0$ we interpret as a fractional number x :

$$x = -(b_M) + \sum_{i=0}^{M-1} \left(2^{i-M} \cdot bw_i \right) \quad (3)$$

Of course if we use a fractional number in Q.M format, the value of coefficient a_0 will be not equal to 1, but a_0 will be equal to $1 - 2^{-M}$.

In order to assure, the stability of digital filters, the poles of the function (1) must be placed in unitary circle in the z plane. Of course, in order to assure, that designed filter will be minimal phase digital filter, the zeros of the function (1) must be placed also in unitary circle in the z plane.

3 Proposed Method HDFD

The proposed method HDFD consists of ninth steps.

In the first step, the set D (see equation 2) consisting of 2^{nb} values is created. For each value from the set D the index value is assigned. The first value from set D possesses index number 1, the last value from the same set is represented by index 2^{nb} . Next, the population Pop is randomly created. The population Pop consists of $PopSize$ individuals. Each individual in population consists of $2 \cdot (n + 1)$ genes (n is represents the filter order). Each gene takes one integer value form the range $[1; 2^{nb}]$. The value written down in each gene, points to the adequate filter coefficient from the set D .

In the second step, the given digital filter is designed using Yule Walker method. The filter coefficients obtained using YW method (for given digital filter) are scaled into the range $[-1; 1]$. Next, these scaled filter coefficients are quantized into fixed-point format selected by user (for example into Q.15 format). At the end of this step, the scaled and quantized filter coefficients are written down into the randomly selected individuals from population Pop .

In the third step, an evaluation of all individuals using objective function FC is performed (objective function FC is described in the fourth section of

this paper). Presented evolutionary algorithm tends to minimize the objective function FC .

In the fourth step, the best individual (having the lowest value of objective function FC) is selected from current population Pop . In the first algorithm iteration, selected the best individual is remembered in the variable $TheBest$. During next algorithm iterations, selected the best individual is remembered as the $TheBest$, if and only if the value of its objective function FC is lower than the value of the objective function for individual actually stored in the variable $TheBest$.

In the fifth step, a selection of individuals to the new population is performed. The tournament selection [12, 13] with the size of tournament group equal to 2 is chosen as a selection operator in the proposed algorithm.

In the sixth step, we have checked if the best individual is located in the population after selection process. If the best individual has been lost then the individual $TheBest$ is inserted in the place of the best individual in current population Pop . Otherwise (when the best individual "survive" the selection process), any changes are not performed.

In the seventh step, a cross-over of individuals in the population Pop is performed. A single one point cross-over [12, 13] is chosen as a cross-over operator. The single cross-over operator, depends on choosing (with probability PC) a pair of individuals form the population Pop , and cutting in randomly chosen point for each pair of individuals. After individual cutting, the cut fragments of parental individuals are exchanged between them. Due to this exchange, the two child individuals are created. The two new individuals (child individuals) are inserted into the place of their parental individuals in the population Pop .

In the eighth step, a mutation of individuals is performed. The mutation operator is executed with probability PM for each gene in each individual in the population Pop . If i -th gene from j -th individual is selected to the mutation, then its new value is determined as follows:

$$NG_{i,j} = \begin{cases} G_{i,j} + 1, & \text{when } r < 0.5 \text{ and } G_{i,j} < 2^{nb} \\ G_{i,j} - 1, & \text{when } r \geq 0.5 \text{ and } G_{i,j} > 1 \\ G_{i,j}, & \text{otherwise} \end{cases} \quad (4)$$

where: $NG_{i,j}$ is a new value of i -th gene in j -th individual in population Pop , $G_{i,j}$ is a current value of i -th gene in j -th individual, r is a random value from the range $[0; 1]$. The $NG_{i,j}$ computation time is very fast. Because in the mutation operator only index of the value is changed. The all possible values of filter coefficients for given DSP system are located in the table (this table is created once during algorithm initialization).

In the ninth step, algorithm termination criteria are checked. Reaching of maximal number of algorithm iteration G_{max} or reaching of solution having the value of the objective function FC equal to 0 are assumed as a termination conditions in the proposed algorithm. If algorithm termination criteria are fulfilled, then the algorithm is stopped and the result stored in $TheBest$ individual is returned as a solution of a given problem. But, if the algorithm termination criteria are not fulfilled, then a jump to the second step of the proposed algorithm is performed.

4 Objective Function FC

The objective function FC is computed as follows (in equation (5-11) the index i represents i -th individuals in population):

$$FC_i = AmplitudeError_i + w \cdot (StabError_i + MinPhaseError_i) \quad (5)$$

$$AmplitudeError_i = \sum_{k=1}^R AmpErr_{i,k} \quad (6)$$

$$AmpErr_{i,k} = \begin{cases} H(f_k)_i - Uppper_{i,k}, & \text{when } H(f_k)_i > Uppper_{i,k} \\ Lower_{i,k} - H(f_k)_i, & \text{when } H(f_k)_i < Lower_{i,k} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

$$StabError_i = \sum_{j=1}^J StabErr_{i,j} \quad (8)$$

$$StabErr_{i,j} = \begin{cases} |p_{i,j}| - 1, & \text{when } |p_{i,j}| \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$MinPhaseError_i = \sum_{q=1}^Q PhaseErr_{i,q} \quad (10)$$

$$PhaseErr_{i,q} = \begin{cases} |z_{i,q}| - 1, & \text{when } |z_{i,q}| \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where: w is a value of penalty factor (during experiments $w = 10^5$ is assumed; the value of w has been taken experimentally, but in general, the value of w must be higher, than the highest value which can be obtained in $AmplitudeError$), $AmpErr_{i,k}$ is a partial value of amplitude characteristics error for k -th value of normalized frequency, $H(f_k)_i$ is a value of amplitude characteristics for k -th value of normalized frequency f , $Lower_{i,k}$ is a value of lower constraint for amplitude characteristics value for k -th value of normalized frequency, $Uppper_{i,k}$ is a value of upper constraint for amplitude characteristics value for k -th value of normalized frequency, $StabErr_{i,j}$ is a partial filter stability error for j -th pole of transmittance function, J is a number of poles of transmittance function, $|p_{i,j}|$ is a value of module for j -th pole of transmittance function, $PhaseErr_{i,q}$ is a partial filter minimal phase error for q -th zero of transmittance function, Q is a number of zeros of transmittance function, $|z_{i,q}|$ is a value of module for q -th zero of transmittance function.

In order to obtain the value of objective function FC for i -th individual in the population, first the amplitude characteristics $H(f)_i$ which coefficients are stored in the i -th individual is computed. The amplitude characteristics is computed using R values of normalized frequency $f \in [0; 1]$ (where 1 represents the Nyquist frequency; in proposed method normalized frequency is divided into R points). Also, the poles and zeros of transmittance function (see equation 1) are computed for each individual in the population Pop . When we have amplitude characteristics and the values of poles and zeros of the transmittance function for i -th individuals, we can compute the objective function FC .

5 Description of Experiments

The four sixteen bit (in Q.15 format) minimal phase digital filters with non-standard amplitude characteristics were designed in order to test the quality of the proposed method. We have assumed following amplitude characteristics: linearly falling (Figure 1a), linearly growing (Figure 1b), quadratic non-linearly falling (Figure 1c), and quadratic non-linearly growing (Figure 1d). The parameters of these characteristics are presented in Figure 1. The axis x represents normalized frequency in the range [0; 1], where value 1 represents Nyquist frequency. The axis y represents attenuation of amplitude characteristics in dB.

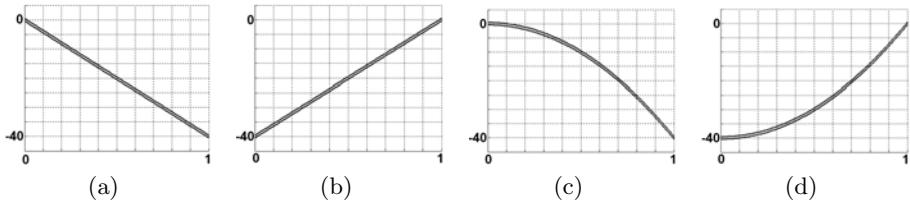


Fig. 1. Shape of amplitude characteristics of designed digital filters: linearly falling LF (a), linearly growing LG (b), non-linearly falling NF (c) and non-linearly growing NG (d)

During experiments we have only changed the values of maximal admissible deviations of amplitude characteristics from $+/- 0$ [dB] to $+/- 0.5$ [dB] with step 0.1 [dB]. Additionally, we have assumed, that normalized frequency was divided into 128 points ($R=128$), and that 10-th order IIR digital filters are designed ($n=10$). Also, we have assumed, that digital filters will be realized using 16 bits word ($nb=16$) in Q.15 fractional format. The remaining parameters of evolutionary algorithm are as follows: number of individuals in population $PopSize=10$, maximal value of evolutionary algorithm generations $G_{max} = 3000$, the probability of cross-over $PC=0.7$, the probability of mutation $PM = 3/(2 \cdot n + 1)$. The symbols in table 1 are as follows: ACh - the kind of amplitude characteristics, $ADev$ - acceptable deviations of amplitude characteristics, $HDFD$ - proposed method, EA - the proposed method without initialization using solution obtained by Yule Walker method, YW - Yule Walker method, $YWSQ$ - Yule Walker method with scaling and quantization of the value of filter coefficients, $time[s]$ - time required to obtain given solution. In this paper, we only present the objective function values for the best results obtained after 25-fold repetition of $HDFD$ and EA method. The Yule Walker method is a deterministic method, therefore after 25-fold repetition, we always obtain the same solution. The values in parentheses represent the number of designed digital filters (where the value of objective function FC was equal to 0).

Based on results presented in table 1, we can see, that using proposed method ($HDFD$) we can design digital filter which can be directly implemented in DSP processors more efficiently than using other methods. Due to hybridization of the evolutionary algorithm with Yule Walker algorithm the results obtained using

Table 1. The best values of objective function FC obtained for each design algorithm

ACh	ADev [dB]	HDFD		EA		YW		YWSQ	
		FC	time [s]	FC	time [s]	FC	time [s]	FC	time [s]
LF	0.5	0 (25)	0.1173	2.5409e+3	20.0721	0.0810	0.0117	24.4591	0.0122
	0.4	0 (25)	0.0806	2.6793e+3	21.4532	0.1859	0.0118	25.3168	0.0132
	0.3	0 (25)	0.5280	2.4790e+3	21.1210	0.4074	0.0119	26.2168	0.0121
	0.2	0 (22)	0.9973	2.5768e+3	20.0711	0.7172	0.0121	27.1776	0.0125
	0.1	0 (1)	19.3967	2.7737e+3	21.3210	1.1747	0.0115	28.2851	0.0126
	0.0	1.8294	21.8165	2.1953e+3	20.1278	4.3616	0.0115	32.1638	0.0119
LG	0.5	0 (25)	0.0880	2.8124e+3	21.2100	0.1840	0.0117	19.8018	0.0123
	0.4	0 (25)	0.1027	2.3786e+3	20.9871	0.3047	0.0119	20.6982	0.0126
	0.3	0 (25)	0.8800	2.7873e+3	20.9212	0.5047	0.0118	21.7346	0.0130
	0.2	0 (7)	2.6767	2.4563e+3	21.3410	0.8079	0.0115	23.1162	0.0129
	0.1	0.2559	21.5643	2.3561e+3	21.0720	1.4024	0.0120	24.9510	0.0129
	0.0	5.9838	21.6453	2.5335e+3	21.0721	13.3234	0.0117	36.8532	0.0131
NF	0.5	0 (25)	0.0220	1.6157e+3	21.1265	0.1368	0.0118	0.3421	0.0128
	0.4	0 (25)	0.0293	1.5905e+3	20.9821	0.2368	0.0117	0.5101	0.0123
	0.3	0 (25)	0.0220	1.6865e+3	20.9891	0.3822	0.0117	0.7101	0.0124
	0.2	0 (25)	0.0660	1.5721e+3	21.1236	0.5839	0.0117	0.9407	0.0122
	0.1	0 (25)	0.7407	1.6321e+3	21.3212	0.9995	0.0114	1.2471	0.0123
	0.0	1.1636	21.8531	1.5695e+3	21.2242	5.4538	0.0114	5.3452	0.0131
NG	0.5	1.1501e-9	21.9110	3.4222e+3	21.8902	0	0.0112	4.3200	0.0127
	0.4	0 (1)	2.0020	3.1432e+3	21.6378	0.0628	0.0115	4.9200	0.0124
	0.3	3.3724e-8	21.3043	3.6298e+3	21.0402	0.1628	0.0116	5.5442	0.0124
	0.2	0.9980	21.7435	3.2156e+3	21.6734	1.6533	0.0124	6.2442	0.0123
	0.1	3.3120	21.8924	3.5498e+3	22.0100	6.4270	0.0121	8.0199	0.0128
	0.0	12.2208	21.7854	3.4269e+3	21.5700	16.0312	0.0116	17.4478	0.0125

HDFD are better than results obtained using other methods. Also, in *HDFD* method the time of design of digital filters has been shortened in relation to evolutionary method (EA).

6 Conclusions

In this paper hybrid evolutionary method to effective design minimal phase digital filters with non-standard amplitude characteristics is presented. The main advantage of proposed method is that designed digital filters can be directly implemented in DSP systems without changes of their properties. Also, it can be noticed, that digital filters with coefficients in different arithmetic format (depend on DSP system) can be easily designed using proposed method, and these filters are always ready to be directly implemented into the hardware (without change of their properties). Due to hybridization the evolutionary algorithm with Yule Walker algorithm the results obtained using *HDFD* are better than results obtained using other methods and these results are obtained faster (in relation to evolutionary algorithm without hybridization with Yule Walker method).

References

1. Karaboga, N., Cetinkaya, B.: Performance Comparison of Genetic Algorithm based Design Methods of Digital Filters with Optimal Magnitude Response and Minimum Phase. In: 46th IEEE Midwest Symposium on Circuits and Systems, Egypt (2003)
2. Venkata, N.D., Evans, B.L.: Optimal Design of Real and Complex Minimum Phase Digital FIR Filters. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 1145–1148 (1999)
3. Slowik, A., Bialko, M.: Design and Optimization of IIR Digital Filters with Non-standard Characteristics Using Continuous Ant Colony Optimization Algorithm. In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) SETN 2008. LNCS (LNAI), vol. 5138, pp. 395–400. Springer, Heidelberg (2008)
4. Orfanidis, S.J.: Introduction to Signal Processing. Prentice-Hall, Englewood Cliffs (1995)
5. Ding, H., Lu, J., Qiu, X., Xu, B.: Anadaptive speech enhancement method for siren noise cancellation. *Applied Acoustics* 65, 385–399 (2004)
6. TMS320C54x DSP Library Programmers Reference, Texas Instruments, Dallas, TX, SPRU422b (2001)
7. TMS320C55x DSP Library Programmers Reference, Texas Instruments, Dallas, TX, SPRU422F (2002)
8. Gan, W.-S., Kuo, S.M.: Teaching DSP Software Development: From Design to Fixed-Point Implementations. *IEEE Transactions on Education* 49(1) (2006)
9. Lyons, R.: Introduction to digital signal processing. In: WKL, Warsaw (2000)
10. Baicher, G.S.: Optimization of Finite Word Length Coefficient IIR Digital Filters Through Genetic Algorithms – A Comparative Study. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4222, pp. 641–650. Springer, Heidelberg (2006)
11. Karaboga, N., Cetinkaya, B.: Design of Minimum Phase Digital IIR Filters by Using Genetic Algorithm. In: Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004, Espoo, Finland, June 9-11 (2004)
12. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs. Springer, Heidelberg (1992)
13. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company Inc., New York (1989)
14. Derrac, J., García, S., Herrera, F.: A first study on the use of coevolutionary algorithms for instance and feature selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
15. Corchado, E., Abraham, A., Ferreira de Carvalho, A.C.P.L.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)
16. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants – Evolutionary and Neural Methods of Training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS (LNAI), vol. 6076, pp. 590–597. Springer, Heidelberg (2010)
17. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72(13–15), 2729–2730 (2009)

Automatic Identification Approach for Sea Surface Bubbles Detection

Juan José Fuertes¹, Carlos M. Travieso², and J.B. Alonso²

¹ Instituto Interuniversitario de Investigación en Bioingeniería y Tecnología Orientada al Ser Humano, Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, España
jjfuertes@labhuman.i3bh.es

² Dpto. de Señales y Comunicaciones, Universidad de Las Palmas de Gran Canaria
Centro Tecnológico para la Innovación en Comunicaciones
Campus de Tafira, 35017 Las Palmas de Gran Canaria, España
{ctravieso,jalonso}@dsc.ulpgc.es

Abstract. In this work a novel system for bubbles detection on sea surface images is presented. This application is basic to verify radiometer satellite systems which are used to the study of the floor humidity and the sea salinity. 160 images of 8 kinds of salinity have been processed, 20 per class. Two main steps have been implemented: image pre-processing and enhancing in order to improve the bubbles features, and segmentation and bubbles detection. A combination system has been performed with Support Vector Machines (SVM) in order to detect the sea salinity, showing a recognition rate of 95.43%.

Keywords: Sea Surface Image, Bubble Detection, Image Processing, Pattern Recognition.

1 Introduction

The weather and nature disaster prediction depend on the availability of global and periodicity information from humidity and sea salinity on their superficial layers. In 1999, European Space Agency (ESA) approved the SMOS mission (Soil Moisture and Ocean Salinity), whose objectives were the study of the floor humidity and the sea salinity of the oceanic surface using a radiometer in the L-band [1][2]. The SMOS mission was based on whole objects that send out energy characterized by the brightness temperature (BT) that is simply defined as the multiplication of the physical temperature by the emissivity. A radiometer measures the emitted energy in the L-Band (1400-1427 MHz) that is reserved for the passive observation, so if the atmosphere temperature is known the emissivity can be measured thanks to the brightness temperature. Consequently, as the emissivity depends on the composition of the objects, the earth humidity and the sea salinity are considered.

The ESA supported two experiments during 2000 and 2001 with the objective of improving the models which show the emissivity on L-Band and verify the sea surface salinity from the radiometer measure. These experiments show the effect of the sea state, the wind and the waves relative to the brightness temperature. However, it is

necessary to fix the contribution of the sea foam and the marine surface with the conditions of the rain and the emissivity, depending on the sea salinity. Then, a new experiment began in 2003: it was called FROG (Foam, Rain, Oil Slicks and GPS-reflections). The goal of this new test was to improve the emissivity models of the sea surface considering the effects of the rain and the foam which had not been considerate before [3]. It is necessary to know the distribution of bubbles ratios of the sea surface and the thickness of foam layer for each kind of salinity in order to understand better the theory models.

Therefore, the goal of this present work is to isolate the foam of the images and the bubbles of the surface layer applying image and analysis techniques, in order to know the kind of salinity from bubbles and foam patterns. This is one of the reasons why this novel system in connection with Hybrid Artificial Systems [4, 5] is proposed, in order to set up a possible investigation about automated systems for bubbles detection.

The next block diagram that we can see in Fig. 1 has been proposed in order to detect the different layers and the variability in the same layer. Then, the features extraction and classification are possible.

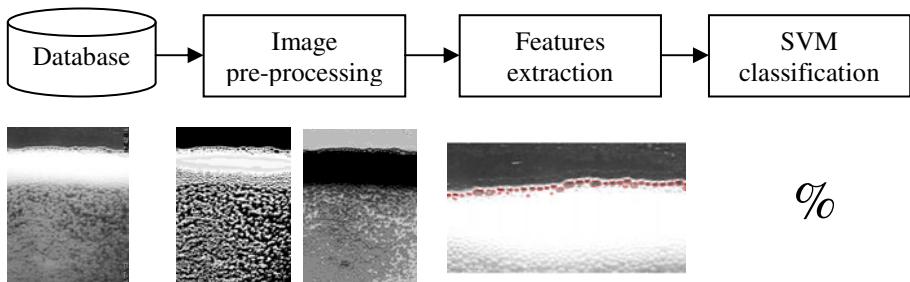


Fig. 1. Schedule of the full automated system

2 Database Used

Initially, it was built a pool where natural and salt sea water was mixed fairly in order to know the quantity of salt dissolved. At deep down, it was placed an air diffuser array with 104 elements which simulated the spin of the sea pumping a constant air flow in order to generate artificial bubbles and the foam layer of the sea. If the air flow is constant, the quantity of bubbles and the thickness of foam layer depend on the kind of salinity (see Fig. 2).

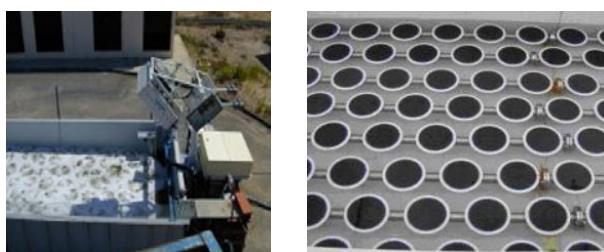


Fig. 2. The pool and air diffusers array situated on its bottom

The studied salinities were 0 psu, 5 psu, 10 psu, 15 psu, 20 psu, 25 psu, 30 psu and 35 psu, where psu stands for “practical salinity unit”. These images were recorded with a video-camera, multisession and with different videos for session. The images were extracted each 3 seconds from the video, storing each one on gray-scale, with 8 bits and a resolution of 640x480 pixels. This quality was chosen in order to have initial resolution for bubbles detection.

For this present work, the Radiometer Research Group from Polytechnic University of Catalonia (UPC) has handed over 160 images, 20 per class. At summary, the characteristics of those images can be seen in Table 1:

Table 1. Image features

Parameters	Data
Number of classes	8
Number of samples per class	20
Acquisition and Quantification	Gray Scale (8 bits, 256 levels)
Resolution	640x480 pixels
Images Format	JPEG

3 Image Pre-processing

The first step of the proposed system is to detect the different layers of the sea-images and the bubbles shape, but it is not a trivial task due to the low contrast of the images and the similarity between classes. Therefore, it was required an image pre-processing step in order to increase the contrast between the different layers and the variability in the same layer.

The proposed method to enhance the details is based on the difference between a transform of the image which notices the relative difference of the luminance and the non-absolute difference (based on the logarithmic behavior of the human system with gamma correction 0.5), and a low-pass filtered image (in order to reduce the noise of high frequency and the edge), resulting in an image with raising regions of the sea-images, as it is shown in Fig. 3.

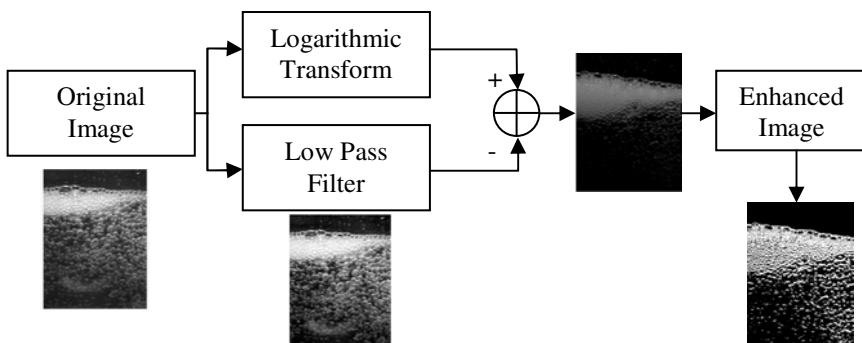


Fig. 3. Enhancing sea-images features

The “*low pass filter*” function used is shown in (1) [6];

$$g(i, j) = \left[\frac{C_{MAX} - C_{MIN}}{f(i, j)_{MAX} - f(i, j)_{MIN}} \right] [f(i, j) - f(i, j)_{MIN}] + C_{MIN} \quad (1)$$

where $f(i,j)$ is the gray level of the input image, $f(i,j)_{MAX}$ is the maximum intensity in the input image, $f(i,j)_{MIN}$ is the minimum intensity in the input image and, C_{MAX} and C_{MIN} correspond to the maximum and minimum values desired in the final histogram. When this transform is applied, the variation of the gray levels is minor. In particular, the histogram was fixed after several tests with gray levels C_{MAX} , C_{MIN} 70 and 40 respectively.

If the difference found between both images is mayor or minor, the enhancement will be mayor or minor in order to increase the dynamic range of the interesting zone. The applied function to “*enhance the image*” responses to (2) [7]:

$$g(x, y) = k_1 \frac{M}{\sigma(x, y)} [f(x, y) - m(x, y)] + k_2 m(x, y) \quad (2)$$

where, $k_1 \cdot \frac{M}{\sigma(x, y)}$ is the gain factor used to enhance the image contrast, and $k_2 \cdot m(x, y)$

is a factor to restore the medium intensity image level ($m(i,j)$) and $\sigma(i,j)$ are the average and standard deviation intensity levels of a centered window in the pixel (x,y) . If the window is small and the image is big, the execution of enhancement is slow and it has to use windows with mayor size in order to get better results. In particular, for our database, they were performed different experiments and the best results were obtained with a size of 35x35 pixels, using k_1 and k_2 with 0.4 and 0.9 respectively. It makes easy segmentation.

4 Segmentation and Bubbles Detection

The main criterion in order to define the different layers and the bubbles was the relation between the number of “on” pixels and “off” pixels in each region, defining as “on pixels”, pixels whose value is 1 when the enhanced image is binarized. It was used this criterion because the foam layer is the image layer with greater number of “on” pixels per layer. In order to add a region to the foam layer, it has to have 70% of “on” pixels or more, and to verify the following conditions:

1. The size of the foam layer cannot exceed a defined value. To add a region, if it exceeds that threshold, the interesting region does not below to the foam layer. Once we have compared the results for different experimental thresholds, the determined size was the 12% of the real size of the image.
2. If the percentage of “on” pixels for a region is lower than 75% and besides, the size of that region exceeds the 12% of the real size of the image, then the interesting region does not below to the foam layer.
3. If the interesting region exceeds the 12% of the size of the image, it only will be added to the foam layer if it is very small versus the layer considered

foam layer, and with both layers it is impossible to reach a sufficient size like to be considered foam layer.

Finally, the bubbles edge in the surface of the foam layer was got combining directional detectors and segmentation with multilayer threshold, in order to achieve the best bubbles detection and accuracy recognition. The method implemented with directional edge detectors is called method 1, and the method 2 is the multilevel segmentation:

Method 1

The enhanced image contains much more information in its different layers, but the present work only weighs up the foam layer. So, it was applied eight convolution masks, one per each angle, 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°, on the enhanced image. Figure 4 shows the edges from the both methods and its logic combination.

Method 2

This method tries to find the edge based on the different gray levels which belong to the bubbles and the gray levels of the rest of pixels of the foam layer. The foam layer is divided in diverse regions applying the segmentation with multilevel threshold, and finally, the image is binarized, getting the bubbles edges (see Fig. 4).

Combination System: Evaluation

In order to justify the use of both methods in the process of bubbles detection, it is shown an example in Figure 4 where it is defined the detection of both methods, and after logic combination, the detection of the most bubbles. The method 1 does not detect certain bubbles of small size. On the other hand, method 2 detects some bubbles with more precision. Combining both methods is achieved a better approximation [8].

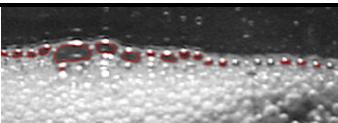
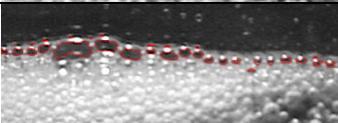
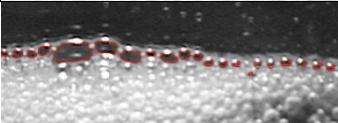
	Method 1: Bubbles edges for directional detectors
	Method 2: Bubbles edges for multi-level threshold
	Bubbles edges combining both previous methods

Fig. 4. Result of our proposed method for bubbles detection

5 Classification System

The main goal of this work was to detect the sea salinity helping scientists to predict weather and nature disaster. Therefore, once the pre-processing and segmentation image has been performed, it has been used Neuronal Network (NN) [9] and Support Vector Machine (SVM) [10-12] to verify how the proposed method works, in supervised classification and using features from bubbles edges and other parameters (contrast, area, correlation, etc.)

We have collected 160 samples from 8 classes, 20 samples per class. Six images of each class are chosen randomly as training samples and the remaining four images are used to test the system. The results obtained with SVM are similar to NN, therefore, we are going to show both of them in average (%) and typical deviation (std). Each test has been done 5 times using RBF (radial basis functions) SVM kernel and NN, finding the result through a cross-validation strategy.

Firstly, it has been differenced the fresh- and salt-water with a success rate of 100%:

Table 2. Fresh- and salt-water detection

Method	Success (%)	Training Time (s)	Simulation Time (s/sample)
NN	100 %	2'4813	8'25*10 ⁻⁵
SVM	100 %	0'0372	6'2*10 ⁻⁴

Afterwards, it has been determined the degree of water salinity. It has been performed three strategies: the first (1 node) with one classifier and seven salinity degrees (seven outputs of the classifier); the second (2 nodes) with two classifiers (the first classifier with six outputs, one output for salinities 10 and 20 psu and five outputs for the rest of salinity degrees, and a second SVM to differentiate between salinities 10 and 20 psu); and the third with decision tree strategy. Each experiment has been done 5 times:

Table 3. Identification of different kinds of salt-water

Method	Technique	Success (%)	Training Time (s)	Simulation Time (s/sample)
NN	1 node	92'07%	2'17	0'0004
	2 nodes	93'16%	3'57	0'00083
	Tree	93'07%	4,66	0'0013
SVM	1 node	93'91%	0'48	0'0072
	2 nodes	94'92%	0'50	0'079
	Tree	94'78%	0'31	0'0049

The best result of the first step is 100% independently of the classifier used but we will considerer NN because it is faster when simulation step is performed. In the second step it is obtained a 94'92% of accuracy recognition with two nodes, but we will considerer a decision tree strategy with 94'78% of recognition rate because it is faster. In conclusion, using a 60% of our database for training process the final recognition rate is $95'43 \pm 0.3\%$ when the two steps are combining.

6 Discussion and Conclusions

In this present work it has been proposed a simple automated image processing system for bubbles detection. The measure of quality has been implemented with supervised classification, reaching a 95.43% in order to discriminate the different kinds of salinity. This experiment is the indicator of a good result of the proposed method that is the most important issue in this work.

In addition, we are working in the combination of new features of the images in order to improve the system accuracy. For the future, the improvement of the image processing techniques and the extraction of new features could be helpful to avoid nature disasters.

Acknowledgments. This present work has been supported by The Spanish Ministry of Foreign Affairs under the research project with reference "D/033858/10" and it has been implemented with the database given for Prof. Adriano Camps from Polytechnic University of Catalonia (UPC) at Spain.

References

1. Camps, A., Vall-llosera, M., Villarin, R., Reul, N., Chapron, B., Corbella, I., Duffo, N., Torres, F., Miranda, J.J., Sabia, R., Monerris, A., Rodríguez, R.: The Emissivity of Foam-Covered Water Surface at L-Band: Theoretical Modeling and Experimental Results From the Frog 2003 Field Experiment. *IEEE Transactions on Geoscience and Remote Sensing* 43(5), 925–937 (2003)
2. Villarino, R., Camps, A., Vall-llosera, M., Miranda, J., Arenas, J.: Sea Foam Effects on the Brightness Temperature at L-Band. In: *IEEE International Geoscience and Remote Sensing Symposium IGARSS 2003*, vol. 5, pp. 3076–3078 (2003)
3. Villarino, R.M.: Empirical Determination of the Sea Surface Emissivity at L-Band: A contribution to ESA's SMOS Earth Explorer Mission. PhD Document, Polytechnic University of Catalonia, Spain (2004)
4. Corchado, E., Abraham, A., Carvalho, A.C.P.L.F.D.: Hybrid intelligent algorithms and applications. *Inf. Sci.*, 2633-2634 (2010)
5. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72, 2729–2730 (2009)
6. Gonzalez, R.C., Wood, R.E.: *Digital Image Processing*. Addison- Wesley, London (2002)
7. Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis and Machine Vision*. Thomson Engineering (2007)

8. Mokhtarian, F., Abbasi, S.: Matching shapes with self-intersections: application to leaf classification. *IEEE Transactions on Image Processing* 13(5), 653–661 (2004)
9. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
10. Steinwart, I., Christmann, A.: Support Vector Machines. Springer, New York (2008)
11. Wang, L.P.: Support Vector Machines: Theory and Application. Springer, Berlin (2005)
12. Kecman, V.: Learning and Soft Computing, Support Vector Machines, Neural Networks and Fuzzy Logic Models. The MIT Press, Cambridge (2001)

The Application of Artificial Intelligence Hybrid in Traffic Flow

Ilija Tanackov, Vuk Bogdanović, Jovan Tepić, Siniša Sremac, and Nenad Ruškić

University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6,
21000 Novi Sad, Serbia
ilijat@uns.ac.rs

Abstract. Traffic flow is a specific line of moving vehicles where the degree of interaction between the factors of the flow is extremely high. The vehicles' interaction is a consequence of human imperfection in driving. For that reason, the determination of traffic flow parameters depends on the drivers' assessment. That is, their abilities to receive signals from other traffic participants about their manner of moving and the regime. The artificial intelligence hybrid Markovian ants in Queuing System has been applied in the traffic flow research in this paper. The driver's human intelligence has been substituted by Swarm intelligence. The analysed entropy of the pheromone signal among the ants in a column is analogue to the entropy of signals among successive vehicles in a traffic flow.

Keywords: Pheromone signal, probability, convolution, entropy.

1 Introduction

The phenomenon of the traffic flow has been the research subject for almost eighty years [1]. The traffic flow is defined by the means of its parameters differently interrelated and interconnected. These relations are conditioned by certain functional dependencies which are the result of the traffic flow nature, as a line of moving vehicles, but they are also conditioned by human indetermination and limitations, while steering. Since the first studies up till now, a car's performance, traffic arteries constructive features and signalization have kept developing. In accordance with the mentioned changes, drivers' education has also evolved, apart from the fact that they start being informed about traffic from the early age.

The importance of traffic flow problem solution is always a live issue for its social and economic implications. So far a large number of acceptable and non-contradictory models for defining some traffic flow parameter dependencies has been made feasible. Basically, there are two dominant concepts. These are the probabilistic concept and the one based on physics theory.

The first solutions were derived in the probabilistic concept [1,2]. The probabilistic concept is based on the inter-reaction of successive vehicles which are highly influenced by individual driver's behavior. A significant advancement has been achieved with a theoretical explanation of the application of Poisson processes in traffic flow [3]. The development of the probabilistic concept to the capital levels [4] still has no

answer to the high inter-dependence in the domain of conditional probability relations among all the drivers and vehicles in the traffic flow.

The solution to this complex question of high inter-relation of the vehicles in the flow were sought in analogies of the traffic flow and hydro-mechanical theories [5]. Since then, a new area of physics has been developed, the physics of traffic flow. The value of this analogy is permanently being proved by new knowledge of the nature of the traffic flow [6, 7, 8].

The difference between the probabilistic and physical approach to the description of traffic flow legitimacy is essential. The probabilistic approach is microscopic and dominantly refers to the driver and the vehicle. The traffic flow physical approach is macroscopic and dominantly refers to the road and capacity.

However, human behavior has a significant role in traffic flow. In addition, psycho-physical statuses of one driver are not constant values and they are liable to variations. Nevertheless, this human phenomenon cannot be expected in physical systems.

Human factor contains intelligence and therefore optimal reaction in complex situations. A large number of vehicles in traffic flow, traffic flow participants' behavior and autocatalytic traffic flow characteristics refer to a possible application of the concept Swarm intelligence. Swarm intelligence (SI) describes the collective behavior of decentralized, self-organized systems, natural or artificial. The concept SI so far has explained in multiple ways the application in traffic [9] and analogue research [10].

2 Markovian Process and Swarm Intelligence Synthesis

For traffic flow defining the following parameters are used [11, 12]: flow, speed, travelling time, occupying, density, interval gap, following distance, concentration. There are certain regularities among the stated parameters. These relations among the flow parameters are usually given for theoretically ideal and practically ideal conditions [13], which imply ignoring human factor influence. Probabilistic methods, in early papers, defined the basic regularities of the vehicles flow by the application of Poisson distribution [3].

The regular relation of flow participants demands the current situation estimation. In that regard, every next action or moving regime change depend on the present decision, not on the present conditions. Otherwise, it results in the degradation of the traffic safety. Owing to that, the logics of the regular traffic flow, with permanent and current participants' relations corresponds to the memory-less system. This logics is exclusively supported by exponential distribution as the only memory-less one.

Unlike theoretical, ideal conditions of traffic flow, in reality many conditions which define heterogeneous traffic flow structure have to be taken into consideration. A large number of variable drivers' characteristics and the vehicle and road features define the high indefinite individual driver's actions. Out of all theoretical distributions, the biggest indefiniteness can be described by the distribution with the largest entropy. That is exponential distribution.

The synthesis of traffic flow and Swarm intelligence analogies is led by Poisson distribution. If the time between the clients in a flow is exponentially distributed, that flow is complied with Poisson distribution, and the processes within the flow are

Markovian. The conditions of Ant algorithms convergence are lead by Poisson distribution [14]. Memory-less and entropy characteristics of exponential distribution are the feature of traffic flow. The stated conditions in multiple ways explain the synthesis of the traffic flow and Swarm intelligence. The microscopic concept is related to Markovian processes, while macroscopic are related to Swarm intelligence. In that way, the basics of the artificial intelligence hybrid is defined. The artificial intelligence hybrids of other species have already been applied in traffic flow studies [15].

3 The Function of Velocity in Palm's Stream

The basic consideration of stream events with the exponential time distribution among the clients $E(\lambda)$ is invariant of clients' velocity. The stream characteristics are usually studied by measuring successive time of clients' approaching taken at a referential position. Let us take into consideration moving of two proportional vehicles' flows, with the correlation coefficient c , $0 < c < 1$. If the client's velocity in the faster flow is constantly equal to $v^{fast} = v = const$, and the velocity in the slower flow is equal to the multiplication result of correlation coefficient c and the velocity v , $v^{slow} = cv$. In an ideal case of proportional distance between the clients $cd^f = d^s$, the period between the clients arrival is identical (Fig. 1).

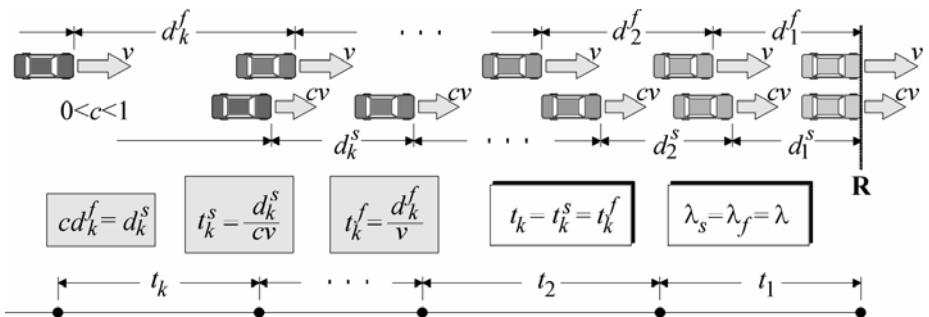


Fig. 1. Identical palm's streams with proportional client's velocity and distances

In Palm's stream, with exponentially distributed time $E(\lambda)$, the average time between two clients' arrival t_{ave} is equal to the reciprocal value of the flow intensity, $1/\lambda$. If the clients' velocity in the flow is homogeneous and $v=const$, then the average time between the clients' arrival can be described by the means of an equation which is related to the stream λ intensity, average distance in the stream d_{ave} and the clients' velocity in the stream v :

$$t_{ave} = \frac{d_{ave}}{v} = \frac{1}{\lambda} \Leftrightarrow d_{ave} = \frac{v}{\lambda}$$

In the real traffic flow, spacing d is the result of driver's estimation. The signal upon which the driver takes a certain distance depends on the evaluation of the possible safe stopping in case of need.

4 Markovian Ant Queueing System – MAQS

MAQS formation is taken from the assumption that the value of the pheromone signal emitted by the previous ant is a random variable of the exponential distribution $f(d)=\xi e^{-\xi d}$, $\xi \geq 0$. The distance between the previous and successive ant is d . The velocity of the previous ant which emits the pheromone signal is arbitrary and constant. The velocity of the successive ant is proportional to the value of the received signal. If the distance between the two successive ants is described as by integer variable x_d of the exponential function of pheromone signal $f(d)$, $X_d=INT(\xi e^{-\xi d})$, then the probability of the two successive ants distance equal to $P(X_d)$. The probability of unit distance X_k is equal to the integral of exponential distribution at the interval $(k, k+1)$ (Fig. 2).

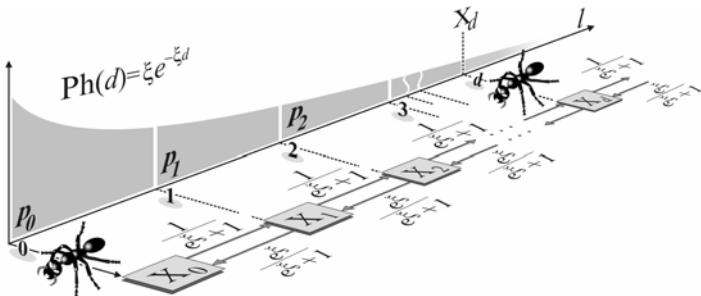


Fig. 2. Ant Queueing System Formation

$$P(X_k) = p_k = \int_k^{k+1} \xi e^{-\xi l} dl = e^{-k\xi} - e^{-(k+1)\xi} = \frac{1}{e^{k\xi}} - \frac{1}{e^{\xi} e^{k\xi}} = \frac{1}{e^{k\xi}} \left(\frac{e^{\xi} - 1}{e^{\xi}} \right)$$

The designed system is Markovian, with discrete conditions and continuous intensity of the pheromone signal. The distance between the two ants can be shown as queueing system. The system condition describes the value of the distance. The designed queueing system has characteristic structure (Fig. 3) [16].

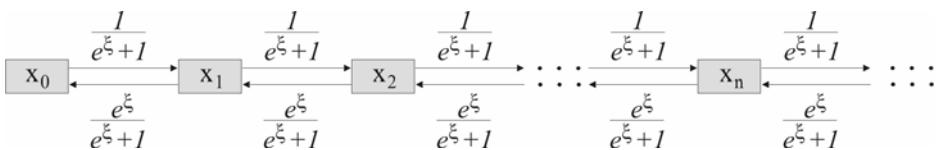


Fig. 3. Markovian Ant Queueing System

In the analogy of the traffic flow, the average distance between ants in MAQS is equal to the average distance between the vehicles in a line. The average distance between two ants is equal to the average number of clients in the system MAQS:

$$\begin{aligned} \sum_{k=0}^{\infty} kp_k &= 0\left(1 - \frac{1}{e^\xi}\right) + 1\left(\frac{1}{e^\xi} - \frac{1}{e^{2\xi}}\right) + \dots + k\left(\frac{1}{e^{k\xi}} - \frac{1}{e^{(k+1)\xi}}\right) + \dots = \frac{1}{e^\xi} + \frac{1}{e^{2\xi}} + \frac{1}{e^{3\xi}} + \dots = \\ &= \sum_{k=1}^{\infty} \left(\frac{1}{e}\right)^\xi = \sum_{k=0}^{\infty} \left(\frac{1}{e}\right)^\xi - 1 = \frac{1}{1 - \frac{1}{e^\xi}} - 1 = \frac{e^\xi}{e^\xi - 1} - 1 = \frac{1}{e^\xi - 1} \end{aligned}$$

5 Entropy of Signal ξ Parameters Functions

The concept MAQS is completely analogue to the vehicle moving in a traffic flow. Each successive car adjusts its velocity on the grounds of the estimation of the previous car's velocity.

The value $l/(e^\xi + 1)$ is the pheromone signal spreading intensity of the previous ant. In the analogy of traffic flow, it is the probability that the previous vehicle will emit the signal of its status in the sense of position (visibility without rear lights, rear lights on, turn signal for parking), velocity, the driving regime (braking- stop lights, turning, turn signals on),etc. According to the concept MAQS, the density of this intensity distribution is exponentially distributed, with the density and distribution function:

$$f(d) = \frac{1}{e^\xi + 1} e^{-d \frac{1}{e^\xi + 1}} ; F(d) = 1 - e^{-d \frac{1}{e^\xi + 1}}$$

The value $e^\xi/(e^\xi + 1)$ is the reaction intensity of the successive vehicle. In traffic flow analogy, that is the probability that the successive vehicle will adjust its velocity to the previous vehicle. According to the concept MAQS, the distribution density of this intensity is exponentially distributed, with the distribution density and distribution function:

$$f(d) = \frac{e^\xi}{e^\xi + 1} e^{-d \frac{e^\xi}{e^\xi + 1}} ; F(d) = 1 - e^{-d \frac{e^\xi}{e^\xi + 1}}$$

These probabilities are realized in the communication channel. In MAQS it is the medium for artificial pheromone transmission. In traffic flow, the information transmission about the status of each previous vehicle is realized on the basis of driver's estimation, optically. For the defined value of exponentially distributed signal of the previous vehicle in a stationary flow, the entropy of signal transmission from the previous vehicle to the successive one, is defined by the means of exponential distribution formula [17].

$$\bar{H}\left(E\left(\frac{1}{e^\xi + 1}\right)\right) = 1 - \ln\left(\frac{1}{e^\xi + 1}\right) = 1 - \ln(1) + \ln(e^\xi + 1) = 1 + \ln(e^\xi + 1)$$

The entropy of velocity adjustment of the successive vehicle in traffic flow is:

$$\bar{H}\left(E\left(\frac{e^\xi}{e^\xi + 1}\right)\right) = 1 - \ln\left(\frac{e^\xi}{e^\xi + 1}\right) = 1 - \ln(e^\xi) + \ln(e^\xi + 1) = 1 + \ln(e^\xi + 1) - \xi$$

The difference between the entropy of signal transmission to the successive vehicle and the velocity adjustment of the successive vehicle to the conditions in the line of vehicles is equal to:

$$\bar{H}\left(E\left(\frac{1}{e^\xi + 1}\right)\right) - \bar{H}\left(E\left(\frac{e^\xi}{e^\xi + 1}\right)\right) = \xi$$

The derived result is conceptually in accordance with the cars' moving conditions in traffic flow. The entropy difference is positive, which means that there is higher indefiniteness of the signal from the previous to the successive vehicle. Each successive vehicle moves in conditions of less entropy. Otherwise, the traffic flow moving would be based on accidental situations. This relation is a clear reflection of autocatalytic traffic flow characteristic, the same feature which could be met in Swarm Intelligence.

The difference of the stated informative indefiniteness results in informative definiteness which is equal to ξ . In MAQS the basic pheromone capacity is equal to ξ . The specific procedure of the successive vehicle is proportional to the basic capacity of the signal ξ emitted by the previous vehicle.

The signal relation between the vehicles in the line can be best illustrated by an example experienced by all average car drivers. In different driving conditions, at the same velocity and the same distance, the value of the signal received from the previous vehicle has different values for normal driving conditions, at the heavy rain, in fog and during night (Fig. 4). The highest value of the signal is in normal driving conditions, $\xi_1 \geq \xi_2, \xi_1 \geq \xi_3$ i $\xi_1 \geq \xi_4$.

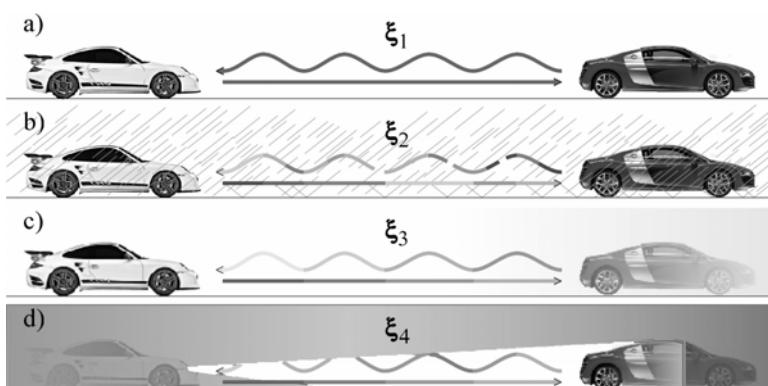


Fig. 4. Different conditions of communication channels between successive vehicles in traffic flow: a) normal, b) rainfall, c) fog, d) night

From the formula of the average distance between vehicles (1) and the average distance between ants in MAQS (3), a functional connection is derived, connection between the informative traffic flow definiteness, car's velocity in traffic flow and traffic flow intensity. The informative definiteness is at the same time the intensity of the basic parameter of the signal between cars in the flow. The signal has the domain.

$$d_{ave} = vt_{ave} = \frac{v}{\lambda} = \frac{1}{e^{\xi} - 1} \Leftrightarrow e^{\xi} - 1 = \frac{\lambda}{v} \Leftrightarrow \xi = \ln\left(\frac{\lambda}{v} + 1\right)$$

If the intensity $\lambda=0$, there are no vehicles in the traffic flow and no signal emission. The traffic flow does not exist. With the increase of the value $\lambda>0$, there is the signal value increase in traffic flow. Namely, with the increase of the value of traffic flow intensity, there is also the density increase, which is the measure of interaction. For the velocity value nought, $v=0$, $\lambda>0$, the maximum value of the signal can be received. If the column does not move, the definiteness is complete and there is no uncertainty in interactions among vehicles.

With the increase of velocity $v>0$, the signal value is decreased, which causes taking longer follow-up distances by the drivers. At high velocity, the theoretical value of the signal converges towards naught. The average distance between ants in a colony, and, by analogy, cars in traffic flow, in that case diverges.

$$\lim_{v \rightarrow \infty} \xi = \lim_{v \rightarrow \infty} \ln\left(\frac{\lambda}{v} + 1\right) = \ln(1) = 0, \quad \lim_{v \rightarrow \infty} \frac{1}{e^{\ln\left(\frac{\lambda}{v} + 1\right)} - 1} = \lim_{v \rightarrow \infty} \frac{1}{\left(\frac{\lambda}{v} + 1\right) - 1} = \infty$$

It means that in traffic flow conditions with high velocity, drivers of successive vehicles "lose the signal" from the previous vehicle and they have to adjust the gap time to the reaction time for safe stopping. However, with higher velocity, there an increase in entropy of the complete status of the previous vehicle in traffic flows. The signal divergence can be taken into account to the limits of sensor system perception, optical system with drivers with analogy of chemosensory system with ants.

At high velocity, the entropy can be decreased only by distance increasing. The phenomenon of this anticipation in human intelligence is in accordance with the results of artificial intelligence hybrid.

A simple entropic system, memoryless exponential processors with Ant Based memory system can be incorporated into the hybrid learning machines concept and its algorithms [18, 19] to increase traffic safety and capacity of roads.

6 Conclusion

The obtained results indisputably suggest that traffic flow is based on informative definiteness which is profiled by each successive driver in a column. The derived concept is completely in accordance with the conclusions confirmed in practice that the highest flow on a road can be achieved only in the domain of lower velocity and higher vehicle interaction, that is, increased signal and decreased follow-up gap. In a continuous line undefined by the signal intensity, velocity increase would lead to flow intensity, which is not the case in the traffic flow.

The artificial intelligence hybrid application is at the same time the synthesis of microscopic (probabilistic) and macroscopic (the flow's physics) concept of solving traffic flow complex problem. The logics of the first results justify further application of artificial intelligence hybrid in traffic flow research.

References

1. Kinzer, J.P.: Application of The Theory of Probability to Problems of Highway Traffic, B.C.E. Thesis, Polytechnic Institute of Brooklyn, Proceedings Institute Traffic Engineering (1933)
2. Adams, W.F.: Road Traffic Considered as A Random Series. *J. Institute Civil Engineering* 4 (1936)
3. Greenshields, B.D., Shapiro, D., Erickson, E.L.: Traffic Performance at Urban Street Intersections, Technical Report No. 1, Yale Bureau of Highway Traffic, New Haven, Conn. (1947)
4. Mahnkea, R., Kaupužsb, J., Lubashevskyc, I.: Probabilistic description of traffic flow. *Physics Reports* 408, 1–130 (2005)
5. Pipes, L.A.: Vehicle acceleration in the hydrodynamic theory of traffic flow. *Transp. Res.* 3, 229–234 (1969)
6. Sun, D.-H., et al.: Effect of looking backward on traffic flow in an extended multiple car-following model. *Physica A* (2010), doi:10.1016/j.physa.2010.10.016
7. Tang, T.Q., Huang, H.J., Zhao, S.G., Shang, H.Y.: A new dynamic model for heterogeneous traffic flow. *Physics Letters A* 373, 2461–2466 (2009)
8. Jin, S., Wang, D., Tao, P., Li, P.: Non-lane-based full velocity difference car following model. *Physica A: Statistical Mechanics and its Applications* 389, 4654–4662 (2010)
9. Teodorović, D.: Swarm intelligence systems for transportation engineering: Principles and applications. *Transportation Research Part C: Emerging Technologies* 16, 651–667 (2008)
10. John, A., Schadschneider, A., Chowdhury, D., Nishinari, K.: Characteristics of ant-inspired traffic flow. *Swarm Intelligence* 2, 25–41 (2008)
11. Traffic flow theory A state-of-the-Art Report, Revised 2001, Organized by the Committee on Traffic Flow Theory and Characteristics (AHB 45), pp. 2-1 (2001)
12. Kuzović, L., Bogdanović, V.: *Teorija saobraćajnog toka*, Fakultet tehničkih nauka, Novi Sad, p. 104 (2010)
13. Traffic flow theory A state-of-the-Art Report, Revised 2001, Organized by the Committee on Traffic Flow Theory and Characteristics (AHB 45), pp. 2-5 (2001)
14. Badr, A., Fahmy, A.: A proof of convergence for Ant algorithms. *Information Sciences* 160, 267–279 (2004)
15. Yin, H., Wong, S.C., Xu, J., Wong, C.K.: Urban traffic flow prediction using a fuzzy-neural approach. *Transportation Research Part C: Emerging Technologies* 10, 85–98 (2002)
16. Tanackov, I., Simić, D., Sremac, S., Tepić, J., Kocić-Tanackov, S.: Markovian ants in a queuing system. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 32–39. Springer, Heidelberg (2010)
17. Menendez, M.L.: Shannon's entropy in exponential families: Statistical applications. *Applied Mathematics Letters* 13, 37–42 (2000)
18. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72, 2729–2730 (2009)
19. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. *Information Science* 180, 2633–2634 (2010)

Diagnosis of Partial Discharge Using Self Organizing Maps and Hierarchical Clustering – An Approach

Rubén Jaramillo-Vacio^{1,2}, Alberto Ochoa-Zezzatti³, S. Jöns¹,
Sergio Ledezma-Orozco⁴, and Camelia Chira⁵

¹ CFE-LAPEM

² Centro de Innovación Aplicada en Tecnologías Competitivas (CIATEC)

³ Universidad Autónoma de Ciudad Juárez

⁴ Universidad de Guanajuato - División de Ingenierías

⁵ Babes-Bolyai University, Cluj-Napoca; Romania

ruben.jaramillo@cfel.gob.mx

Abstract. This paper shows a first approach in a diagnosis selecting the different features to classify measured of partial discharges (PD) activities into underlying insulation defects or source that generate PD. The results present different patterns using a hibrid method with Self Organizing Maps (SOM) and Hierarchical clustering, this combination constitutes an excellent tool for exploration analysis of massive data like partial discharge on underground power cables. The SOM has been used for nonlinear feature extraction. Therefore, the clustering method has been fast, robust, and visually efficient.

Keywords: Self Organizing Map, Hierarchical clustering, Partial Discharge, diagnosis.

1 Introduction

The partial discharge (PD) is a common phenomenon which occurs in insulation of high voltage, this definition is given in [1]. In general, the partial discharges are in consequence of local stress in the insulation or on the surface of the insulation. This phenomenon has a damaging effect on the equipments, for example transformers, power cables, switchgears, and others. The first approach in a diagnosis is selecting the different features to classify measured PD activities into underlying insulation defects or source that generate PD's. In particular for solid insulation like XLPE on power cables where a complete breakdown seriously damages the test object the partial discharge measurement is a tool for quality assessment [2].

In [3] two variations of data mining approaches are discussed using examples from different diagnostic measurements and tests. The use of Kohonen mapping applied to existing and new data is illustrated using examples of dissolved gas analysis, tap-changer monitoring and insulator testing. In addition, two other examples are given to illustrate him potential for data mining to produce useful information from comprehensive modern diagnostic monitoring of high voltage cables and of circuit breakers. A new task force within SC15 has been established to embark on a coordinated and focused approach to develop guidelines for data mining in practice.

In [4] proposes a method for the analysis of condition monitoring data, and demonstrates this method in its discovery of useful knowledge from trip coil data captured from a population of in-service distribution circuit breakers and empirical UHF data captured from laboratory experiments simulating partial discharge defects typically found in HV transformers. This discovered knowledge then forms the basis of two separate decision support systems for the condition assessment/defect classification of these respective plant items.

In this paper is shown a first approach in a diagnosis selecting the different features to classify measured PD activities into underlying insulation defects or source that generate PD's. Some different results that are presented of the systems on site and shows the different patterns using Self Organizing Maps (SOM) and Hierarchical clustering, and show how a combination of a method that reduces the dimensionality of dataset to a relatively small number of clusters like SOM with a hierarchical clustering technique provides a very efficient tool for exploratory analysis of partial discharge patterns. Multidimensional scaling (MDS) is a nonlinear feature extraction technique [5]. It aims to represent a multidimensional dataset in two or three dimensions such that the distance matrix in the original k -dimensional feature space is preserved as faithfully as possible in the projected space. The SOM, or Kohonen Map [6], can also be used for nonlinear feature extraction. It should be emphasized that the goal here is not to find an optimal clustering for the data but to get good insight into the cluster structure of the data for data mining purposes. Therefore, the clustering method should be fast, robust, and visually efficient.

2 Partial Discharge: Concepts

Partial discharges occur wherever the electrical field is higher than the breakdown field of an insulating medium:

- Air: 27 kV/cm (1bar)
- SF6: 360 kV/cm (4bar)
- Polymers: 4000 kV/cm

There are two necessary conditions for a partial discharge to occur in a cavity:

- presence of a starting electron to initiate an avalanche
- the electrical field must be higher than the ionization inception field of the insulating medium

They are generally divided into three different groups because of their different origins:

- **Corona Discharges** – Occurs in gases or liquids caused by concentrated electric fields at any sharp points on the electrodes.
- **Internal Discharges** – Occurs inside a cavity that is surrounded completely by insulation material; might be in the form of voids (e.g. dried out regions in oil impregnated paper-cables), delaminations (e.g. in laminated insulation) or cracks (e.g. fatigue stress). Treeing discharges (current pulses within an electrical tree) which may be caused by imperfections in the solid insulation.

- **Surface Discharges** – Occurs on the surface of an electrical insulation where the tangential field is high e.g. end windings of stator windings.

The charge that a PD generates in a cavity (Figure 1) is called the physical charge and the portion of the cavity surface that the PD affects is called the discharge area. E_{applied} is the applied electric field and q_{physical} is the physical charge [7].

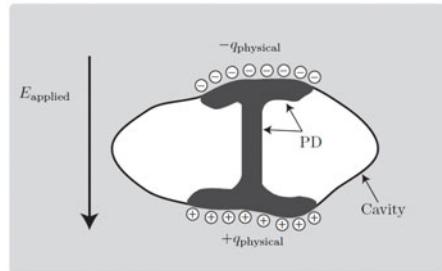


Fig. 1. Schematic picture of a partial discharge in a cavity

The pulse repetition rate n is given by the number of partial discharge pulses recorded in a selected time interval and the duration of this time interval. The recorded pulses should be above a certain limit, depending on the measuring system as well as on the noise level during the measurement. The pulse repetition frequency N is the number of partial discharge pulses per second in the case of equidistant pulses. Furthermore, the phase angle ϕ and the time of occurrence t_i are information on the partial discharge pulse in relation to the phase angle or time of the applied voltage with period T :

$$\phi_i = 360(t_i / T) \quad (1)$$

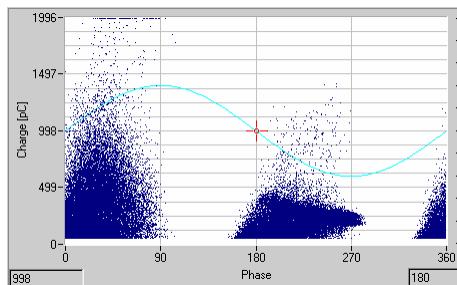


Fig. 2. Example of the partial discharge pattern of measurement

In the measurement equipment is recorded the number of the detected partial discharges with a certain combination of phase and charge (Figure 2). This graph shows the behavior of partial discharge in a cavity under high voltage rising.

In PD diagnosis test, it is very important to classify measured PD activities, since PD is a stochastic process, namely, the occurrence of PD depends on many factors, such as temperature, pressure, applied voltage and test duration; moreover PD signals contain noise and interference [8]. Therefore, the test engineer is responsible for choosing proper methods to diagnosis for the given problem. In order to choose the features, it is important to know the different source of PD, an alternative is though pattern recognition. This task can be challenging, nevertheless, features selection has been widely used in other field, such as data mining [9] and pattern recognition using neural networks [10]. This research only shows test on laboratory without environment noise source, and it is a condition that does not represent the conditions on site, in [11] was presented the noise levels on site based on previous experiences.

The phase resolved analysis investigates the PD pattern in relation to the variable frequency AC cycle. The voltage phase angle is divided into small equal windows.

The analysis aims to calculate the integrated parameters for each phase window and to plot them against the phase position ϕ .

- $(n - \phi)$: the total number of PD pulses detected in each phase window plotted against the phase position.
- $(q_a - \phi)$: the average discharge magnitude in each phase window plotted against the phase position ϕ , where q_a is average discharge magnitude.
- $(q_m - \phi)$: the peak discharge magnitude for each phase window plotted against ϕ , where q_m is peak discharge magnitude.

3 Self Organizing Maps (SOM) and Hierarchical Clustering

3.1 SOM

The Self Organizing Map developed by Kohonen, is the most popular neural network models. The SOM algorithm is based on unsupervised competitive learning, which means that the training in entirely data-driven and that the neurons of the map compete with each other [12].

Supervised algorithms, like multi-layered perceptron, required that the target values for each data vector are known, but the SOM does not have this limitation [10]. The SOM is a neural network model that implements a characteristics non-linear mapping from the high-dimensional space of input signal onto a typically 2-dimensional grid of neurons. The SOM is a two-layer neural network that consists of an input layer in a line and an output layer constructed of neurons in a two-dimensional grid as is shown in Figure 3.

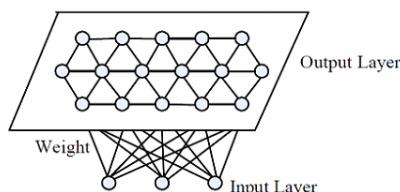


Fig. 3. System structure of SOM

The neighborhood relation of neuron i , an n -dimensional weight vector w is associated; n is the dimension of input vector. At each training step, an input vector \mathbf{x} is randomly selected and the Euclidean distances between \mathbf{x} and \mathbf{w} are computed.

The image of the input vector and the SOM grid is thus defined as the nearest unit w_{ik} and best-matching unit (BMU) whose weight vector is closest to the \mathbf{x} :

$$D(x, w_i) = \sqrt{\sum_k (w_{ik} - x_k)^2} \quad (2)$$

The weight vectors in the best-matching unit and its neighbors on the grid, are moved towards the input vector according the following rule:

$$\begin{aligned} \Delta w_{ij} &= \delta(c, i) \alpha(x_j - w_{ij}) \\ \Delta w_{ij} &= \alpha(x_j - w_{ij}) \text{ to } i = c \\ \Delta w_{ij} &\text{ to } i \neq c \end{aligned} \quad (3)$$

where c denote the neighborhood kernel around the best-matching unit and α is the learning rated and δ is the neighborhood function (Figure 4).

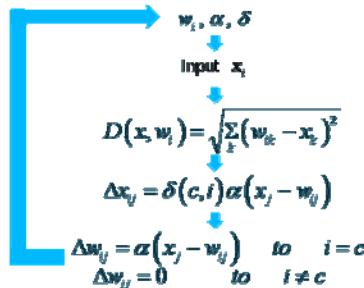


Fig. 4. Flow chart of SOM

The number of panels in the SOM is according the $A \times B$ neurons, the U-matrix representation is a matrix U ($(2A-1) \times (2B-1)$) dimensional [13].

3.2 Hierarchical Cluster Analysis

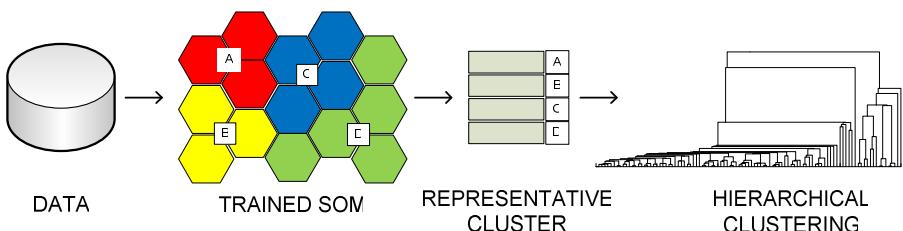
Cluster analysis or clustering, is to assign a set of observations into subsets so that observations in the same cluster are similar in some attribute. Clustering is a method of unsupervised learning, and a common technique in statistical data analysis used in many fields, for example data mining, pattern recognition, image analysis, information retrieval, and bioinformatics. A widely adopted definition of optimal clustering is a partitioning that minimizes distances within and maximizes distances among clusters. However, this leaves much room for variation: within- and between-clusters distances can be defined in several ways; see Table 1. The selection of the distance criterion depends on application. In this paper, Euclidean distance is used because it is widely worn with SOM [14].

Table 1. Within-Clusters distance $S(Q_k)$ and Between-clusters distance $d(Q_k, Q_l)$ [14]

Within-Clusters distance	$S(Q_k)$
Average distance	$S_a = \frac{\sum_{i,j} \ x_i - x_j\ }{N_k(N_k - 1)}$
Nearest neighbor distance	$S_{nn} = \frac{\sum_i \min_j \ x_i - x_j\ }{N_k}$
Centroid distance	$S_c = \frac{\sum_i \ x_i - c_k\ }{N_k}$
Between-clusters distance	$d(Q_k, Q_l)$
Single linkage	$d_s = \min_{i,j} \{ \ x_i - x_j\ \}$
Complete linkage	$d_{co} = \max_{i,j} \{ \ x_i - x_j\ \}$
Average linkage	$d_a = \frac{\sum_{i,j} \ x_i - x_j\ }{N_k N_l}$
Centroid linkage	$d_{ce} = \ c_k - c_l\ $

4 Analysis of PD Data

PD measurements for power cables are generated and recorded through laboratory tests. Corona was produced with a point to hemisphere configuration: needle at high voltage and hemispherical cup at ground. Surface discharge XLPE cable with no stress relief termination applied to the two ends. High voltage was applied to the cable inner conductor and the cable sheath was grounded, this produces discharges along the outer insulation surface at the cable ends. And internal discharge was used a power cable with a fault due to electrical treeing. Were considered the pattern characteristic of univariate phase-resolved distributions as inputs, the magnitude of PD is the most important input as it shows the level of danger, for this reason the input in the SOM the raw data is the peak discharge magnitude for each phase window plotted against ($qm - \phi$). In figure 5 shows the conceptual diagram training.

**Fig. 5.** The component interaction between SOM and Hierarchical clustering

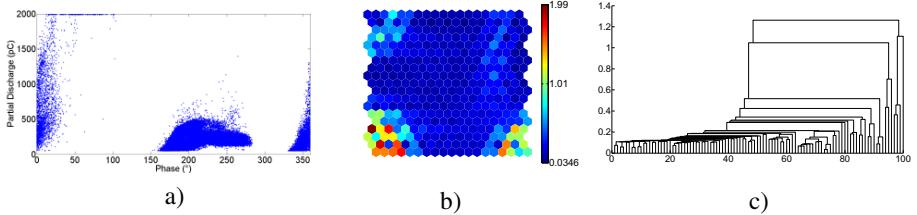


Fig. 6. Surface discharge: a) Scatter plot, b) U-matrix, c) Dendrogram

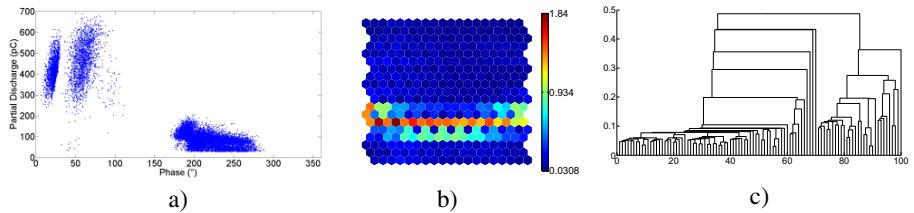


Fig. 7. Void discharge: a) Scatter plot, b) U-matrix, c) Dendrogram

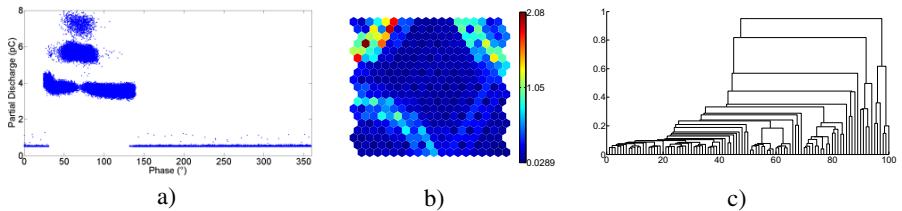


Fig. 8. Corona discharge point at ground: a) Scatter plot, b) U-matrix, c) Dendrogram

In the cases analyzed, the original dataset is 1 million of items, was used a U-matrix of 20×20 cells to extract features. The color of a hexagon represents the average distance between the estimated data vector at that position on the map and the neighboring hexagons. Blue color indicates that the data groups are “close”, or more similar, and red means that the data groups are “far”, or less similar. Therefore, we can identify clustering membership by locating red “walls” among data groups.

In figures 6a, 7a and 8a, is shown the scatter plot of $q_m - \phi$, but it is not possible to extract a pattern characteristic. In figures 6b, 7b and 8b, is shown the U-matrix, were the clusters are separates and it is possible to analyze patterns among clusters, in figures 6c, 7c and 8c. The final representation is the hierarchical tree, in which the branch length is proportional to the difference of the average expression patterns of the cluster under the bifurcation of the tree.

Were tested 63 dataset of diagnostic test at power cables, obtaining a very fast data representation and 95% confidence in the discrimination of partial discharge source, considering noise and combined sources.

5 Conclusions and Future Works

This paper shows the different patterns using Self Organizing Maps (SOM) and Hierarchical clustering methods, this combination constitutes an excellent tool for exploration analysis of massive data like partial discharge on power cables underground. The use of SOM followed by the application of Hierarchical Clustering methods constitutes a fast and reasonably accurate method for exploratory analysis and diagnosis within a large amount of data. For future work is possible to develop an analysis and diagnosis on underground power cables with rule-extraction algorithm like decision tree.

References

- [1] IEC 60270 Ed. 2. High-voltage test techniques - Partial discharge measurements (2000)
- [2] Wills, L.: Electrical Power Cable Engineering. Marcel Dekker, Inc., New York (1999)
- [3] McGrail, A.J., Gulski, E.: Data mining techniques to assess the condition of high voltage electrical plant. In: CIGRÉ (2002)
- [4] Strachan, S.M., Stephen, B.: Practical applications of data mining in plant monitoring and diagnostics. IEEE Power Engineering Society General Meeting (2007)
- [5] Kantardzic, M.: Data Mining; Concepts, Methods and Algorithms. Wiley, New York (2003)
- [6] Johnson, R.A., Wichern, E.W.: Applied Multivariate Statistical Analysis, 5th edn. Prentice-Hall, Englewood Cliffs (2002)
- [7] Forssén, C. Modelling of cavity partial discharges at variable applied frequency. Sweden: Doctoral Thesis in Electrical Systems. KTH Electrical Engineering (2008)
- [8] Edin, H.: Partial discharge studies with variable frequency of the applied voltage. Sweden: Doctoral Thesis in Electrical Systems. KTH Electrical Engineering (2001)
- [9] Lai, K., Phung, B.: Descriptive Data Mining of Partial Discharge using Decision Tree with genetic algorithms. In: AUPEC (2008)
- [10] Salama, M.M.A.: PD pattern recognition with neural networks using the multilayer perception technique. IEEE Transactions on Electrical Insulation 28, 1082–1089 (1993)
- [11] Markalous, S.: Detection and location of Partial Discharges in Power Transformers using acoustic and electromagnetic signals. Stuttgart University: PhD Thesis (2006)
- [12] Kohonen, T.: Engineering Applications of Self Organizing Map. Proceedings of the IEEE (1996)
- [13] Rubio-Sánchez, M.: Nuevos Métodos para Análisis Visual de Mapas Auto-organizativos. PhD Thesis. Madrid Politechnic University (2004)
- [14] Vesanto, J., Alhoniemi, E.: Clustering of the Self Organizing Map. IEEE Transactions on Neural Networks 11(3), 1082–1089 (2000)

Bayesian Segmentation of Magnetic Resonance Images Using the α -Stable Distribution

Diego Salas-Gonzalez¹, Matthias Schlögl¹, Juan M. Górriz¹,
Javier Ramírez¹, and Elmar Lang²

¹ Dpt. Signal Theory, Networking and Communications,
ETSIIT-UGR, University of Granada, 18071 Granada, Spain
`{dsalas,gorriz,javierrp}@ugr.es`

² Institute of Biophysics, University of Regensburg, 93040 Regensburg, Germany
`elmar.lang@biologie.uni-regensburg.de`

Abstract. In this work, a segmentation method of Magnetic Resonance images (MRI) is presented. On the one hand, the distribution of the grey (GM) and white matter (WM) are modelled using a mixture of α -stable distributions. A Bayesian α -stable mixture model for histogram data is used and the unknown parameters are sampled using the Metropolis-Hastings algorithm, therefore, voxel intensity information is included in the model via a parameterized mixture of α -stable distribution which allows us to calculate the likelihood. On the other hand, spatial information is also included: the images are registered to a common template and a prior probability is given to each intensity value using a normalized segmented tissue probability map. Both informations, likelihood and prior values, are combined using the Bayes' Rule. Performance of the segmentation approaches using spatial prior information, intensity values via the likelihood and combining both using the Bayes' Rule are compared. Better segmentation results are obtained when the latter is used.

Keywords: Magnetic Resonance Imaging, Image Segmentation, α -Stable Distribution, Mixture Model.

1 Introduction

The principal goal of the segmentation process is to partition an image into different regions which are homogeneous with respect to one or more features. Segmentation is an important tool in medical image processing and it has been useful in many applications [18][3][11]. In Magnetic Resonance Images (MRI), segmentation is performed to divide the entire image into subregions such as the White Matter (WM), Grey Matter (GM), and Cerebrospinal Fluid (CSF) spaces of the brain.

A wide range of segmentation techniques has been proposed. Nevertheless, there is not a procedure to perform segmentation which can be considered the best one, as they depend on type of image, application and they differ in the assumption about the type of images.

Mixture of Gaussians is a statistical model which has been widely applied in Magnetic Resonance brain image segmentation [29][17][10]. The goal is to fit the histogram of MRI intensity values using a mixture of two α -stable distributions considering the information given by the histogram data (frequency counts and bin locations).

The Bayesian α -stable mixture model has been previously developed in [13][5] for the symmetric case (skewness parameter $\beta = 0$) and in [14][12], for the general skewed α -stable distribution with 4 parameters. The Bayesian α -stable distribution which is used in this work is a modification of the work presented in [14] where a Bayesian finite mixture of α -stable distribution model was proposed. In this work, we use a mixture model for the case in which the observed data is given in form of histogram. α -stable distribution can be viewed as a generalization of the Gaussian mixture model, therefore, the proposed methodology could be applied instead of the Gaussian mixture model which is widely used as a previous step in many brain MR image segmentation approaches [29][17][10].

Hybrid intelligent systems are becoming popular due to their capabilities in handling many real world complex problems, involving imprecision, uncertainty, vagueness and high-dimensionality. They provide us with the opportunity to use both, our knowledge and raw data to solve problems in a promising way. This multidisciplinary research field is in continuous expansion in the artificial intelligence research community [5][4][1]. In this work, we combine some existing methods used in artificial intelligence as Bayesian parameter estimation, Monte Carlo methods (Gibbs sampling/Metropolis sampling) and mixture modelling. We also propose some modification of existing methods: α -stable mixture modelling for histogram data in order to build a multidisciplinary approach to perform brain image segmentation of magnetic resonance images.

2 Material and Methods

2.1 MRI Database

18 brain magnetic resonance images from the Internet Brain Segmentation Repository (IBSR) were selected because manual segmentation of the database is also available. This segmentation is the result of semi-automated segmentation techniques which require many hours of efforts by a trained expert, even in that case these segmentations can not be considered 100% ‘ground truth’, nevertheless they are a good starting point for testing the results obtained using the Bayesian α -stable mixture model approach. Thus, we use the information given by manually segmented images in order to check the performance of the proposed methodology. Furthermore, this dataset has been widely used in the literature [6][7].

2.2 Mixture of α -Stable Likelihood

The α -stable probability density function $f_{\alpha,\beta}(y|\gamma, \delta)$ has four parameters: $\alpha \in (0, 2]$ is the characteristic exponent which sets the level of impulsiveness, $\beta \in [-1, +1]$ is the skewness parameter, ($\beta = 0$, for symmetric distributions and

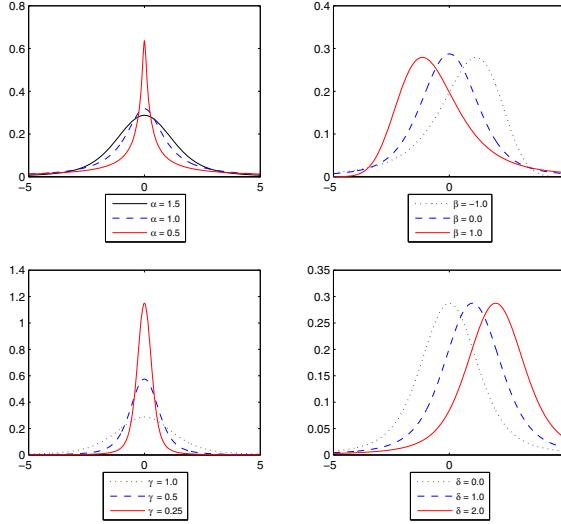


Fig. 1. α -stable probability density function with reference parameters $\alpha = 1.5$, $\beta = 0$, $\gamma = 1$ and $\delta = 0$ with changing: (a) Characteristic exponent α . (b) Skewness parameter β . (c) Dispersion γ . (d) Location parameter δ .

$\beta = \pm 1$ for the positive/negative stable family respectively), $\gamma > 0$ is the scale parameter, also called dispersion, and δ is the location parameter.

Figure 1 shows the α -stable probability density function for different values of the parameters. We use the distribution with parameters $\alpha = 1.5$, $\beta = 0$, $\gamma = 1$ and $\delta = 0$ as reference. This figure also explain the name of the parameters: α controls the degree of impulsiveness. When α decreases, the degree of impulsiveness increases. β controls the skewness and its sign, if the asymmetry is on the left or the right. γ controls the concentration of the samples along the bulk of the distribution. Lower values of γ correspond with higher concentration of the samples. Lastly, different values of γ produce the same probability density function but shifted in the x-axis.

The mixture of α -stables probability density functions $f_{\alpha,\beta}(y|\gamma, \delta)$ is given by:

$$p_Y(y) = \sum_{j=1}^k w_j f_{\alpha_j, \beta_j}(y|\gamma_j, \delta_j), \quad (1)$$

$$0 \leq w_j \leq 1 (\forall j) \text{ and } \sum_{j=1}^k w_j = 1, \quad (2)$$

where w_j is the mixture proportion or weight for component j and $p_Y(y)$ is the probability density function (pdf) of the data vector y .

Due to the great number of voxels in MR images, it is not computationally efficient to consider the model described in Equation (1) as is. Therefore instead

of using intensity values of voxels as vector observation we propose to consider an α -stable mixture model for histogram data (similar to the presented in [16] for Gaussian Mixture Model in the context of classification of functional brain images). Thus, histogram of MR intensity values can be fitted using a mixture model even when number of voxels is very large (typically $\sim 5 \cdot 10^5$ samples).

In the case in which the observed data is given in form of a histogram, the maximum likelihood estimation can also be used in a modified way [16, 8]. Let h_i be the height of each histogram bin (frequency counts) and x_i the bin locations, the likelihood can be expressed as

$$L(\theta|x) = \prod_{i=1}^{N_B} [p(x_i|\theta)]^{h_i}, \quad (3)$$

where N_B is the number of histogram bars and θ denotes the unknown parameters of the model. The total number of observations in that case is equal to the sum of the histogram heights, $N = \sum_{i=1}^{N_B} h_i$, therefore the complexity and computational burden of the model reduces considerably (from $N \sim 5 \cdot 10^5$ samples to $N_B \sim 50$ bin locations).

We infer on the unknown parameters of the α -stable mixture model for histogram data using the Metropolis-Hastings sampling method. See [15, 14] for more details.

Therefore, once the α -stable mixture parameters have been estimated, it is possible to calculate for each image j the likelihood for a given intensity value I : the probability that this voxel i belongs to White Matter $p_{j,i}(I|WM)$ or Grey

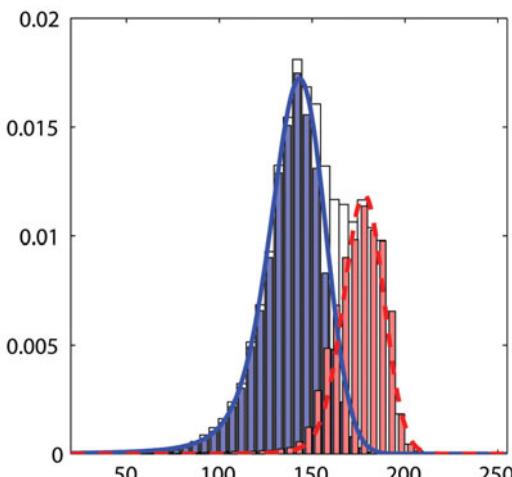


Fig. 2. Histograms with the intensity values of a sample Magnetic Resonance image. Blue bars: histogram of Grey Matter segmented manually. Red bars: histogram of White Matter segmented manually. The predicted two-component α -Stable mixture density is also plotted. Continuous line: first α -Stable component. Dotted line: second α -Stable component.

Matter $p_{j,i}(I|GM)$. Figure 2 shows the histogram of intensity values and the predicted α -stable densities for a given image of the dataset used to estimate the likelihood.

2.3 Spatial Prior Information

Manual segmentation is available for the MRI database used in this work. Using this information, we have built a segmented tissue probability map registering each manually segmented images to a T1 MRI template. Then, for a given image j , the segmented tissue probability map is calculated following a leave-one-out methodology: all the images except the j th are used to calculate the relative frequency of occurrence of white, grey or cerebro spinal fluid tissue. This process is repeated for each image in the database. Therefore, we obtain $p_{j,i}(WM)$, $p_{j,i}(GM)$ and $p_{j,i}(CSF)$ which denote the probability that voxel i of an image j belongs to white matter, grey matter or cerebrospinal fluid, respectively. Let note that, after spatial normalization, the same voxel position i in different images j refer to the same anatomical positions in the brain and, therefore, this renders possible to perform the calculation of frequency of occurrence of different brain tissue for a dataset of magnetic resonance images.

2.4 Bayesian Rule: Posterior Brain Image

Once the α -stable likelihood and the spatial tissue probability prior are estimated for each image and intensity values, we combine both information using the Bayes' rule:

$$p_{j,i}(WM|I) \propto p_{j,i}(I|WM)p_{j,i}(WM), \quad (4)$$

$$p_{j,i}(GM|I) \propto p_{j,i}(I|GM)p_{j,i}(GM), \quad (5)$$

and allocate the tissue of a given voxel i to WM if $p_{j,i}(WM|I) > p_{j,i}(GM|I)$, or otherwise to GM.

3 Results

Three methods of segmentation of images are tested using 18 MR images from the IBSR repository. Figure 3 plots the accuracy rate for each image in the case in which only white matter is considered, grey matter or the sum of white and grey matter voxels. In general, using only spatial prior information leads to lower performance of the segmentation approach. Furthermore, using the three methods studied in this work, Grey matter is segmented more accurately than White matter tissue. This figure also shows that combining spatial prior information and α -stable likelihood leads to an improvement of the segmentation process. Specifically, an accuracy rate up to 90% is obtained for some of the images under study.

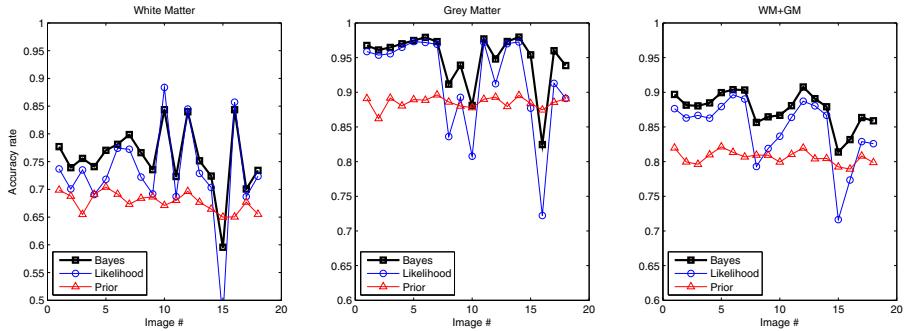


Fig. 3. Segmentation accuracy. *Left:* White Matter. *Center:* Grey Matter. *Right:* White and Grey Matter.

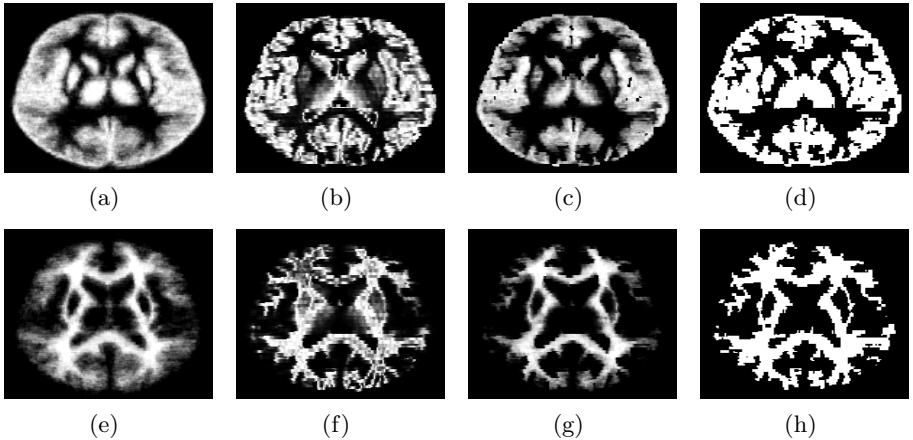


Fig. 4. Transaxial slice of one MR brain images used in this work. *First row:* Grey Matter. *Second row:* White Matter. Ranging from probability $p = 0$ to $p = 1$ using a linear grayscale colormap, *First column:* Prior probability map. *Second column:* α -stable likelihood. *Third column:* Bayesian posterior probability. *Fourth column:* Manually segmented tissue.

Figure 4 shows a transaxial slices of one brain image used in this work. First row represents, for each voxel, the probability to belong to grey Matter. A linear grayscale map is used where, for white voxels, probability $p = 1$ and for black voxels, $p = 0$ are chosen. First column plots the prior probability map. Second column is the α -stable likelihood: the probability to belongs to Grey Matter (the first component of the α -stable mixture model). White voxels, $p_{j,i}(I|GM) = 1$ and black voxels, $p_{i,j}(I|GM) = 0$. Third column is the posterior probability $p_{j,i}(GM|I)$ calculated using the Bayes' rule and, lastly, in fourth column white voxels are those which were considered as Grey Matter by a manual segmentation

Table 1. Accuracy rate obtained in the segmentation of MRI using the three methods under study

	Method	Mean Accuracy Rate	Error
WM	Likelihood	0.73	0.09
	Prior	0.67	0.02
	Posterior	0.76	0.06
GM	Likelihood	0.92	0.07
	Prior	0.89	0.01
	Posterior	0.95	0.04
WM+GM	Likelihood	0.85	0.05
	Prior	0.81	0.01
	Posterior	0.88	0.03

procedure. Analogously, second row depicts the same information but related to White matter brain tissue instead of Grey matter.

Table 1 shows the mean accuracy values and the error calculated using the standard deviation for the 18 images considered. It can be seen that better segmentation performance is obtained when we combine intensity information using an α -stable mixture parametrization of the histogram of intensity values along with spatial prior information calculating the posterior image using the Bayes' rule.

4 Conclusions

In this work, a segmentation approach combining mixture of α -stable modelling and segmented tissue probability map via Bayes' rule is proposed. The performance of the methodology is tested in a set of 18 MRI images. Results show that combining spatial prior information and α -stable likelihood outperforms the results obtained using only spatial information or mixture of α -stable likelihood. Specifically, an accuracy rate ranging from 0.81 to 0.91 are obtained when white matter and grey matter tissue are segmented for the MRI database considered.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72(13–15), 2729–2730 (2009)
2. Ashburner, J., Friston, K.: Unified segmentation. *NeuroImage* 26, 839–851 (2005)
3. Clarke, L.R., Velthuizen, R.R., Camacho, M.A., Heine, J.J., Vaidyanathan, M., Hall, L.O., Thatcher, R.W., Silbiger, M.L.: Mri segmentation: methods and applications. *Magnetic Resonance Imaging* 13(3), 334–368 (1995)
4. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. *Information Sciences* 180(14), 2633–2634 (2010)
5. Derrac, J., Garca, S., Herrera, F.: Algorithms for instance and feature selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)

6. García-Sebastián, M., Fernández, E., Grana, M., Torrealdea, F.J.: A parametric gradient descent mri intensity inhomogeneity correction algorithm. *Pattern Recognition Letters* 28(13), 1657–1666 (2007)
7. García-Sebastián, M., González, A.I., Graña, M.G.: An adaptive field rule for non-parametric mri intensity inhomogeneity estimation algorithm. *Neurocomputing* 72, 3556–3569 (2009)
8. Górriz, J.M., Lassl, A., Ramírez, J., Salas-Gonzalez, D., Puntonet, C.G., Lang, E.W.: Automatic selection of rois in functional imaging using gaussian mixture models. *Neuroscience Letters* 460(2), 108–111 (2009)
9. Greenspan, H., Ruf, A., Goldberger, J.: Constrained gaussian mixture model framework for automatic segmentation of mr brain images. *IEEE Transactions on Medical Imaging* 25(9), 1233–1245 (2006)
10. Merisaari, H., Parkkola, R., Alhoniemi, E., Teräs, M., Lehtonen, L., Haataja, L., Lapinleimu, H., Nevalainen, O.S.: Gaussian mixture model-based segmentation of mr images taken from premature infant brains. *Journal of Neuroscience Methods* 182(1), 110–122 (2009)
11. Pham, D.L., Xu, C., Prince, J.L.: Current methods in medical image segmentation. *Annual Review of Biomedical Engineering* 2(1), 315–337 (2000)
12. Salas-Gonzalez, D., Kuruoglu, E.E., Ruiz, D.P.: Bayesian estimation of mixtures of skewed alpha stable distributions with an unknown number of components. In: *Proceedings of the 14th European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, September 4-8 (2006)
13. Salas-Gonzalez, D., Kuruoglu, E.E., Ruiz, D.P.: Estimation of mixtures of symmetric alpha stable distributions with an unknown number of components. In: *Proceedings. 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006*, France, May 14-19, pp. 545–548 (2006)
14. Salas-Gonzalez, D., Kuruoglu, E.E., Ruiz, D.P.: Finite mixture of stable distributions. *Digital Signal Processing* 19(2), 250–264 (2009)
15. Salas-Gonzalez, D., Kuruoglu, E.E., Ruiz, D.P.: Modelling with mixture of symmetric stable distributions using gibbs sampling. *Signal Processing* 90(3), 774–783 (2010)
16. Segovia, F., Górriz, J.M., Ramírez, J., Salas-Gonzalez, D., Álvarez, I., López, M., Chaves, R., Padilla, P.: Classification of functional brain images using a gmm-based multi-variate approach. *Neuroscience Letters* 474(1), 58–62 (2010)
17. da Silva, A.R.F.: Bayesian mixture models of variable dimension for image segmentation. *Computer Methods and Programs in Biomedicine* 94, 1–14 (2009)
18. Suetens, P., Bellon, E., Vandermeulen, D., Smet, M., Marchal, G., Nuyts, J., Mortelman, L.: Image segmentation: methods and applications in diagnostic radiology and nuclear medicine. *European Journal of Radiology* 17, 14–21 (1993)

On-Line Valuation of Residential Premises with Evolving Fuzzy Models

Edwin Lughofer¹, Bogdan Trawiński², Krzysztof Trawiński³, and Tadeusz Lasota⁴

¹ Johannes Kepler University Linz, Department of Knowledge-Based Mathematical Systems,
Altenbergerstrasse 69, A-4040 Linz, Austria

² Wrocław University of Technology, Institute of Informatics,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

³ European Centre for Soft Computing, Edificio Científico-Tecnológico, 3^a Planta,
C. Gonzalo Gutiérrez Quirós S/N, 33600 Mieres, Asturias, Spain

⁴ Wrocław University of Environmental and Life Sciences, Dept. of Spatial Management
Ul. Norwida 25/27, 50-375 Wrocław, Poland

edwin.lughofer@jku.at, bogdan.trawinski@pwr.wroc.pl,
krzysztof.trawinski@softcomputing.es, tadeusz.lasota@wp.pl

Abstract. In this paper, we investigate on-line fuzzy modeling for predicting the prices of residential premises using the concept of evolving fuzzy models. These combine the aspects of incrementally updating the parameters and expanding the inner structure on demand with the concepts of uncertainty modeling in a possibilistic and linguistic manner (achieved through fuzzy sets and fuzzy rule bases). We use the FLEXFIS approach as learning engine for evolving fuzzy (regression) models, exploiting the Takagi-Sugeno fuzzy model architecture. The comparison with state-of-the-art expert-based premise estimation was based on a real-world data set including prices for residential premises within the years 1998 to 2008, and showed that FLEXFIS was able to outperform expert-based method.

Keywords: evolving fuzzy systems, on-line fuzzy modeling, property valuation, expert-based premise estimation.

1 Introduction

In nowadays sales comparison approaches for residential premises, it is necessary to have transaction prices of the properties sold whose attributes are similar to the one being appraised. If good comparable transactions are available, then it is possible to obtain reliable estimates for the prices of the residential premises. Prior to the evaluation, the appraiser must conduct a thorough study of the appraised property using available sources of information such as cadastral systems, transaction registers, performing market analyses, making an on-site inspection. The estimations are usually subjective and are based on human experience and intuition rather than on objective data. One of the most common state-of-the art approach of human estimations (conducted by many experts) is based on sliding windows, nearest neighbors elicitation and averaging of past prices to estimate a new price. Automated valuation models

(AVMs) were also proposed, which are based on statistical models such as multiple regression analysis [19], soft computing and geographic information systems (GIS) [25]. Many intelligent methods have been developed to support appraisers' works: neural networks [24], fuzzy systems [9] or case-based reasoning [21]. If all the data would be ordered by the transaction date, they would constitute some form of a data stream which in turn could reflect the changes of real estate market in the course of time. This motivated us also to use evolving models, which are able to process streams of data and to learn, update, expand their memory on-line on demand.

We investigate data-driven fuzzy rule-based systems (FRBS) as appropriate model architecture to build reliable models in the field of residential premises. Data-driven fuzzy systems in particular were recognized as the ones being able of approximating any real continuous function on a compact set with an arbitrary accuracy. The prove of this statement was presented in [7] as well as in [14]. Furthermore, FRBSs have capability of knowledge ex-traction and representation when modeling complex systems in a way that they could be understood by a human being. Interpretability of fuzzy systems is a characteristic that definitely favors this type of models [1–3] [6], as it is often necessary to understand the behavior of the given model. Another important point is that data-driven fuzzy systems can be permanently updated on demand based on new incoming samples as is the case for on-line measurements or data streams. Technologies for providing such updates with high performance (both in computational times and predictive accuracy) are called *evolving fuzzy systems* [15]. This was mainly because of the increasing amount of data and the requirement to have models at hand which are up-to-date by permanently including dynamically changing operating and environmental conditions, new system states or new class labels (in case of evolving fuzzy classifiers [4]).

In this paper, we exploit the so-called FLEXFIS method [17] as specific evolving fuzzy systems approach which incrementally evolves clusters (which are associated with rules) and performs a recursive adaptation of consequent parameters by using local learning approach. FLEXFIS can cope with both, batch (off-line) mode and incremental on-line mode when applying it to data streams: the learning phase activates and updates the model whenever a sample is sent, while requiring low resources.

Hybrid Artificial Intelligence Systems (HAIS) became a very powerful tool, when applying to the real-world problems. They were successfully applied in the field of shopping assistance [8], learning machines [10], intelligent robotics [11], and intrusion detection [12]. In this work, FLEXFIS, which combines clustering with least squares regression to generate fuzzy model, is considered as the HAIS model.

2 Fuzzy Model Architecture

Most of the conventional EFS approaches used for modeling residential premises rely on one common model architecture, the so-called Takagi-Sugeno (TS) fuzzy systems [22], which in a specific form (called fuzzy basis function networks [23]) are defined

$$\hat{f}(x) = \hat{y} = \sum_{i=1}^C l_i \psi_i(x) . \quad (1)$$

with the normalized membership functions:

$$\psi_i(x) = \frac{e^{-\frac{1}{2} \sum_{j=1}^p \frac{(x_j - c_{ij})^2}{\sigma_{ij}^2}}}{\sum_{k=1}^C e^{-\frac{1}{2} \sum_{j=1}^p \frac{(x_j - c_{kj})^2}{\sigma_{kj}^2}}} . \quad (2)$$

and consequent functions:

$$l_i = w_{i0} + w_{i1}x_1 + w_{i2}x_2 + \dots + w_{ip}x_p . \quad (3)$$

The symbol x_j denotes the j -th input variable (static or dynamically time-delayed), c_{ij} the center and σ_{ij} the width of the Gaussian fuzzy set in the j -th premise part of the i -th rule. As conjunction operator, the product t-norm is applied [13].

These type of fuzzy systems become very popular during the last two decades, especially in data-driven design and control applications.

3 Fuzzy Modeling in On-Line Mode with Evolving Fuzzy Systems

In the following, we present *FLEXFIS* approach as one of the most widely used EFS approaches in order to cope with the dynamics of (on-line) streaming data over time and applied to the prediction of residential premises in the evaluation section.

3.1 Basic Concepts

The FLEXFIS approach, short for FLEXible Fuzzy Inference Systems was first introduced in [18] and significantly extended version in [17], and designed for the purpose of incremental learning of TS fuzzy systems from data streams in a sample-wise single-pass manner. This means that always one sample can be loaded, sent into the FLEXFIS learning engine where the model is updated and immediately discarded, afterwards. In this sense, the method needs low resources 1.) with respect to computational complexity and 2.) with respect to virtual memory and hence is feasible for on-line modeling applications, where models should be kept-up-to-date as early as possible in order to account for new operating conditions, systems states etc. The basic steps in FLEXFIS approach can be summarized as follows:

1. Rule evolution and updating of antecedent parameters in the cluster space with the help of an incremental evolving clustering variant. The vigilance parameter ρ steers the tradeoff between plasticity and stability of the algorithm as it represents a decision threshold between rule evolution and update based on the distance of a new sample to the current cluster partition (see Step 4 in Algorithm 1).
2. Recursive adaptation of consequent parameters exploiting the local learning approach (parameters are updated for each rule separately).
3. Balancing out a non-optimal situation by adding a correction vector to the vector of consequent parameters.
4. In the extended version (Step 11 in Algorithm 1): Detecting redundant rules with the help of specific overlap measures (two variants: one-dimensional

intersection points of fuzzy sets and inclusion metric) and performing on-line rule merging/pruning after each incremental update cycle.

The steps in FLEXFIS are summarized in Algorithm 1.

3.2 The Algorithm

Building the aspects as discussed in the previous subsections together, leads to the rough (pseudo) algorithm of FLEXFIS as defined in Algorithm 1.

Algorithm 1 FLEXFIS+ (FLEXible Fuzzy Inference Systems from Data Streams)

1. Perform an initial training in batch mode (on pre-collected training samples or first on-line samples) and obtain the optimal value for the vigilance parameter ρ (e.g. within grid search); estimate the (initial) ranges of all features; [optional the training can be conducted from scratch with initial ranges of features pre-defined or extracted from first on-line samples]
2. For each new incoming data sample x do

Clustering Phase:

3. Normalize x to $[0,1]$ and the cluster centres according to the estimated feature ranges in Step 1.
4. If $\|x - c_{win}\| > \rho$ with c_{win} the centre coordinates of the nearest cluster, then evolve a new rule by increasing the number of rules $C = C + 1$ and setting its center to the current data sample $c_C = x$ and its range of influence in each direction by $\sigma_C = 0$.
5. Else Update the center of the nearest cluster c_{win} by $c_{win} = c_{win} + \eta(x - c_{win})$ with η a decreasing learning gain, and its range of influence by recursive variance formula including rank-one modification [20].
6. Transfer the clusters back to original feature space, according to the ranges of the features.

End of Clustering Phase

7. Project modified/evolved cluster to the axes in order to update/evolve the fuzzy set partition in each dimension (feature): the centres of the fuzzy sets are associated with the corresponding centre coordinates of the clusters, the widths of the fuzzy sets are set to $\max(\sigma, \epsilon)$ with ϵ a small positive constant, in order to avoid numerical instabilities.
8. Add correction vectors to the linear consequent parameters and correction matrices to the inverse Hessian matrices estimated in the previous step.
9. Perform recursive fuzzily weighted least squares for all C rules.
10. Update the ranges of all features.
11. Perform merging of similar rules.

4 Experimental Setup and Results

The data set used in experiments was drawn out from a rough data set containing above 50 000 records referring to residential premises transactions accomplished in one Polish big city with the population of 640 000 within eleven years from 1998 to 2008. In this period most transactions were made with non-market prices when the council was selling flats to their current tenants on preferential terms. Nine following features were pointed out as main drivers of premises prices: usable area of premises, age of a building, number of rooms in a flat, floor on which a flat is located, number of storeys in a building, geodetic coordinates Xc and Yc of a building, distance from the city centre and distance from the nearest shopping centre. From these nine features, the following five quantitative features were selected as the main price drivers by the experts (also used in the expert-based premise estimation): Area, Age, Storeys, Rooms, and Centre. Next, the data set was cleaned by removing items with lacking or erroneous values of attributes. The final data set counted 5303 records and comprised all premises which values could be estimated by the experts. Due to the fact that the prices of premises change substantially in the course of time, the whole 11-year data set cannot be used to create data-driven models using machine learning in off-line mode. The sizes of the year data sets are given in Table 1.

Table 1. Number of instances in half-year datasets

Dataset	#Inst	Dataset	#Inst
1998	269	2003	653
1999	477	2004	546
2000	329	2005	580
2001	463	2006	677
2002	530	2007	575
		2008	204

The experiments conducted for the on-line phase are based on a streaming data context which are simulated by using data of consecutive years joined together as they were recorded. This means that in the incremental learning context the same order of the data samples is supported as the temporal order in which the samples were stored onto hard-disc, yielding a one-to-one simulation of the real on-line case. In order to achieve a data stream with significant length, we joined five consecutive years to form one training data set (e.g. Years 1998 to 2002 to form a data set of 2068 samples) and used the subsequent year as separate test data set in order to verify the predictive power of the models. The one-year window for the test data set is fixed in this way, as also the expert-based premise estimation relies on the same window size.

The heuristic expert-based premise estimation method is summarized in Algorithm 2, where the premises located in each zone were classified into 243 groups determined by the most important features Area, Age, Storeys, Rooms, and Centre selected as the main price drivers. The six zones were: 1.) The central one, 2.) Near-medium, 3.) Eastern-medium, 4.) Western-medium, 5.) South-western-distant and 6.) North-eastern-distant.

Algorithm 2. Expert-Based House Price Estimation Method

1. Take the next premises to estimate.
2. Check the completeness of values of all five features and note a transaction date.
3. Select all premises sold earlier than the one being appraised, within current and one preceding year and assigned to the same group.
4. If there are at least three such premises calculate the average price taking the prices updated for the last day of a given transaction year.
5. Return this average as the estimated value of the premises.
6. Repeat steps 1 to 5 for all premises to be appraised.
7. For all premises not satisfying the condition determined in Step 4 extend the quality zones by merging 1 & 2, 3 & 4, and 5 & 6 zones. Moreover, extend the time window to include current and two preceding years.
8. Repeat steps 1 to 5 for all remaining premises.

Table 2 presents the mean absolute errors between predicted and actual premise premises on separate test data sets as well in a periodic hold-out manner (indicated in the first column). From this table someone can realize that

1. FLEXFIS is able to out-perform state-of-the-art expert-based estimation for all data sets – the outperformance is significant with a p-value of 0.0033 obtained from a pairwise Wilcoxon signed rank test
2. Including the pruning option in FLEXFIS achieves a lower complexity of the final evolved models in terms of the number of rules while keeping the error rates at a similar level. The pruning is also essential for interpretable fuzzy partitions in the evolved fuzzy models as shown in Figure 1 (without the pruning option these partitions appear quite weird with a lot of significantly over-lapping fuzzy sets).

Table 2. Mean absolute errors (MAEs) on separate test sets (before the slashes) and on periodic held-out test in the training stream (after the slashes) and number of rules obtained when using FLEXFIS with and without removing local, in the middle column the MAEs of the expert-based predictions are reported

Tr. / Test	FLEXFIS conv.	FLEXFIS + pruning	Expert	# Rules	# Rules with pruning
1998-2002 / 2003	0.049 0.044	/ 0.049 0.043	0.078	9	9
1999-2003 / 2004	0.072 0.039	/ 0.074 0.039	0.106	12	10
2000-2004 / 2005	0.072 0.049	/ 0.068 0.049	0.121	12	9
2001 – 2005 / 2006	0.130 0.051	/ 0.125 0.052	0.134	10	8
2002 – 2006 / 2007	0.089 0.058	/ 0.089 0.059	0.138	14	10
2003 – 2007 / 2008	0.110 0.060	/ 0.115 0.055	0.145	13	7

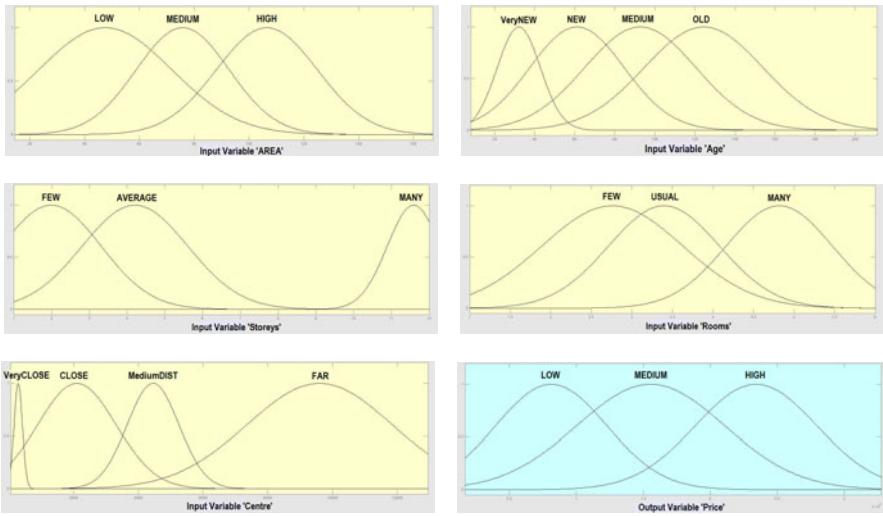


Fig. 1. Fuzzy partitions for the five input variables (a) to (e): Area, Age, Storeys, Rooms, Centre and the output variable Price (f)

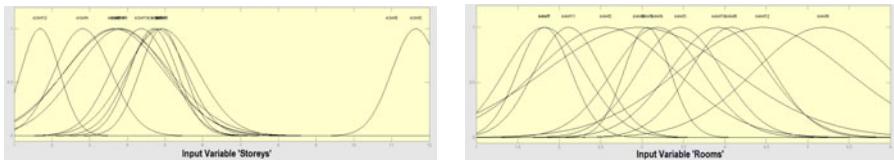


Fig. 2. Fuzzy partitions for the two input variables Storeys and Rooms when not using any merging/pruning option in FLEXFIS – compare with those in Figure 1

5 Conclusions and Future Work

In this paper, we investigated the usage of incremental data-driven evolving fuzzy modelling technique FLEXFIS for the purpose to estimate the future prices of residential premises based on past recordings. A comprehensive data set including over 5300 samples and recorded during the time span 1998 to 2008 served as basis for benchmarking our method versus a state-of-the-art expert-based estimation methods, widely used by professional appraisers due to its simplicity and the fact that many of them do not have enough skills to use statistical methods based on the least squares or polynomial regression which can provide more precise estimations. It turned out that FLEXFIS can significantly out-perform expert-based estimation. Based on the fact that our EFS method produces models which are allowing (at least some) linguistic interpretation of the modeled dependencies (as could be verified when applying rule merging/pruning techniques), it can be in fact seen as a reliable and powerful alternative for valuation of residential premises.

Acknowledgements. This work was funded partially by the Austrian fund for promoting scientific research (FWF, contract number I328-N23, acronym IREFS). It reflects only the authors' views.

References

1. Alonso, J.M., Magdalena, L., Gonzalez-Rodriguez, G.: Looking for a good fuzzy system interpretability index: An experimental approach. *International Journal of Approximate Reasoning* 51, 115–134 (2009)
2. Alonso, J.M., Magdalena, L., Guillaume, S.: HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism. *International Journal of Intelligent Systems* 23(7), 761–794 (2008)
3. Alonso, J.M., Magdalena, L.: Are fuzzy systems as interpretable (readable and understandable) as the fuzzy community usually claims? In: Third Workshop on Fuzzy Logic and Soft Computing (LFSC 2010) to be held as part of the Spanish Congress on Computer Science (CEDI 2010), Valencia (2010) (in press)
4. Angelov, P.P., Lughofer, E., Zhou, X.: Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems* 159(23), 3160–3182 (2008)
5. Carreira-Perpinan, M.A.: A review of dimension reduction techniques. Technical report, CS-96-09, Dept. of Computer Science, University of Sheffield, U.K (1997)
6. Casillas, J., Cordon, O., Herrera, F., Magdalena, L.: Interpretability Issues in Fuzzy Modeling. Springer, Heidelberg (2003)
7. Castro, J.L., Delgado, M.: Fuzzy systems with defuzzification are universal approximators. *IEEE Transactions on Systems, Man and Cybernetics* 26, 149–152 (1996)
8. Corchado, J.M., Bajo, J., De Paz, J.F., Rodriguez, S.: An execution time neural-CBR guidance assistant. *Neurocomputing* 72, 2743–2753 (2009)
9. Gonzalez, M.A.S., Formoso, C.T.: Mass appraisal with genetic fuzzy rule-based systems. *Property Management* 24(1), 20–30 (2006)
10. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72, 2729–2730 (2009)
11. Duro, R.J., Graña, M., de Lope, J.: On the potential contributions of hybrid intelligent approaches to Multicomponent Robotic System development. *Information Sciences* 180, 2635–2648 (2010)
12. Herrero, A., Corchado, E., Pellicer, M.A., Abraham, A.: MOVIH-IDS: A mobile-visualization hybrid intrusion detection system. *Neurocomputing* 72, 2775–2784 (2009)
13. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. New York London. Kluwer Academic Publishers, Dordrecht (2000)
14. Kosko, B.: Fuzzy systems as universal approximators. *IEEE Transactions on Computers* 43(11), 1329–1333 (1994)
15. Lughofer, E.: Evolving Fuzzy Systems—Methodologies, Advanced Concepts and Applications. Springer, Heidelberg (2011)
16. Lughofer, E.: Extensions of vector quantization for incremental clustering. *Pattern Recognition* 41(3), 995–1011 (2008)
17. Lughofer, E.: FLEXFIS: A robust incremental learning approach for evolving TS fuzzy models. *IEEE Trans. on Fuzzy Systems* 16(6), 1393–1410 (2008)
18. Lughofer, E., Klement, E.P.: FLEXFIS: A variant for incremental learning of Takagi-Sugeno fuzzy systems. In: Proceedings of FUZZ-IEEE 2005, Reno, Nevada, U.S.A, pp. 915–920 (2005)

19. Nguyen, N., Cripps, A.: Predicting housing value: A comparison of multiple regression analysis and artificial neural networks. *Journal of Real Estate Research* 22(3), 313–336 (2001)
20. Qin, S.J., Li, W., Yue, H.H.: Recursive PCA for adaptive process monitoring. *Journal of Process Control* 10, 471–486 (2000)
21. Taffese, W.Z.: Case-based reasoning and neural networks for real state valuation. In: AIAP 2007: Proceedings of the 25th IASTED International Multi-Conference, pp. 84–89. ACTA Press (2007)
22. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics* 15(1), 116–132 (1985)
23. Wang, L.X., Mendel, J.M.: Fuzzy basis functions, universal approximation and orthogonal least-squares learning. *IEEE Transactions on Neural Networks* 3(5), 807–814 (1992)
24. Worzala, E., Lenk, M., Silva, A.: An exploration of neural networks and its application to real estate valuation. *Journal of Real Estate Research* 10, 185–202 (1995)
25. Wyatt, P.J.: The development of a GIS-based property information system for real estate valuation. *International Journal of Geographical Information Science* 11, 435–450 (1997)

Investigation of Genetic Algorithms with Self-adaptive Crossover, Mutation, and Selection

Magdalena Smętek and Bogdan Trawiński

Wrocław University of Technology, Institute of Informatics,

Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

{magdalena.smetek, bogdan.trawinski}@pwr.wroc.pl

Abstract. A method of self-adaptive mutation, crossover and selection was implemented and applied in four genetic algorithms. So developed self-adapting algorithms were then compared, with respect to convergence, with a traditional genetic one, which contained constant rates of mutation and crossover. The experiments were conducted on six benchmark functions including two unimodal functions, three multimodal with many local minima, and one multimodal with a few local minima. The analysis of the results obtained was supported by statistical nonparametric Wilcoxon signed-rank tests. The algorithm employing self-adaptive selection revealed the best performance.

Keywords: self-adaptive GA, self-adaptive mutation, self-adaptive crossover, self-adaptive selection, benchmark functions.

1 Introduction

The problem of adapting the values of various parameters to optimize processes in Evolutionary Computation (EC) has been extensively studied for last two decades. The issue of adjusting genetic algorithms (GA) or Evolutionary Algorithms (EA) to the problem while solving it still seems to be a promising area of research. The probability of mutation and crossover, the size of selection tournament, or the population size belong to the most commonly set parameters of GA/EA. A few taxonomies of parameter setting forms in EC have been proposed [1], [7], [13]. Angeline [1] determines three different adaptation levels of GA/EA parameters: population-level where parameters that are global to the population are adjusted, individual-level where changes affect each member of the population separately, and component-level where each component of each member may be modified individually. The classification worked out by Smith and Fogarty [13] is based on three division criteria: what is being adapted, the scope of the adaptation, and the basis for change. The latter is further split into two categories: evidence upon which the change is carried out and the rule or algorithm that executes the change.

Eiben, Hinterding, and Michalewicz [7] devised a general taxonomy distinguishing two major forms of parameter value setting, i.e. parameter tuning and parameter control. The first consists in determining good values for the parameters before running GA/EA, and then tuning the algorithms without changing these values during the run (contradiction to the dynamic nature of GA/EA). The second form lies in dynamic

adjusting the parameter values during the execution. The latter can be categorized into three classes: deterministic (parameters are modified according to some deterministic rules without using any feedback from the optimization process), adapting (some feedback is also used to modify parameters) and self-adapting parameter control (parameters are encoded into the chromosomes and undergo mutation and recombination).

A series of parameter control methods have been proposed in the literature [3], [9], [12]. Several mechanisms of mutation and crossover adaptation and self-adaptation have been developed and experimentally tested [2], [4], [5]. Very often benchmark functions are employed to carry out the experiments to validate effectiveness and convergence of novel techniques and to compare them with other methods [6], [14]. Yao [15] categorized them into three groups: unimodal functions, multimodal with many local minima, and multimodal with a few local minima.

Our former investigations on the use of evolutionary algorithms to learn the rule base and learn membership functions of fuzzy systems devoted to aid in real estate appraisal showed it is a laborious and time consuming process [10]. Therefore, we intend to examine the usefulness of incorporating self-adapting techniques into our genetic fuzzy systems aimed to generate models for property valuation [8].

The goal of the present paper was to implement a self-adapting method of mutation, crossover, and selection setting in GAs based on an idea developed by Maruo et al. [11] and to test it using six benchmark function belonging to three classes, i.e. unimodal (U), multimodal with many local minima (MM), and multimodal with a few local minima (MF).

2 Self-adapting Method of Mutation, Crossover, Selection Setting

A self-adaptive method was implemented employing a binary encoding of a chromosome which, besides the solution, i.e. an argument or arguments of a given benchmark function, comprises mutation and crossover rates, thereby making them subject to evolution. The solution is represented with the accuracy of six decimal places, whereas the rates of both mutation and crossover of two decimal places. The mutation and crossover rates can take real values from the ranges of [0.0,0.3] and [0.0,1.0] respectively. To encode a real value X in a binary chromosome Y formula (1) was used:

$$Y = [X \cdot 10^d]_2 \quad (1)$$

where d denotes an accuracy and $[Q]_2$ means the conversion of Q to the binary system. In turn, to obtain a real value from the chromosome formula (2) was applied:

$$X = [Y]_{10}/10^d \quad (2)$$

where $[Q]_{10}$ denotes the conversion of Q to the decimal system. According to the above rules 5 genes are required for the mutation rate and 7 gens for the crossover rate. The chromosome with two self-adaptive parameters is shown in Fig. 1.

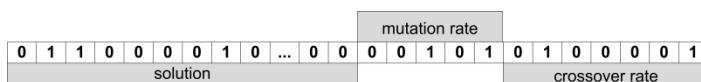


Fig. 1. Chromosome with self-adaptive parameters

Mutation. The self-adaptive mutation is illustrated in Fig. 2. It differs from a standard GA mutation which remains constant during the run. Each chromosome from the population can be subject to the mutation. A $N \times M$ matrix with real, randomly selected values from the range of [0.0,0.3] is created, where N is the number of chromosomes in the population, and M stands for the number of genes in the chromosome. Each gene in each chromosome is connected with one real value in the matrix. The self-adaptation of the mutation proceeds as follows. For each chromosome from the population:

- extract the genes representing the mutation rate from the chromosome,
- calculate the value of mutation rate extracted from chromosome,
- if the value from the matrix is lower than the value of the mutation rate taken from the chromosome, then the chromosome mutates in a traditional way,
- the matrix remains unchanged during the run.

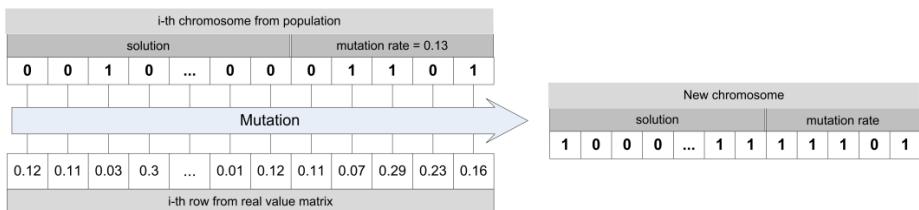


Fig. 2. Self-adaptive mutation

Crossover. The self-adaptive crossover depicted in Fig. 3 is also different from a traditional GA crossover. A $N \times 1$ table with real, randomly selected values from the range of [0.5,1.0] is created, where N is the number of chromosomes in the population. Each chromosome is connected with one real value in the table. The self-adaptation of the crossover proceeds in the following way. For each chromosome from population:

- extract the genes representing the crossover rate from the chromosome,
- calculate the value of crossover rate extracted from the chromosome,
- if the value from the table is lower than the value of crossover rate from the chromosome, then the chromosome is selected to a classic crossover process,
- the table remains unchanged during the run.

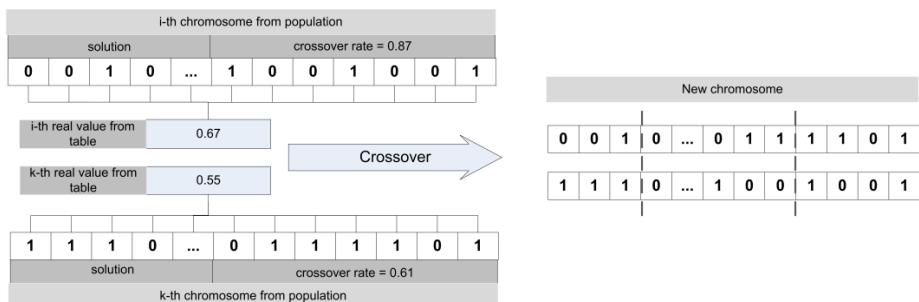


Fig. 3. Self-adaptive crossover

Selection. Self-adapting selection depends on the control of a population size. Each chromosome is connected with one integer value, which expresses the aging level of the chromosome. At the beginning this value is set 3 for each chromosome. The self-adaptation of the selection proceeds as follows. For each chromosome from the population:

- subtract 1 from the aging level,
- add 1 to the aging level if the value of the chromosome fitness function is lower than median of the values of all chromosomes or subtract 1 from the aging level in the opposite case,
- subtract 2 from the aging level if the population size has been increased 10 times and the fitness function of the chromosome is not in top 1000 values of the fitness function in the whole population,
- remove from the population all chromosomes with the aging level lower or equal to zero.

3 Plan of Experiments

The main goal of our experiment was to compare, in respect of a convergence, a classic GA with four self-adapting GAs in which rates of mutation, crossover, or population size were incorporated. Following denotation was used in the remaining part of the paper: GA - a classic genetic algorithm, SAM – a self-adaptive genetic algorithm with adapting mutation rate, SAC – a self-adaptive genetic algorithm with adapting crossover rate, SAMC – a self-adaptive genetic algorithm with adapting both mutation and crossover rates, SAS – a self-adaptive genetic algorithm with adapting selection (i.e. population size).

Table 2. Benchmark functions used in experiments

Type	Function	n	Domain	f_{min}
U	$f_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
	$f_2(x_1, x_2) = -\cos(x_1) \cos(x_2) e^{-(x_1-\pi)^2 + (x_2-\pi)^2}$	2	[-100,100]	-1
MM	$f_3(x) = -20e^{-0.2\sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}} - e^{\sum_{i=1}^n \cos(2\pi x_i)} + 20 + e$	30	[-1,-1]	0
	$f_4(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	30	[-5.12,5.12]	0
	$f_5(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1$	30	[-600,600]	0
	$f_6(x_1, x_2) = (1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \cdot (30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$	2	[-2,2]	3

The algorithms were used to find minimal values (f_{min}) of six benchmark functions: f_1 – De Jong's function, f_2 – Easom function, f_3 – Ackley's Path function, f_4 – Rastrigin's function, f_5 – Griewangk's function, f_6 – Goldstein-Price function. They were arranged in 3 groups: unimodal (U), multimodal with many local minima (MM), and multimodal with few local minima (MF). The functions we employed are listed in Table 2.

The parameters of a classic GA were as follows: mutation rate was equal to 0.15, crossover rate to 0.80, and population size was set to 100. A seven-chromosome tournament method is applied to the selection process. The chromosome length depends on the accuracy of the solution. One third of genes in the chromosome undergo a mutation. A two-point crossover with randomly selected positions of breaking was applied. The initial aging value was 3.

One fitness function was used and based on commonly known mean absolute error measure (MAE) expressed in the form of formula (3) where y_i denotes actual value and \hat{y}_i – predicted value by a model of i -th case. The fitness function, denoted by MAEy, was calculated for the output value of a given benchmark function. It determined how near the optimum was the output value of the function. In this case N was always equal 1.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (3)$$

All five algorithms, i.e. GA, SAM, SAC, SAMC, SAS, were executed independently 50 times and the final values of MAEy were calculated as an average over 50 runs for best individuals found by respective algorithms. 50 initial populations composed of 100 chromosomes were randomly created and they were the same for all algorithms in each run. In order to investigate the convergence of individual algorithms, 100 generations were carried out in each run and the values of MAEy were calculated after each five cycles.

Moreover, nonparametric Wilcoxon signed-rank tests were carried out for MAEy provided by the algorithms by the 100-th generation over 50 independent runs for individual benchmark functions.

4 Results of Experiments

The performance of GA, SAM, SAC, SAMC, SAS algorithms on respective benchmark functions in respect of MAEy measures was shown in Fig. 4-9. In each case SAS, SAC, and SAMC, revealed better convergence than GA. SAM achieved similar results as GA. Moreover, SAS, SAC and SAMC produced lower values of MAEy than SAM. SAS achieved the best results for all functions and in each case. The advantage of SAM and SAMC algorithms over GA is apparent particularly on De Jong's, Ackley's Path, Goldstein-Price function.

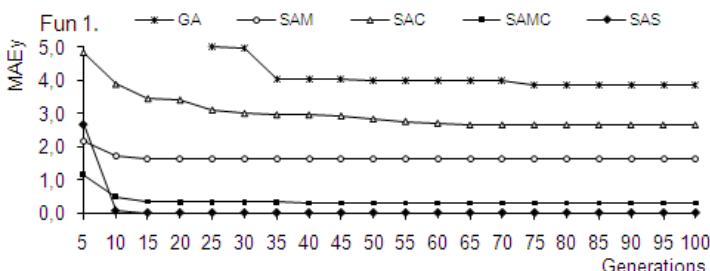
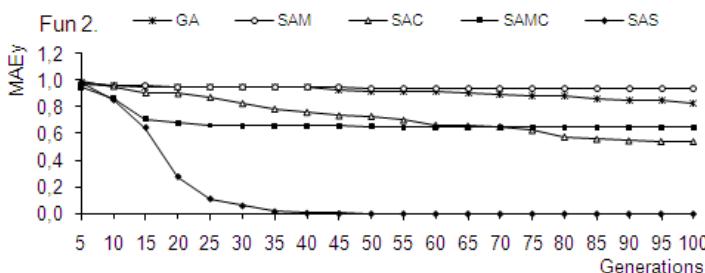
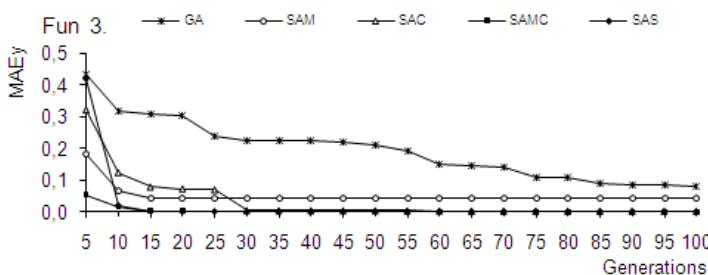
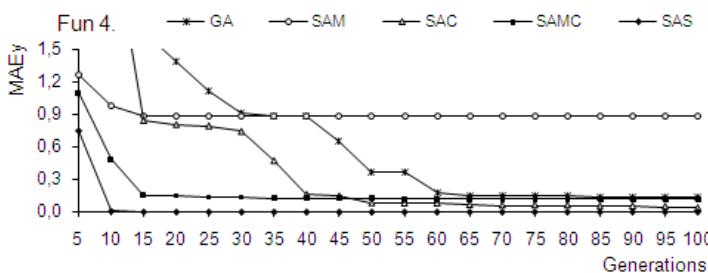
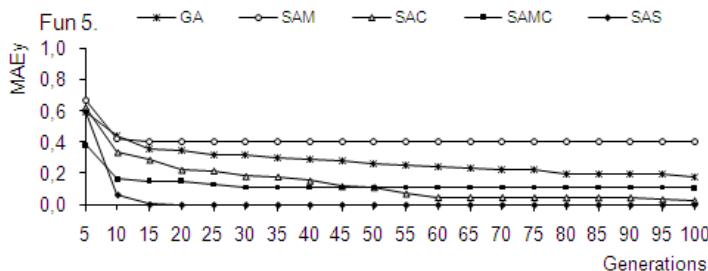
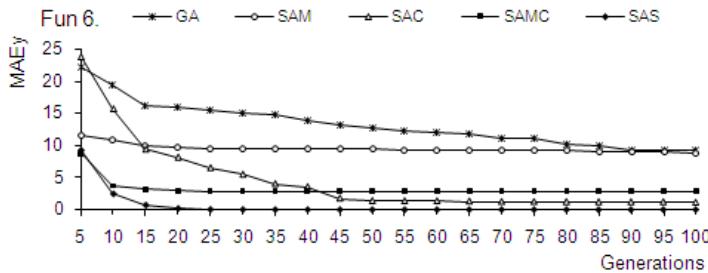


Fig. 4. Performance of algorithms on De Jong's function in terms of MAEy

**Fig. 5.** Performance of algorithms on Easom function in terms of MAEy**Fig. 6.** Performance of algorithms on Ackley's Path function in terms of MAEy**Fig. 7.** Performance of algorithms on Rastrigin's function in terms of MAEy**Fig. 8.** Performance of algorithms on Griewangk's function in terms of MAEy

**Fig. 9.** Performance of algorithms on Goldstein-Price function in terms of MAEY**Table 3.** Results of Wilcoxon tests for GA, SAM, SAC, SAMC, SAS algorithms

Alg vs Alg	f_1	f_2	f_3	f_4	f_5	f_6
GA vs SAM	\approx	\approx	+	\approx	+	\approx
GA vs SAC	-	-	-	-	-	-
GA vs SAMC	-	-	-	-	\approx	-
GA vs SAS	-	-	-	-	-	-
SAM vs SAC	\approx	-	-	-	-	-
SAM vs SAMC	-	-	-	-	-	-
SAM vs SAS	-	-	-	-	-	-
SAC vs SAMC	-	\approx	-	\approx	\approx	\approx
SAC vs SAS	-	-	-	-	-	-
SAMC vs SAS	-	-	-	-	-	-

The results of the Wilcoxon test are given in Table 3, where +, -, and \approx denote that the first algorithm in a pair performed significantly better than, significantly worse than, or statistically equivalent to the second algorithm, respectively. Main outcome is as follows: GA was significantly worse than SAS, SAC, SAMC for each benchmark function, besides one case. Differences between SAC, SAMC and between GA, SAM are not so clear. Only SAS performed significantly better than all other algorithms for all of functions.

5 Conclusions and Future Work

The experiments aimed to compare the convergence of a classic genetic algorithm (GA) with four self-adapting genetic algorithms (SAM, SAC, SAMC, SAS) in which the rates of mutation and crossover or population size were dynamically evolved. Six benchmark functions were employed including two unimodal, three multimodal with many local minima, and one multimodal function with a few local minima.

The results showed that almost all self-adaptive algorithms revealed better convergence than traditional genetic one. Moreover, SAS, SAC and SAMC produced lower values of fitness function than SAM. SAS was the best algorithm for all functions and in each case. The advantage of SAM and SAMC algorithms over GA became particularly apparent on De Jong's, Ackley's Path, Goldstein-Price function.

Statistical nonparametric Wilcoxon signed-rank tests allowed for the analysis of behaviour of the algorithms on individual benchmark functions.

Further research is planned to extend the self-adaptive parameters to include the tournament size of selection, and number of cross-points. More criteria of the algorithm assessment will be taken into account. The possible application of the self-adaptive techniques to create genetic fuzzy models assisting with real estate appraisal will also be considered.

Acknowledgments. This paper was partially supported by the Polish National Science Centre under grant no. N N516 483840.

References

1. Angeline, P.J.: Adaptive and self-adaptive evolutionary computations. In: Palaniswami, M., Attikiouzel, Y. (eds.) *Computational Intelligence: A Dynamic Systems Perspective*, pp. 152–163. IEEE Press, New York (1995)
2. Bäck, T.: Self-adaptation in genetic algorithms. In: Varela, F.J., Bourgine, P. (eds.) *Proc. First European Conference on Artificial Life, Toward a Practice of Autonomous Systems*, pp. 263–271. MIT Press, Cambridge (1992)
3. Bäck, T., Schwefel, H.-P.: An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation* 1(1), 1–23 (1993)
4. Cervantes, J., Stephens, C.S.: Limitations of Existing Mutation Rate Heuristics and How a Rank GA Overcomes Them. *IEEE Trans. Evolutionary Computation* 13(2) (2009)
5. Deb, K., Beyer, H.-G.: Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation* 9(2), 197–221 (2001)
6. Digalakis, J.G., Margaritis, K.G.: An Experimental Study of Benchmarking Functions for Genetic Algorithms. *Int. J. Computer Math.* 79(4), 403–416 (2002)
7. Eiben, E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 3(2), 124–141 (1999)
8. Herrera, F., Lozano, M.: Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. *Soft Computing* 7(8), 545–562 (2003)
9. Hinterding, R., Michalewicz, Z., Eiben, A.E.: Adaptation in Evolutionary Computation: A Survey. In: *Proceedings of the Fourth International Conference on Evolutionary Computation (ICEC 1997)*, pp. 65–69. IEEE Press, New York (1997)
10. Król, D., Lasota, T., Trawiński, B., Trawiński, K.: Investigation of evolutionary optimization methods of TSK fuzzy model for real estate appraisal. *International Journal of Hybrid Intelligent Systems* 5(3), 111–128 (2008)
11. Maruo, M.H., Lopes, H.S., Delgado, M.R.: Self-Adapting Evolutionary Parameters: Encoding Aspects for Combinatorial Optimization Problems. In: Raidl, G.R., Gottlieb, J. (eds.) *EvoCOP 2005. LNCS*, vol. 3448, pp. 154–165. Springer, Heidelberg (2005)
12. Meyer-Nieberg, S., Beyer, H.-G.: Self-Adaptation in Evolutionary Algorithms. In: Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.) *SCI*, vol. 54, pp. 47–75. Springer, Heidelberg (2007)
13. Smith, J.E., Fogarty, T.C.: Operator and parameter adaptation in genetic algorithms. *Soft Computing* 1(2), 81–87 (1997)
14. Tang, K., Li, X., Suganthan, P.N., Yang, Z., Weise, T.: Benchmark Functions for the CEC 2010 Special Session and Competition on Large Scale Global Optimization, Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China (2009), <http://nical.ustc.edu.cn/cec10ss.php>
15. Yao, X., Liu, Y.: Fast evolution strategies. *Contr. Cybern* 26(3), 467–496 (1997)

Hybrid Multi-agent System for Knowledge Management in Distributed Control System

Dariusz Choinski, Mieczyslaw Metzger, and Witold Nocon

Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland
{dariusz.choinski,mieczyslaw.metzger,witold.nocon}@polsl.pl

Abstract. DCS (Distributed Control System) software systems are one of the typical applications in industry used for monitoring, measurement and control. The paper deals with knowledge management based on Hybrid Multi-Agent System environment and the presented solutions are intended to enable dual abstraction of DCS both for ontology and control system design in a platform independent way. Hybrid Multi-Agent based software systems successfully join both reliability and efficiency in knowledge management processes which is a key aspect of robust automation software system. The proposed hybrid control system is designed for stimulation of wastewater treatment pilot process.

Keywords: Knowledge management, Ontology, Distributed Control System, Hybrid Multi-Agent System, OPC, Bioprocess stimulation.

1 Introduction and Problem Formulation

Modern industrial processes are characterised by combination of technological equipment with complexity of automation and information control systems. For such complex systems (e.g. Rockwell Automation software supports distributed operations with 70000 tags) standard separate design and operation of processes are not sufficient. The most import aim is not only Computer System integration but knowledge engineering and acquisition. The Knowledge Engineering in current real life automation software systems solutions is a major problem that determines its efficiency. This paradigm refers to all kinds of industries and it is not an easy task to cope with. In most situations operators need to have deep functional level of knowledge about automation system in order to act appropriately according to various types of situations which also means that only system specialists can cooperate with the system efficiently. It is required to have an extremely wide area of expertise in such situations. Control of different biotechnology related processes may serve as examples [1], [2]. Even the end user has to be familiar to some extent with the automation software system in order to perform quick and efficient analysis on the acquired information data. Deep functionality knowledge can not be acquired instantly and thus each company has to employ rather a large number of experienced specialists which can be problematic at some point.

The DCS (Distributed Control System) architecture and software is based on rules for structuring and formalism of the technological projects [3]. The structure defines

dependencies and relations between objects that are considered in the process of designing, construction, realising, operation, exploitation and disposal. The particular structures are organized hierarchically and should specify the information regarding the system, content of the formalized documents and composition of referential designations.

The presented topic regards problems connected to pattern recognition. For this type of problems, it is necessary to obtain classifiers relating to different domains of competence [4]. It also relates to evolutionary algorithms and coevolutionary algorithms for the reduction of data during selection of the searched for patterns [5]. Finding solutions to the above problems is facilitated by hybridization, which is one of the techniques enabling the realization of computational intelligence [6], and neurocomputing [7].

On the other hand, experience gained during synthesis of real multi-agent systems show, that it is extremely important to be able to extract and transmit knowledge and language describing this knowledge during software creation. The effect is ontology, which although is by itself static, enables agents to behave dynamically depending on changing environmental conditions. This feature of obtaining dynamical behaviour based on a static knowledge is probably the most important advantages and behaviours for designers of multi-agent systems from the point of view of distributed control systems applications.

Ontologies look simple for discrete systems because of being based on discrete logic. However, for continuous processes with infinite number of states when the problem is more difficult, creating ontology may be possible in some cases [8]. Using description with decomposition to subsystems and functions, it is possible to treat subsystems as hybrid automata [9] and to apply flexible control of hybrid systems [10], [11]. The major advantage of a hybrid system deals with its particular properties when event-driven part (hybrid automaton subsystem) and time-driven part (continuous state equations subsystem) can be corrected separately without disturbing the whole system. Hybrid control facilitates recognition of discrete states required by technology. If it is possible to define those states and technology rules, than binding of those states by ontology is also possible. Practice of processes automation shows that the automated plant may be represented as a hybrid state machine [10], [12]. A-prior knowledge about process is based on rules of the hybrid systems with explicit models. One of the important features describing such systems refers to the description of condition for transitions between states.

The proposed hybrid control system depicted in Fig. 1 compiles techniques discussed above and enables dual abstraction of Distributed Control System [13]. Considering those facts, the automation software system user faces challenging tasks involving fast knowledge management and data analysis processes. Those are major problems which will emerge sooner or later in modern systems. Hybrid Multi-Agent Systems (HMAS) relieves the user to some extent from both hard analysis and decision making processes making them less error prone. Such system notion is focused around ontology concept. In most cases ontologies are considered as system knowledge. This knowledge is modelled and implemented beforehand HMAS application and it is known prior to its runtime. Basically, ontology describes static properties of the system as well as runtime states in which the system can reside and conditions of transitions between those states. Based on the ontology, agents can share their common knowledge and work over common system problems because sharing knowledge is the natural way for single pieces of Multi-Agent software system to achieve their goals.

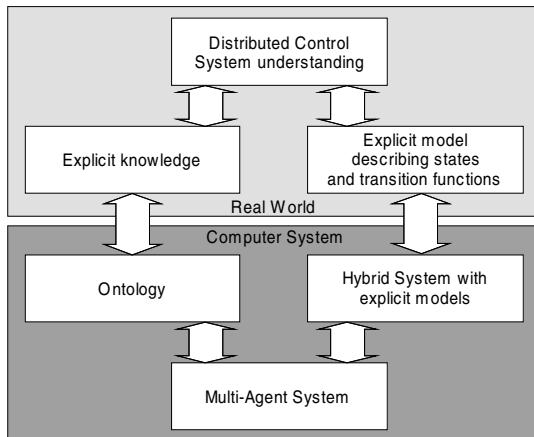


Fig. 1. Dual abstraction in Distributed Control System

The paper is organised as follows. The Hybrid Multi-Agent System is presented in section 2, while knowledge management in Hybrid System is presented in section 3. Finally, in section 4 the HMAS application for stimulation of wastewater treatment pilot plant is presented and concluding remarks are outlined in section 5.

2 Multi-agent System

Sharing knowledge is the natural way for single pieces of Multi-Agent software system to achieve their goals. By achieving single goals agents transform the system from one designed and implemented state to another one based on encoded actions. In order to achieve its goals agent may sometimes even retake decision making power from the user and perform a designed scenario of actions on its own thus making it faster, less error prone and more efficient. It is worth mentioning that even in such situations, the user holds superior role in the system so his interaction with the system is preserved.

There are many solutions of how a Multi-Agent System can be structured [14], [15]. Prior to Multi-Agent System implementation, automation system granulation should be established. Granulation of the system shows how agents should correspond to system resources. Such knowledge verifies weight of the source code that corresponds to each agent. This also informs about particular agent functionalities [16].

3 Knowledge Sharing Based on Hybrid System

As practice of processes automation shows, automated plants may be represented as hybrid state machines. The presented Hybrid System consists of two subsystems: a subsystem representing a continuous dynamic and a subsystem representing discrete events and mutual interactions with other systems. Those conditions are given either explicitly – are computed beforehand based on the knowledge about the plant and are

dependent on a measurable state, or implicitly - are given by differential and algebraic equations describing a divisible state space and transitions specifications [10]. Those equations must be solved on-line using appropriate methods [17], [18]. In both cases, knowledge about continuous controlled plant (as a whole) may be approximated in order to use discrete control on the higher level of the hierarchy. Such hybrid system is described with two sets:

- Ω – set of continuous state variables. Range of those variables depends on the features of the considered system i.e. system constraints, measurements, activator's capabilities, etc.;
- Φ – set of events conditions enabling transitions between states. The Φ set can be divided into two subsets:
 - Φ_u – subset of uncontrollable events – transitions caused by internal or external process behaviour but the control system can not prevent them;
 - Φ_c – subset of controllable events – transitions which can be initiated and/or prevented by the control system, so those transitions can be used to obtain desired process state.

The state can change in two ways: instantaneously by discrete transition described by the sequences S of actions from source state to target state or in a time pass according to a trajectory of the state variables change as a function f of input and output variables.

Agents are responsible for transitions, therefore Φ describes the agent sensors and Ω describes common agent effectors. Apart from effectors used commonly by all agents, each agent possesses other effectors that are characteristic and dependant on the knowledge source. The type of agent depends on the way the state changes and on the agent's knowledge. Agents are specified on a design level in an iterative process in addition to knowledge needed. Changes in system can be performed as discrete events when certain conditions defined in Φ are fulfilled. Control of a hybrid automaton is a complex task, which can be achieved with Multi-Agent Systems (MAS). The proposed structure of control agents is hierarchical and consists of three layers: control, supervisory and expert. Each layer performs tasks at a designated level of abstraction, and actively cooperates to achieve control goals in a most efficient manner.

The lowest layer consists of Control Agents. A Control Agent is bound directly to control instrumentation of the controlled plant. It implements all the required control algorithms for direct influence on process. Agents in control layer work in time-driven mode to be able to maintain time determined control algorithms. Uncontrollable events are treated as disturbances for Control Agents. The layer in the middle of the control system is formed by Supervisory Agent. Each agent may supervise a set of Control Agents and monitor system performance quality. Supervisory Agent, partially utilising expert knowledge, is capable of more advanced state recognition and trajectory planning, and may decide when to switch the system into some Ω_i state (or through some $\Omega_1, \dots, \Omega_n$ states) to fulfil a given goal. The plan is then carried out by the specific Control Agent as a proxy. The top layer contains Expert Agents. Expert Agent is a system interface to external sources of knowledge such as human experts. Expert Agent's task is to supply Supervisory Agents with additional knowledge about the system. It is expected for them to work on abstract knowledge, not the specific realisation. It can be treated as a universal expert for a specific process class. Experts are designed to serve their general knowledge. Both supervisory layer and expert

layer work in an event-driven mode, which in consequence means, that the multi-agent layered structure decomposes the control system into two parts: time-driven (Control Agents) and event-driven (Supervisory and Expert Agents). Such decomposition allows for separate analysis and decrease complexity level of the control system. The cooperation may proceed on the three levels of interaction strategies. The first is the level of the Competitive Agents. Such Agents may try to change the decision based on the subset of controllable events. The second is level of Collaborative Agents which share their knowledge maximizing benefits of communication in critical system controls and cooperating based on uncontrollable events. The last level is the level of Hostile Agents, which should interrupt others agents with improper information or connected to out of order actuator. The interaction is controlled by JADE-based [19],[20] MAS system.

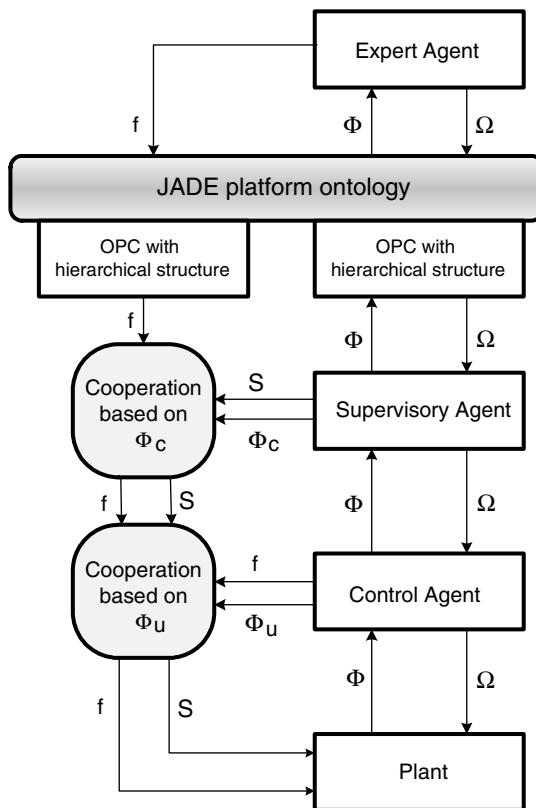


Fig. 2. Knowledge sharing

4 HMAS Application for Stimulation of Biological Pilot Plant

A two-tank wastewater treatment pilot plant designed and operated at the Institute of Automatic Control serves as an example of different modes of control - both standard

control and advanced intelligent control [10],[11]. The plant can work in the classic architecture (bioreactor plus settler) as well as sequencing batch reactor (SBR) with one or two tanks. More detailed plant description with intelligent HMI for operating monitoring and visualization can be found in [11]. Actually, one of two DCS systems (FieldPoint from National Instruments and Logix from Rockwell) physically connected to the pilot plant can realize information processing and control.

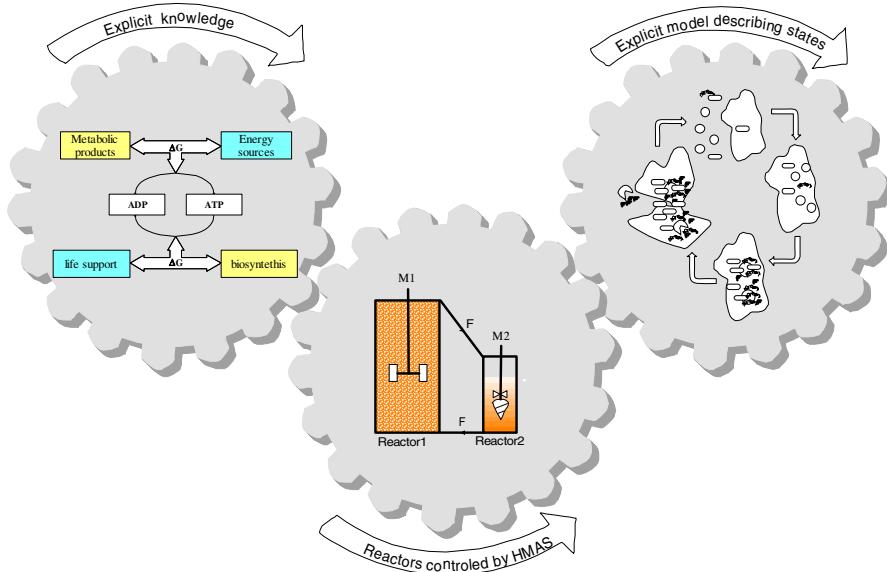


Fig. 3. Principle of stimulation and modulation control of physical properties for synchronization of metabolic cycles.

The presented idea of dual DCS abstraction with knowledge management based on HMAS can be utilized for a new idea of directed stimulation and modulation of biological process (Fig. 3) actually implemented and tested on the pilot plant.

For easier presentation, the proposed control idea is symbolically represented by three mutually related gearwheels (Fig. 3). The central drive is a two tank control system based on a hybrid MAS. One of those tanks is a biological reactor. The other tank realizes a controlled retention of activated sludge by using an original mixing system. The manipulated variables are: mixing speeds in both tanks (M1 and M2) and recycle flow F (different flows to and from the main reactor may be used if needed). Additionally, standard oxygen, redox and pH control loops are realized in the reactor.

The principle of control is based on the knowledge about metabolic cycles that is used for synchronization of metabolic paths. The left gear represents the well-known scheme of biochemical conversions taking place in living cells. This scheme presents a simplified knowledge that the control systems uses while controlling the bioprocess stimulation. The metabolic paths take into consideration the catabolism (substrate → product + energy) as well as the anabolism (substrate + energy → product).

One of the most promising physical mechanisms for stimulation of bioprocesses is control of biomass states in the two-tank system (right gear). Experimentally selected values of mixing force M1 and M2 and flow F enable to control the ratio of activated sludge flocs to free particles. This is forced by controlling the concentration of activated sludge in the second tank. Changes in flocs to free particles ratio stimulates the desired process states.

5 Concluding Remarks

Using agents and multi agent systems (AMAS) is already well founded in the manufacturing industry. A good application area for AMAS into process industry are biotechnological processes because of their level of complexity and unpredictability. The solution proposed in this paper, namely a hybrid multi agent system is dedicated to automation of biological process by stimulation of metabolic cycles that are forced by changing the ratio between flocs and free particles, thus allowing satisfactory process parameters to be obtained. Future work will be directed towards finding other means of stimulation of metabolic processes. One such possibility may be the utilization of dual substrate feeding of biomass.

Acknowledgments. This work was supported by the Polish Ministry of Science and Higher Education, grant no. N N514 471539.

References

1. Choiński, D., Metzger, M., Nocoń, W.: Multiscale three-phase flow simulation dedicated to model based control. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part II. LNCS, vol. 5102, pp. 261–270. Springer, Heidelberg (2008)
2. Nocoń, W., Metzger, M.: Predictive Control of Decantation in Batch Sedimentation Process. *AICHE Journal* 56, 3279–3283 (2010)
3. Wainwrighta, D., Waring, T.: Three domains for implementing integrated information systems: redressing the balance between technology, strategic and organisational analysis. *International Journal of Information Management* 24, 329–334 (2004)
4. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants - Evolutionary and Neural Methods of Training. *HAIS* (1), 590–597 (2010)
5. Derrac, J., García, S., Herrera, F.: A First Study on the Use of Coevolutionary Algorithms for Instance and Feature Selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) *HAIS* 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
6. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)
7. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72(13-15), 2729–2730 (2009)
8. Leduc, R.J., Lawford, M., Dai, P.: Hierarchical Interface-Based Supervisory Control of a Flexible Manufacturing System. *IEEE Transactions on Control Systems Technology* 14, 654–668 (2006)

9. van der Schaft, A.J., Schumacher, J.M.: Compositionality issues in discrete, continuous, and hybrid systems. *International Journal of Robust and Nonlinear Control* 11(5), 399–539 (2001)
10. Choinski, D., Metzger, M., Nocon, W.: Decomposition of Information for Plant Collaborative Design and Set-Up. In: Luo, Y. (ed.) CDVE 2008. LNCS, vol. 5220, pp. 125–132. Springer, Heidelberg (2008)
11. Choinski, D., Metzger, M., Nocon, W.: Cooperative operating control based on virtual resources and user-suited HCI. In: Luo, Y. (ed.) CDVE 2009. LNCS, vol. 5738, pp. 216–223. Springer, Heidelberg (2009)
12. Sprinkle, J.: Generative Components for Hybrid Systems Tools. *Journal of Object Technology* 4(3), 35–40 (2005)
13. Gruber, T.: A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2), 2105–2127 (1993)
14. Ruey-Shun, C., Duen-Kai, C.: Apply ontology and agent technology to construct virtual observatory. *Expert Systems with Applications* 34, 2019–2028 (2008)
15. Malucelli, A., Palzer, D., Oliveira, E.: Ontology-based Services to help solving the heterogeneity problem in e-commerce negotiations. *Electronic Commerce Research and Applications* 5, 29–43 (2006)
16. Kularbhettong, K., Clayton, G., Meesad, P.: A Hybrid System based on Multi-Agent System in the Data Preprocessing Stage. *International Journal of Computer Science and Information Security* 7, 199–202 (2010)
17. Metzger, M.: Comparison of the RK4M4 RK4LIN and RK4M1 methods for systems with time-delays. *Simulation* 52(5), 189–193 (1989)
18. Czeczot, J., Laszczyk, P., Metzger, M.: Local balance-based adaptive control in the heat distribution system - Practical validation. *Applied Thermal Engineering* 30(8-9), 879–891 (2010)
19. Bellifemine, F., Caire, G., Greenwood, D.: Developing multi-agent systems with JADE. John Wiley & Sons, Chichester (2007)
20. Choinskis, D., Senik, M.: Collaborative Control of Hierarchical System Based on JADE. In: Luo, Y. (ed.) CDVE 2010. LNCS, vol. 6240, pp. 262–269. Springer, Heidelberg (2010)

SVM with Bounds of Confidence and PLS for Quantifying the Effects of Acupuncture on Migraine Patients

M. López¹, J.M. Górriz¹, J. Ramírez¹, D. Salas-Gonzalez¹,
R. Chaves¹, and M. Gómez-Río²

¹ Dept. of Signal Theory, Networking and Communications
University of Granada, Spain

² Department of Nuclear Medicine
Hospital Universitario Virgen de las Nieves, Granada, Spain

Abstract. In this work, SPECT brain images are analyzed automatically in order to determine whether acupuncture, applied under real conditions of clinical practice, is effective for fighting migraine. To this purpose two different groups of patients are randomly collected and received *verum* and *sham* acupuncture, respectively. Acupuncture effects on brain perfusion patterns can be measured quantitatively by dealing with the images in a classification context. Partial Least Squares are used as feature extraction technique, and Support Vector Machines with bounds of confidence are used to quantify the acupuncture effects on the brain activation pattern. Conclusions of this work prove that acupuncture produces new brain activation patterns when applied to migraine patients.

Keywords: SVM, PLS, bounds of confidence, SPECT, acupuncture, migraine.

1 Introduction

Acupuncture is widely used to treat headache, and can be applied as the sole therapy or as part of a more complex treatment program. Despite its popularity, there persists some controversy as the differentiation between the specific and the nonspecific effects of acupuncture. Reports in both animals and humans have suggested that acupuncture may result from the activation and deactivation of a number of brain structures such as the lateral and posterior hypothalamus, the lateral septum, the dorsal hippocampus or medial centro-median nucleus of the thalamus, the arcuate nucleus, the ventral median nucleus of the hypothalamus [7].

Single Photon Emission Computed Tomography (SPECT) is a widely used technique to study the functional properties of the brain. In general, tomographic radiopharmaceutical imaging provides *in vivo* three-dimensional maps of a pharmaceutical labeled with a gamma ray emitting radionuclide. The distribution of radionuclide concentrations are estimated from a set of projectional

images acquired at many different angles around the patient. SPECT imaging techniques employ radioisotopes which decay emitting predominantly a single gamma photon.

On the other hand, migraine is a chronic neurologic disease that can severely affect the patient's quality of life. Several authors have studied and analyzed the cerebral blood flow of migraine patients in SPECT imaging [1] and, in recent years, many randomised studies have been carried out to investigate the effectiveness of acupuncture as a treatment for migraine although it remains a controversial issue [11].

Our aim in this paper is to quantitatively determine whether acupuncture, applied under real conditions of clinical practice in the area of primary healthcare, is effective for fighting migraine. This quantitative approach used for SPECT image assessment is based on an analysis of the images in a classification framework [13]. First, Partial Least Squares (PLS) is used as feature extraction technique for gathering useful information contained in the images at the same time a reduction of the feature space dimensionality is performed [4]. Secondly, Support Vector Machine (SVM)-based classifiers are trained and tested on the PLS features so that quantitative assessment can be made from the evaluation of the classifier in terms of accuracy rate and bounds of confidence. This measures are used to quantify the effects of acupuncture on migraine patients.

2 Image Acquisition and Preprocessing

SPECT images used in this work were taken with a PRISM 3000 machine and a SIEMENS ECAT 47 respectively. 3D brain perfusion volumes are reconstructed from projection data using the filtered backprojection (FBP) in combination with a Butterworth noise filter. The SPM software [5] is used for the spatial normalization of the images, in order to ensure that the voxels in different images refer to the same anatomical positions in the brain [9], giving rise to volumes of size $68 \times 95 \times 79$. Finally, intensity level of the SPECT images is normalized for each sample to its maximum intensity, which is computed for each volume individually by averaging over the 3% of the highest voxel intensities following a procedure similar to [10].

3 Feature Extraction and Classification

3.1 Partial Least Squares

PLS [12] is a method for modeling relations between sets of observed variables by means of latent variables. The basic idea of PLS is to construct new predictor variables, latent variables, as linear combinations of the original variables summarized in a matrix \mathbf{V} of descriptor variables (features) and a vector \mathbf{y} of response variables (class labels).

Let $\mathbf{V} \in \mathbb{R}^n$ denote an n -dimensional space of image vectors and similarly let $y \in \mathbb{R}$ be a 1-dimensional space representing the class labels. Let the number of

samples be N . PLS decomposes the zero-mean $N \times n$ matrix \mathbf{V} and the $N \times 1$ zero-mean vector \mathbf{y} into

$$\begin{aligned}\mathbf{V} &= \mathbf{T}\mathbf{P}^T + \mathbf{E} , \\ \mathbf{y} &= \mathbf{U}\mathbf{q}^T + \mathbf{f} ,\end{aligned}\quad (1)$$

where \mathbf{T} and \mathbf{U} are $N \times p$ matrices containing p extracted latent vectors, the $n \times p$ matrix \mathbf{P} and the $1 \times p$ vector \mathbf{q} represent the loadings and the $N \times n$ matrix \mathbf{E} and the $N \times 1$ vector \mathbf{f} are the residuals. The PLS method, using the nonlinear iterative partial least squares (NIPALS) algorithm [12], constructs a set of weight vectors (or projection vectors) $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_p\}$ such that

$$[cov(\mathbf{t}_i, \mathbf{u}_i)]^2 = \max_{|\mathbf{w}_i|=1} [cov(\mathbf{V}\mathbf{w}_i, \mathbf{y})]^2 , \quad (2)$$

where \mathbf{t}_i is the i -th column of matrix \mathbf{T} , \mathbf{u}_i the i -th column of matrix \mathbf{U} and $cov(\mathbf{t}_i, \mathbf{u}_i)$ is the sample covariance between latent vectors \mathbf{t}_i and \mathbf{u}_i . After the extraction of the latent vectors \mathbf{t}_i and \mathbf{u}_i , the matrix \mathbf{V} and vector \mathbf{y} are deflated by subtracting their rank-one approximations based on \mathbf{t}_i and \mathbf{u}_i . This process is repeated until the desired number of latent vectors is extracted.

The dimensionality reduction is performed by projecting the feature vectors \mathbf{x}_i , $i = 1, \dots, N$ onto the weight vectors $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_p\}$, obtaining the PLS coefficients \mathbf{x}_i ($1 \times p$) as a result. These vectors of coefficients are used in classification.

3.2 Support Vector Machines

SVM [2] are widely used for pattern recognition in a number of applications by its ability to learn from experimental data [83]. SVM separate binary labeled training data by the hyperplane

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 , \quad (3)$$

where \mathbf{w} is known as the weight vector and w_0 as the threshold. This hyperplane is maximally distant from the two classes (known as the maximal margin hyperplane). The objective is to build a function $f : \mathbb{R}^p \rightarrow \{\pm 1\}$ using training data that is, p -dimensional patterns \mathbf{x}_i obtained in the feature extraction step, and class labels y_i :

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N) \in \mathbb{R}^p \times \{\pm 1\} , \quad (4)$$

so that f will correctly classify new examples (\mathbf{x}, y) . In this work, only the linear case is treated.

In a binary classification task, the classifier may make either of the two mistakes: considering like positive a test sample that was initially labeled as negative (i.e., the associated real label is $y = -1$) which is called False Positive (FP), or the contrary, which is called False Negative (FN). When positive and negative samples are correctly classified they are marked as True Positive (TP) and True Negative (TN), respectively. The performance of the classifier can be assessed by computing accuracy, sensibility (ability of the classifier to detect true positive

samples) and specificity (ability of the classifier to detect true negative samples) rates, which are defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} , \quad (5)$$

$$\text{Sensibility} = \frac{TP}{TP + FN} , \quad \text{Specificity} = \frac{TN}{TN + FP} , \quad (6)$$

3.3 Bounds of Confidence

In case of dealing with non linearly separable data, the maximal margin hyperplane defined by SVM does not guarantee a correct classification of each single sample in its corresponding subspace. Bounds of confidence were introduced in [6] for defining a “security zone” as a region bordering on the hyperplane where samples should not be considered for classification to avoid risky decisions. Given the training set T containing samples labeled as positive ($y = 1$) or negative ($y = -1$) corresponding to class ω_1 and ω_2 respectively, we first define two metrics to approximate error rates E^- and E^+ as

$$E^-(f_-) = \frac{\text{card}(\mathbf{x}|f(\mathbf{x}) \leq f_-, \text{cls}(\mathbf{x}) = \omega_1, \mathbf{x} \in T)}{\text{card}(\mathbf{x}|f(\mathbf{x}) \leq f_-, \mathbf{x} \in T)} ,$$

$$E^+(f_+) = \frac{\text{card}(\mathbf{x}|f(\mathbf{x}) \leq f_+, \text{cls}(\mathbf{x}) = \omega_2, \mathbf{x} \in T)}{\text{card}(\mathbf{x}|f(\mathbf{x}) \leq f_+, \mathbf{x} \in T)} ,$$

where card is the cardinality of a set and cls provides the class memberships of samples. A properly trained SVM should rank samples from the most negative samples to the most positive samples. After the training step, the outcomes of the SVM allow us to find two thresholds, the negative bound (NB) and the positive bound (PB). Basically, the NB is the highest score whose E^- is less than the given error rate for the negative class and the PB is the lowest score whose E^+ is less than the given error rate for the positive class; all other samples are rejected because the decision is too risky. Obviously, the nature of the dataset affects these bounds, therefore, PB and NB could be used as a quantitative method for the assessment of the classification task. The computation of the NB and the PB are detailed in next section.

4 Empirical Data Analysis and Results

Our SPECT database consists of 25 migraine patients (13 subjected to individualized active or verum acupuncture and 12 with minimal or sham acupuncture. For each patient, SPECT image is acquired as a baseline 20 minutes following the injection of ^{99m}Tc -ECD. After this initial image acquisition the patients are given the acupuncture session (verum or sham) under the procedure explained in [11] and finally, post image acquisition is acquired 30 minutes later the acupuncture session. All pre-acupuncture acquisitions are considered as belonging to

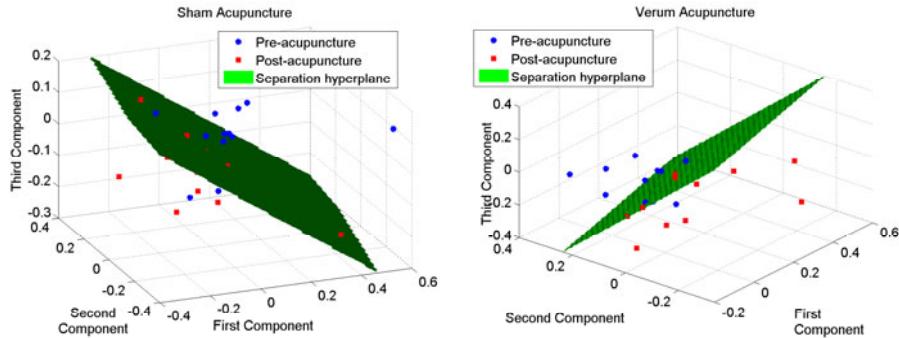


Fig. 1. Distribution of PLS features extracted from images and decision hyperplane designed by the SVM. Verum acupuncture provides more easily separable features in comparison with sham acupuncture.

class ω_1 and post-acupuncture scans to class ω_2 . Images are also separated in group G1 or G2 if the acupuncture modality was verum or sham, respectively. Thus, each volume is included in one of these groups: $X_{\omega_1}^{G1}, X_{\omega_2}^{G1}, X_{\omega_1}^{G2}$ and $X_{\omega_2}^{G2}$.

The aim of the experiments is to demonstrate the effects of the application of acupuncture on the patients by evaluating the performance of the classifier and by computing the NB and PB for two different sets of samples: $X_1 = \{\mathbf{x}, |\mathbf{x} \in (X_{\omega_1}^{G1}, X_{\omega_2}^{G1})\}$ and $X_2 = \{\mathbf{x}, |\mathbf{x} \in (X_{\omega_1}^{G2}, X_{\omega_2}^{G2})\}$. If acupuncture is effective in migraine patients, then class separation between pre- and post-acupuncture acquisitions must be easier for samples included in G1 (verum acupuncture) than for those in G2 (sham acupuncture). In order to reduce the computational cost of the algorithms, the image space is first reduced by the application of PLS so that the previous vectors \mathbf{x} refer to the extracted PLS scores, which will be used as features to train the SVM.

Let us be $ErrorNeg$ and $ErrorPos$ the maximum error rates allowed for class negative and positive, respectively. The complete algorithm for computing the bounds of confidence is given below:

Bounds Computing Algorithm. Given a training set \mathbf{X} and labels y :

1. Make feature vectors \mathbf{x}_i up from PLS application to \mathbf{X} .
2. Find all the support vectors whose class is positive and whose score is negative, denoted as SVPN.
3. Rank in ascending order the scores of SVPN, denoted as S-SVPN and its size as NN
4. $S\text{-SVPN}(NN + 1) = 0, i = 1$
5. Do
 - NB = S-SVPN(i)
 - Compute E^- of S-SVPN(i)
 - $i = i + 1$
 - While $E^- \leq ErrorNeg$

6. If $i \leq NN + 1$, $NB = S\text{-SVPN}(i - 2)$.
7. Perform similar steps from 2 to 6 to find the PB.

The algorithm above is first applied to group X_1 and secondly to group X_2 . Table 1 shows the NB and PB computed for each group X_1 and X_2 when varying the maximum error allowed from 0.1 to 1, and setting $ErrorNeg = ErrorPos = \epsilon$. Bounds computed for X_1 are in most cases smaller than bounds computed for X_2 when the same probability error of misclassification is set, i.e., the risky zone is wider for sham acupuncture samples compared to verum modality for a given probability error. This measure can be interpreted as a greater separation among pre- and post-acupuncture activation brain patterns of features extracted from patients to whom verum acupuncture was applied. This fact highlights significant effects on patients that underwent verum acupuncture in comparison to patients to whom sham acupuncture was applied.

Table 1. [NB, PB] computed for X_1 and X_2 and different values of $ErrorNeg = ErrorPos = \epsilon$

ϵ	X_1	X_2
0.1	[-0.45, 0]	[-0.62, 0]
0.2	[-0.45, 0]	[-0.62, 0]
0.3	[-0.45, 0]	[-0.62, 0]
0.4	[-0.45, 0.18]	[-0.62, 0]
0.5	[-0.45, 0.18]	[-0.62, 0.18]
0.6	[-0.45, 0.18]	[-0.62, 0.18]
0.7	[-0.45, 0.15]	[-0.62, 0.18]
0.8	[-0.45, 0.15]	[-0.62, 0.18]
0.9	[-0.45, 0.15]	[-0.62, 0.18]
1	[-0.45, 0.15]	[-0.62, 0.18]

Apart from the PB and NB, we also presents the accuracy rates obtained by the classifier when pre- and post-acupuncture acquisitions are analyzed for each group in a classification context. Figure 2 plots the 3 first PLS coefficients extracted from the images as 3D points, and the separation hyperplane designed by a linear SVM classifier. Visually it can be seen that classes become more easily separable for verum acupuncture. If test samples are drawn form different probability distribution functions (pdf) – related to significantly different perfusion patterns, the classification rates should be large unlike if samples correspond to the same pdf. In the latter case, the accuracy rate in the classification task performed by an SVM should be near to 0.5 (random classifier). This procedure is also expected to report different results using verum or sham acupuncture in the treatment of migraine.

Figure 2 shows values of accuracy, sensibility and specificity obtained for each group when the number of PLS extracted coefficients increases. Due to the small number of samples that make each group up, leave-one-out cross validation strategy was used to train and test the classifier and to compute the averaged final

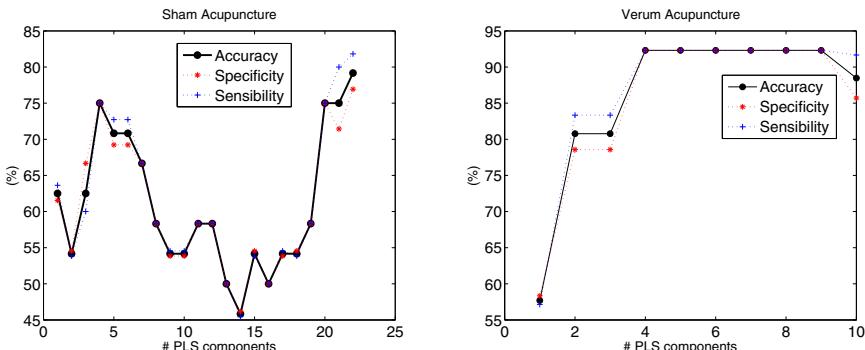


Fig. 2. Accuracy, sensibility and specificity values obtained when linear SVM is used to classify PLS features extracted from pre-acupuncture and post-acupuncture image brain volumes. The classifier is able to distinguish greater differences in perfusion patterns when verum acupuncture is applied (right) compared to sham acupunctre (left)

values. Accuracy curves support the conclusions extracted from the bounds of confidence in terms of better performance of the classifier in distinguishing pre- and post-acupuncture acquisitions as two different classes. For verum acupuncture separation between classes is reached with more than 90% accuracy while sham acupuncture acquisitions are badly separated. This fact highlights the presence of acupuncture effects on patients by changing the perfusion pattern that can be measured by means of PB and NB.

5 Conclusions

In this work, acupuncture effects produced on migraine patients are measured by means of PLS and SVM with bounds of confidence using brain SPECT images. PLS features extracted from patients to whom verum acupuncture was applied provided more easily separable classes compared to sham modality samples, which was quantified by computing the NB and PB of confidence. On the other hand, the computed bounds matched the performance of the SVM in terms of accuracy rate when pre- and post-acupuncture PLS features are separated in a classification task, yielding good separability results for verum modality while the classifier responds to sham acupuncture samples as a random classifier. These results highlight significant effects on patients that underwent verum acupuncture in comparison to patients to whom sham acupuncture was applied.

Acknowledgments. This work was partly supported by the Spanish Government under the PETRI DENCLASES (PET2006-0253), TEC2008-02113, NA-POLEON (TEC2007-68030-C02-01) projects and the Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía, Spain) under the Excellence Projects TIC-02566 and TIC-4530. The SPECT database was provided by “Virgen de las Nieves” hospital in Granada (Spain).

References

1. Battistella, P.A., Ruffilli, R., Dalla Pozza, F., Pitassi, I., Casara, G.L., Boniver, C., Bendagli, A., Condini, A.: *99mTc HM-PAO SPECT in pediatric migraine. Headache: The Journal of Head and Face Pain* 30(10), 646–649 (1990)
2. Burges, C.J.C.: A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2), 121–167 (1998)
3. Chaves, R., Ramírez, J., Górriz, J., López, M., Salas-Gonzalez, D., Álvarez, I., Segovia, F.: SVM-based computer-aided diagnosis of the Alzheimer's disease using t-test NMSE feature selection with feature correlation weighting. *Neuroscience Letters* 461(3), 293–297 (2009)
4. Corchado, E., Abraham, A., de Carvalho, A.C.: Hybrid intelligent algorithms and applications. *Information Science* 180(4), 2633–2634 (2010)
5. Friston, K.J., Ashburner, J., Kiebel, S.J., Nichols, T.E., Penny, W.D.: *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, London (2007)
6. Li, M., Sethi, I.K.: Confidence-based classifier design. *Pattern Recognition* 39(7), 1230–1240 (2006)
7. Pomeranz, B., Stux, G.: *Scientific Basis of Acupuncture*. Springer, Berlin (1989)
8. Ramírez, J., Górriz, J., Romero, A., Lassl, A., Salas-Gonzalez, D., López, M., Río, M.G.: Computer aided diagnosis of Alzheimer type dementia combining support vector machines and discriminant set of features. *Information Sciences* (2009), doi:10.1016/j.ins.2009.05.012 (in press)
9. Salas-Gonzalez, D., Górriz, J.M., Ramírez, J., Lassl, A., Puntonet, C.G.: Improved Gauss-Newton optimization methods in affine registration of SPECT brain images. *IET Electronics Letters* 44(22), 1291–1292 (2008)
10. Saxena, P., Pavel, D.G., Quintana, J.C., Horwitz, B.: An automatic threshold-based scaling method for enhancing the usefulness of tc-HMPAO SPECT in the diagnosis of alzheimer#146s disease. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) *MICCAI 1998. LNCS*, vol. 1496, pp. 623–630. Springer, Heidelberg (1998)
11. Vas, J., Modesto, M., Méndez, C., Perea-Milla, E., Aguilar, I., Carrasco-Lozano, J., Faus, V., Martos, F.: Effectiveness of acupuncture, special dressings and simple, low-adherence dressings for healing venous leg ulcers in primary healthcare: study protocol for a cluster-randomized open-labeled trial. *BMC Complementary and Alternative Medicine* 8(29) (2008), doi:10.1186/1472-6882-8-29
12. Wold, H.: Partial least squares. *Encyclopedia of Statistical Sciences* 6, 581–591 (1985)
13. Wozniak, M., Zmyslony, M.: Designing fusers on the basis of discriminants – evolutionary and neural methods of training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) *HAIS 2010. LNCS*, vol. 6076, pp. 590–597. Springer, Heidelberg (2010)

An Intelligent Automated Recognition System of Abnormal Structures in WCE Images

Piotr Szczypinski, Artur Klepaczko, and Michal Strzelecki

Technical University of Lodz, Institute of Electronics
ul. Wolczanska 211/215, 90-924 Lodz, Poland
{pms,aklepaczko,mstrzel}@p.lodz.pl

Abstract. In this paper we study the problem of classification of wireless capsule endoscopy images (WCE). We aim at developing a computer system that would aid in medical diagnosis by automatically detecting images containing pathological alterations in an 8-hour-long WCE video. We focus on three classes of pathologies – ulcers, bleedings and petechia – since they are typical for several diseases of the intestines. The main contribution is the performance evaluation of five feature selection and classification algorithms: minimization of classification error probability, Vector Supported Convex Hull, Support Vector Machines, Radial Basis Function and Perceptron-based Neural Networks, in application to WCE images. Experimental results show that none of the methods ouperforms the others in all tested pathology classes. Instead, a classifier ensemble can be built to accumulate evidence from multiple learning schemes, each specialized in recognition of a single type of abnormality.

Keywords: Wireless capsule endoscopy, feature selection, convex hull, support vector machines, artificial neural networks.

1 Introduction

Wireless capsule endoscopy (WCE) [7][16] is a non-invasive technique that allows visualization of the whole human small intestine. The WCE system consists of a pill-shaped capsule with a built-in video camera, light-emitting diodes, video signal transmitter and battery, as well as a video signal receiver-recorder device. The wireless capsule endoscope used in this study produces color images of the internal lumen. One of the many drawbacks of WCE is that the interpretation of the video sequence produced by the capsule endoscope demands significant effort and remains a time consuming procedure. This research aims at development of an intelligent system that would allow automatic detection of WCE video frames showing pathological changes and thus supporting medical diagnosis (Fig. 1). Such systems have been already proposed for computer-aided evaluation of endoscopic images or video sequences, including detection and classification of different pathologies such as polyps, tumors, intestine adenomas, ulcer, and bleeding [2][9][6]. The detailed survey of various approaches to detection of abnormalities in WCE images is presented in [8]. According to

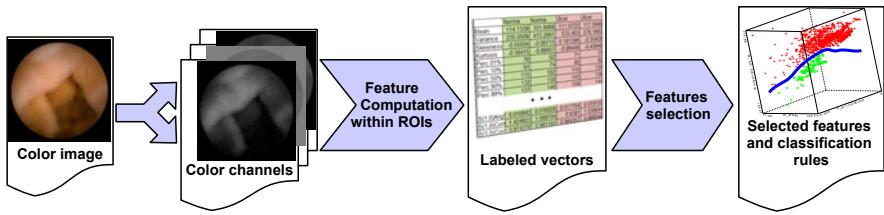


Fig. 1. Diagram of the proposed system

the conclusion of this survey, the most promising methods utilize the color- and image-processing such as color transformations [3][4], texture analysis [1][13] or both approaches [12] along with tools based on machine learning capabilities, such as artificial neural networks of different architectures. Recently, much attention is paid to the integration of different learning and adaptation techniques to achieve a synergy by fusion of these techniques in the framework of hybrid intelligent systems [5].

Since the analysis of WCE images generates a multidimensional feature space, it is difficult to arbitrarily choose a particular model representative for pathological tissue color and texture. Limitation of the large feature set to a few most relevant and least redundant features improves the classification process, not only by reducing the computational requirements, but also by leading to more accurate classification results. However, in the literature there is a lack of research focusing on feature selection methods for classification of WCE images. The objective of this paper is to compare five such methods: minimization of classification error probability along with correlation coefficient between selected features (POE) [14], Support Vector Machines (SVM) with the radial kernel, Radial Basis Function Networks (RBFN), Multilayer Perceptron Neural Networks (MPNN) and the new approach formulated by the authors based on Vector Supported Convex Hull (VSCH) [18]. Evaluation of the investigated techniques allows designing of a hybrid system implementing a combination of feature selection and classification methods providing better system performance [20].

The remainder of this paper is structured as follows. The notion of texture analysis and the software tools used in the research are described in Sect. 2. In Sect. 3 we recall the general principles of the feature subspace selection methods. In the experimental part (Sect. 4), the efficiency of VSCH is tested in comparison with SVM, RBFN, POE and MPNN. Finally, Sect. 5 presents obtained analysis results along with discussion, while the Sect. 6 concludes the paper.

2 Texture Analysis

A texture can be viewed as a complex composition of spatially organized, repeated patterns, which are perceived to demonstrate specific brightness, color, size, roughness, directivity, randomness, smoothness, granulation, etc. A texture may carry substantial information about the structure of physical objects — human tissues or organs in particular. Consequently, textural image analysis is

an important issue in image processing and understanding, also in medical applications. To perform such analysis, mathematically defined texture properties are computed.

In our study we use the MaZda 4.7 software [19][17] for textural feature computation. The software is capable of conducting a quantitative analysis of texture within arbitrarily selected regions of interest (ROI). There are three categories of feature computation approaches that MaZda utilizes: statistical (based on image histogram, gradient, co-occurrence matrix, run-length matrix), model-based (implementation of the autoregressive model) and image transform (based on the Haar wavelet). MaZda may be used to compute textural descriptors based on color components of RGB, YUV, YIQ, HSB, XYZ and L*a*b color spaces. The textural features computed for different color components can be combined to obtain a comprehensive characterization of a colored texture. Therefore, feature vectors computed by MaZda may include even 3000 elements per individual region of interest. The need for dimensionality reduction becomes indispensable.

3 Vector Supported Convex Hull Method

The VSCH is a discriminant analysis method of supervised learning for reduction of vectors dimensionality and for data classification. Usually only a limited number of features carry relevant information needed for discrimination. The VSCH aims at finding a subset of descriptors, which present best discrimination ability to separate two classes (sets) of vectors. Moreover the VSCH produces a classification rule to separate the two classes.

To explain the concept of VSCH let us assume input data consist of two sets (classes) of feature vectors in an n -dimensional space. All the features are real numbers. We search for a k -dimensional subspace ($k < n$) such that vectors of the set number one form a cluster surrounded by vectors of the set number two. Let us consider a convex hull of set one in a k -dimensional subspace of feature vectors space. There are several algorithms for finding the convex hull of sets of vectors. In our implementation we use the quick-hull algorithm [1]. Now we define a coefficient Q_1 . It is the number of vectors belonging to the second class, which also belong to the convex hull built on class number one. The next step is to find a centroid (or gravity center) of the convex hull. Then the convex hull is isotropically scaled up around the fixed centroid. We find the maximum scaling factor a for which Q_1 does not increase. Now we define a coefficient Q_2 , which is equal to the reciprocal of a . Since a is larger than 1, Q_2 is a fraction. On the other hand Q_1 is an integer number equal or higher than 0. Now, we combine the two coefficients and define a comprehensive Q factor as:

$$Q = Q_1 + Q_2. \quad (1)$$

The Q factor specifies the discriminative power of the k -dimensional feature space. A low value of Q indicates the analyzed feature subspace has better class separability. The algorithm for feature space reduction based on VSCH searches through all the 1D, 2D, 3D and 4D feature subsets and computes Q for each

subset. The subset with the lowest Q is chosen for further analysis and classification. The algorithm also produces rules for classification in the form of a set of inequalities defining the convex hull scaled up by factor of \sqrt{a} . In many medical applications it is crucial not to overlook any indications of pathology. Such indications usually are later verified by medical experts and may be rejected. If they are mistakenly rejected by an automatic method, an expert may never notice the pathology. Therefore, it is important to find methods characterized by a minimal false negative error. The VSCH method reveals a property that is particularly useful in biomedical image analysis. The method produces classification rules, for which (for the training set vectors) the false negative error is equal to zero. The minimization of false positive errors is a secondary goal, which is achieved directly by minimization of Q_1 .

4 Experiment

In the experimental part of this study we analyzed a set 20 WCE video sequences from which we selected 30 frames showing different forms of bleedings, 50 frames with ulcerations, 100 images of petechia and over 600 images with no visible pathology structures. The latter group served as a reference class in the analyzed discrimination problems. Regions of bleeding, ulceration and petechia (regions of interest) were manually depicted within the images. Then all the selected frames were divided into circular overlapping subregions, each of 2009 pixels area (Fig. 2a). For images showing abnormal changes, textural features were computed within circular subregions enclosed within the depicted regions of interest. For reference images textural features were computed within circular subregions enclosed within the whole image field of view. The number of calculated feature vectors amounted to 128 for each pathology and 1024 in the case of reference images. These vectors were then combined into three 2-class data sets, each containing one class representing abnormal tissues and the reference group. The data sets were first analyzed by the VSCH method and then by feature selection and classification procedures that employed SVM, RBF and MPNN network algorithms. In the three latter cases, feature space exploration was performed using exhaustive search [2][10][15] and relevance of a feature subset was assessed by the accuracy of the corresponding classifier. In the case of the

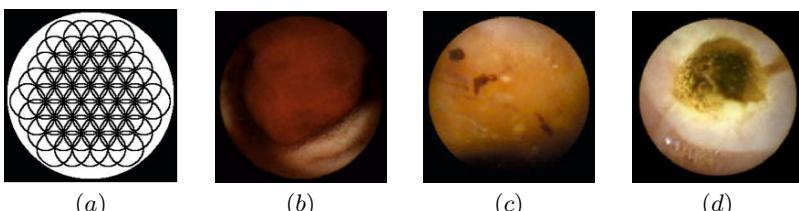


Fig. 2. Overlapping circular regions used in the study (a) and sample images of analyzed pathologies: bleeding (b), petechia (c), and ulcer (d)

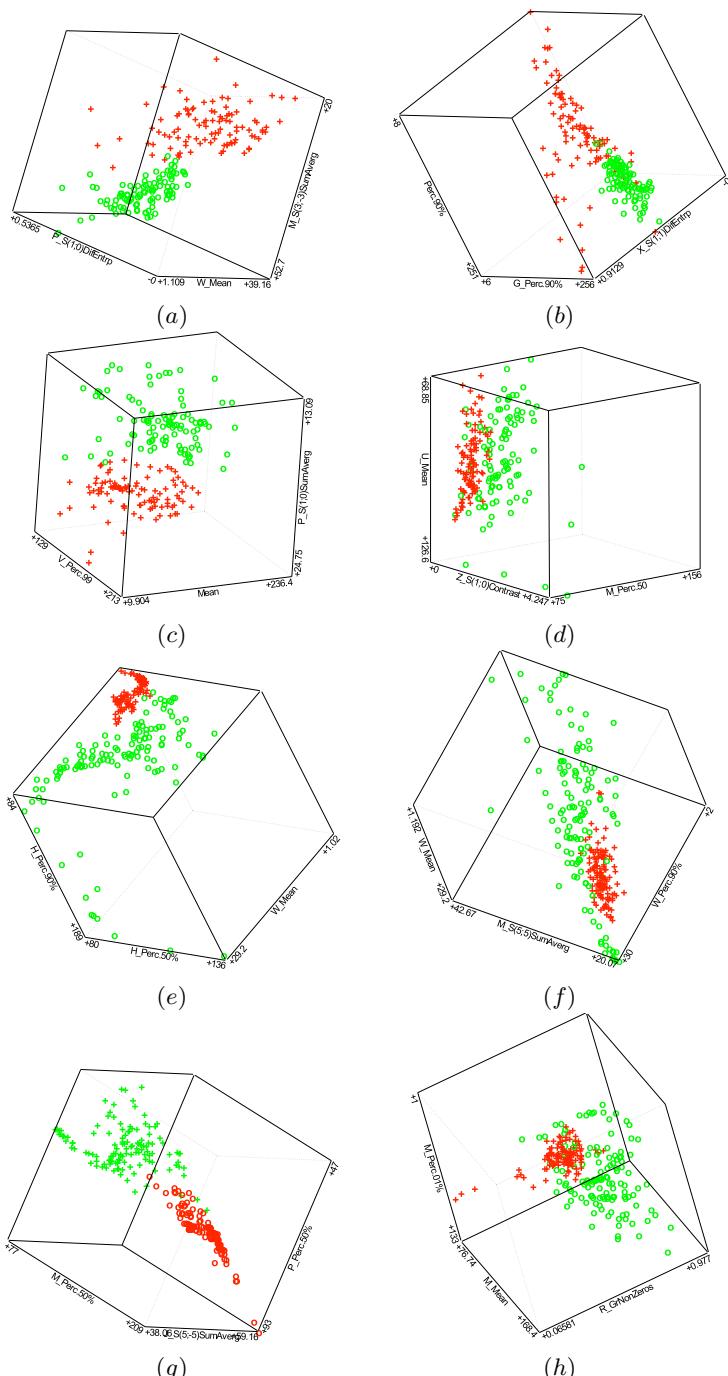


Fig. 3. Scatter plots of data vectors in the best feature spaces selected by different methods for bleeding vs. normal: VSCH (a), RBFN (c), MPNN (e), POE (g) and for ulcer vs. normal: VSCM (b), RBFN (d), MPNN (f), POE (h)

POE feature selection method, the MPNN was used for classification. The classification specificity and sensitivity computed for the training and test data sets are presented in Table 1. Sample scatter plots of feature vectors in the selected texture parameter spaces are shown in Fig. 3.

Table 1. Classification results for training and testing sets

		Training set				
Pathology type	Evaluation criterion	Algorithm				
		VSCH	MPNN	POE	SVM	RBFN
Ulcer	Specificity	0.988	0.977	0.969	0.945	0.956
	Sensitivity	1.000	0.984	1.000	1.000	0.969
Bleeding	Specificity	1.000	1.000	1.000	0.981	1.000
	Sensitivity	1.000	1.000	1.000	1.000	1.000
Petechiae	Specificity	0.966	0.977	0.984	0.936	0.891
	Sensitivity	1.000	0.969	0.992	0.984	0.981
Testing set						
Ulcer	Specificity	1.00	0.95	0.90	0.88	0.76
	Sensitivity	0.84	0.82	0.90	0.98	0.93
Bleeding	Specificity	0.99	0.98	0.78	0.97	0.99
	Sensitivity	0.98	0.99	1.00	1.00	1.00
Petechiae	Specificity	0.77	0.81	0.82	0.94	0.87
	Sensitivity	0.82	0.79	0.83	0.90	0.92

5 Results and Discussion

Analysis of the obtained results leads to the following conclusions. First of all, the performed experiments confirm that texture analysis provides a practical numerical description of the WCE images. It is possible to classify different types of visualized tissues basing on the selected, most relevant texture parameters. Among the calculated attributes, color component features appear to be the best at discriminating abnormal and normal regions. Secondly, the sensitivity and specificity measures shown in Table 1 are comparable for all five of the tested approaches to feature selection and classification. The POE method leads to less accurate classification, especially in the case of training sets. The VSCH method appears to be overoptimistic when predicting the specificity ratio on the training sets. This results directly from the intrinsic nature of the algorithm that aims at construction of a convex hull around all vectors from a chosen pathology class. We expect this error to be larger than zero if another set of vectors is used for verification.

Implementation of both SVM and RBFN involves problem-specific parameterization of the kernel function. Frequently, one must experiment with several

values of power exponents (both in polynomial or radial basis functions) before a final choice can be made. Application of MPNN requires appropriate selection of the number of neurons in the hidden layer. On the other hand, VSCH is a non-parametric method and does not require any fine-tuning to solve particular tasks. Moreover, it does not require any feature space standardization. Also any other linear transformation of the feature space has no influence on the result produced by the method.

6 Conclusion

This paper presents an intelligent system for automated recognition of abnormal structures in WCE images based on the most discriminative texture descriptors. The system employs analysis of both texture and color descriptors to ensure a discrimination between normal and pathological tissues. Five techniques of feature selection and classification were compared and discussed. It was demonstrated that the new feature selection method based on the Vector Supported Convex Hull provides accurate classification of tested pathologies in WCE images. Particularly, it provides a high sensitivity rate ensuring that images containing pathological changes will not be classified as healthy. This is very important in medical diagnosis; however, the VSCH method should be tested on larger data sets to confirm its good performance. The advantage of the proposed approach when compared to artificial neural network based techniques and the SVM method is much shorter computation time required for selection of feature subset (420s for VSCH versus 2200-2400s in the case of RBFN, MPNN and SVM using Core 2 Duo Intel microprocessor).

Currently, the proposed system is tested in the Gastroenterology Department, Medical University of Lodz. Further works focus on detection of a larger number of pathologies (e.g. polyps, adenomas), discrimination of different types of a given pathology (e.g. different kinds of ulcers or bleeding) and construction of a hybrid system combining selected classifier tools.

Acknowledgments. The WCE videos used in this study were provided by Dr. D. Nageswar Reddy, Asian Institute of Gastroenterology, Hyderabad, India. The authors would like to thank Ms. Monika Kołodziejska and Dr. Piotr Daniel for manual segmentation of images.

References

1. Barber, C.B., Dobkin, D.P., Huhdanpaa, H.T.: The quickhull algorithm for convex hulls. *The Quickhull Algorithm for Convex Hulls* 22(4), 469–483 (1996)
2. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. *Artificial Intelligence* 97, 245–271 (1997)
3. Coimbra, M., Campos, P., Cunha, J.P.S.: Extracting clinical information from endoscopic capsule exams using mpeg-7 visual descriptors. In: *The 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, EWIMT 2005.*, pp. 105–110 (2005)

4. Coimbra, M.T., Cunha, J.P.S.: Mpeg-7 visual descriptors–contributions for automated feature extraction in capsule endoscopy. *IEEE Transactions on Circuits and Systems for Video Technology* 16(5), 628–637 (2006)
5. Corchado, E., Abraham, A., de Carvalho, A.C.P.L.F.: Hybrid intelligent algorithms and applications. *Information Science* 180, 2633–2634 (2010)
6. Iakovidis, D.K., Maroulis, D.E., Karkanis, S.A.: An intelligent system for automatic detection of gastrointestinal adenomas in video endoscopy. *Comput. Biol. Med.* 36(10), 1084–1103 (2006)
7. Iddan, G., Meron, G., Glukhowsky, A., Swain, P.: Wireless capsule endoscopy. *Nature* 405(6785), 417–418 (2000)
8. Karargyris, A., Bourbakis, N.: Capsule endoscopy and endoscopic imaging: A survey on various methodologies presented. *IEEE Engineering in Medicine and Biology Magazine* 29(1), 72–83 (2010)
9. Kodogiannis, V.S., Boulougoura, M., Lygouras, J.N., Petrounias, I.: A neuro-fuzzy-based system for detecting abnormal patterns in wireless-capsule endoscopic images. *Neurocomputing* 70(4-6), 704–717 (2007)
10. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial Intelligence* 97, 273–324 (1997)
11. Li, B., Meng, M.H.: Ulcer recognition in capsule endoscopy images by texture features. In: 7th World Congress on Intelligent Control and Automation, WCICA 2008, pp. 234–239 (June 2008)
12. Mackiewicz, M., Berens, J., Fisher, M.: Wireless capsule endoscopy video segmentation using support vector classifiers and hidden Markov models. In: Proceedings of the Int. Conference on Medical Image Understanding and Analyses (2006)
13. Mackiewicz, M., Berens, J., Fisher, M., Bell, G.D.: Colour and texture based gastrointestinal tissue discrimination. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 597–600 (2006)
14. Mucciardi, A., Gose, E.: A comparison of seven techniques for choosing subsets of pattern recognition properties. *IEEE Trans. on Computers* 20, 1023–1031 (1971)
15. Pudil, P., Somol, P.: Current feature selection techniques in statistical pattern recognition. In: Kurzynski, M., et al. (eds.) Computer Recognition Systems. Advances in Soft Computing, pp. 53–70. Springer, Heidelberg (2005)
16. Swain, P., Fritscher-Ravens, A.: Role of video endoscopy in managing small bowel disease. *GUT* 53, 1866–1875 (2004)
17. Szczypinski, P.: (2010), <http://www.elete1.p.lodz.pl/Mazda>,
<http://www.elete1.p.lodz.pl/Mazda> (visited, May 2010)
18. Szczypinski, P., Klepaczko, A.: Convex hull-based feature selection in application to classification of wireless capsule endoscopic images. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2009. LNCS, vol. 5807, pp. 664–675. Springer, Heidelberg (2009)
19. Szczypinski, P., Strzelecki, M., Materka, A., Klepaczko, A.: Mazda - a software package for image texture analysis. *Comp. Methods Prog. Biomed.* 94, 66–76 (2009)
20. Wozniak, M., Zmyslony, M.: Designing fusers on the basis of discriminants – evolutionary and neural methods of training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 590–597. Springer, Heidelberg (2010)
21. Zhang, S., Yang, W., Wu, Y.-L., Yao, R., Cheng, S.-D.: Abnormal region detection in gastroscopic images by combining classifiers on neighboring patches. In: Proc. 8. Int. Conf. Machine Learning and Cybernetics, Baoding, pp. 2374–2379 (2009)

Effective Diagnosis of Alzheimer's Disease by Means of Distance Metric Learning

R. Chaves, J. Ramírez, J.M. Górriz, D. Salas-Gonzalez, M. López,
I. Illán, F. Segovia, and A. Olivares

University of Granada, Periodista Daniel Saucedo Aranda s/n,
18071, Granada (Spain)
{rosach, javierrp, gorriz,
dsalas, miriamlp, illan, fsegovia, aolivares}@ugr.es
<http://sipba.ugr.es/>

Abstract. In this paper we present a novel classification method of SPECT images for the early diagnosis of the Alzheimer's disease (AD). The proposed method is based on distance metric learning classification with the Large Margin Nearest Neighbour algorithm (LMNN) aiming to separate examples from different classes (Normal and AD) by a large margin. In particular, we show how to learn a Mahalanobis distance for k-nearest neighbors (KNN) classification. It is also introduced the concept of energy-based model which outperforms both Mahalanobis and Euclidean distances. The system combines firstly Normalized Minimum Square Error (NMSE) and t-test selection with secondly Kernel Principal Components Analysis (KPCA) to find the main features. Applying KPCA trick in the feature extraction, LMNN turns into Kernel-LMNN (KLMNN) with better results than the first. KLMNN reaches results of accuracy=96.91%, sensitivity=100% ,specificity=95.35% outperforming other recently reported methods such as Principal Component Analysis(PCA) in combination with Linear Discriminant Analysis (LDA) evaluated with Support Vector Machines (SVM) or linear SVM.

Keywords: SPECT Brain Imaging, Alzheimer's disease, Distance Metric Learning, Kernel Principal Components Analysis.

1 Introduction

Alzheimer's Disease (AD) is the most common cause of dementia in the elderly and affects approximately 30 million individuals worldwide [1]. Its prevalence is expected to triple over the next 50 years due to the growth of the older population. To date there is no single test or biomarker that can predict whether a particular person will develop the disease. With the advent of several effective treatments of AD symptoms, current consensus statements have emphasized the need for early recognition.

SPECT (Single Positron Emission Computed Tomography) is a widely used technique to study the functional properties of the brain [2]. After the reconstruction and a proper normalization of the SPECT raw data, taken with Tc-99m

ethyl cysteinate dimer (ECD) as a tracer, one obtains an activation map displaying the local intensity of the regional cerebral blood flow (rCBF). Therefore, this technique is particularly applicable for the diagnosis of neuro-degenerative diseases like AD.

In order to improve the prediction accuracy especially in the early stage of the disease, when the patient could benefit most from drugs and treatments, computer aided diagnosis (CAD) tools are desirable. At this stage in the development of CAD systems, the main goal is to reproduce the knowledge of medical experts in the evaluation of a complete image database, i.e. distinguishing AD patients from controls, thus errors from single observer evaluation are avoided achieving a method for assisting the identification of early signs of AD.

In the context of *supervised* multivariate approaches, the classification is usually done by defining feature vectors representing the different SPECT images and training a classifier with a given set of known samples [3]. Firstly, the feature selection [4] consists of a combination of 3D cubic NMSE (Normalized Minimum Square Error) features over regions of interest (ROIs) that are selected by a t-test feature selection with feature correlation weighting. Secondly, kernel principal component analysis (KPCA) is applied on the previous step extracted features as dimension reduction to a lower subspace, which is subsequently used to train a distance metric learning-based classifier. Some techniques were developed to learn weights of features to change the distance structure of samples in nearest neighbor classification. Euclidean distance, the most commonly used, assumes that each feature is equally important and independent from others. By contrast, a distance metric with good quality such as Mahalanobis, should identify relevant features assigning different weights or importance factors to the extracted ROIs [5]. Only when the features are uncorrelated, the distance under a Mahalanobis distance metric is identical to that under the Euclidean distance metric. On the other hand, our work has been inspired by energy-based metric learning, obtaining with it the best results in terms of accuracy, specificity and sensitivity. Many issues of image analysis can be modeled and coped by designing an energy function; $U(x,y)$ which captures the interaction between the unknown variables $x = (x_i)_i$ to be estimated, and the observed variables (the measurements or data), $y = (y_j)_j$. A standard, although complex in general, problem is then the minimization of this function with respect to x, y being known. Other intricate issues, such as estimation of internal parameters or validation of selected models, also arise within this framework [6].

2 Materials and Methods

2.1 Subjects and Preprocessing

Baseline SPECT data from 97 participants were collected from the Virgen de las Nieves hospital in Granada (Spain). The patients were injected with a gamma emitting ^{99m}Tc -ECD radiopharmaceutical and the SPECT raw data was acquired by a three head gamma camera Picker Prism 3000. A total of 180 projections were taken with a 2-degree angular resolution. The images of the brain

cross sections were reconstructed from the projection data using the filtered backprojection (FBP) algorithm in combination with a Butterworth noise removal filter. The SPECT images are first spatially normalized using the SPM software, in order to ensure that voxels in different images refer to the same anatomical positions in the brain allowing us to compare the voxel intensities of different subjects. Then we normalize the intensities of the SPECT images with a method similar to [7]. After the spatial normalization, one obtains a $95 \times 69 \times 79$ voxel representation of each subject, where each voxel represents a brain volume of $2 \times 2 \times 2 \text{ mm}^3$. The SPECT images were visually classified by experts of the Virgen de las Nieves hospital using 4 different labels (43 NOR, 30 AD1, 20 AD2 and 4 AD3) to distinguish between different levels of the presence of typical characteristics for AD. However our CAD system was designed for two classes (NOR and AD) because of its usefulness for the early diagnosis of AD.

3 Feature Extraction

In this article, we propose to apply a combination of VAF (Voxels as Features), NMSE and Kernel PCA. First of all, controls are averaged in a tridimensional image ($sm(x, y, z)$) as it is shown in figure 1. In functional imaging, each voxel carries a grey level intensity $I(x_j)$, which is related to the regional cerebral blood flow, glucose metabolism, etc. in the brain of a patient, depending on the image acquisition modality. Secondly, it is obtained a 3D $mask(x, y, z)$ that consists of all the voxels with $sm(x, y, z) > a_T$. The threshold a_T is equivalent to the 50% of the maximum Intensity in $sm(x, y, z)$. Baseline VAF is a way of including in $vaf(x, y, z)$ all the voxels inside the obtained $mask(x, y, z)$ and considering them as features. Therefore, voxels outside the brain and poorly activated regions are excluded from this analysis. In this way, no explicit knowledge about the disease is needed, avoiding the inclusion of a priori information about the pathology into the system. Each SPECT is following divided into 3D $v \times v \times v$ cubic

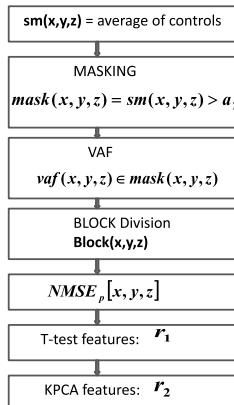


Fig. 1. Feature Extraction Process

voxels defining Regions of Interest (ROIs), or $\text{block}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ centered in $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ coordinates which are inside of $\text{vaf}(\mathbf{x}, \mathbf{y}, \mathbf{z})$, as it is referenced in figure 1. Then, it is calculated the Normalized Minimum Squared Error or $NMSE_p(\mathbf{x}, \mathbf{y}, \mathbf{z})$ for each subject and block. NMSE is given by equation 11 as:

$$NMSE_p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \frac{\sum_{l,m,n=-v}^v [f(\mathbf{x} - l, \mathbf{y} - m, \mathbf{z} - n) - g_p(\mathbf{x} - l, \mathbf{y} - m, \mathbf{z} - n)]^2}{\sum_{l,m,n=-v}^v [f(\mathbf{x} - l, \mathbf{y} - m, \mathbf{z} - n)]^2} \quad (1)$$

where $f(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is the mean voxel intensity at $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, and $g_p(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is the voxel intensity of the p subject. The most discriminant ROIs (r_1 in figure 1) are obtained with an absolute value two-sample t-test-based with pooled variance estimate on NMSE features as in [8]. The latter method, PCA is a multivariate approach often used in neuroimaging to significantly reduce the original high-dimensional space of the original brain images to a lower dimensional subspace [9] called r_2 in figure 1. PCA aims to find the projection directions that maximize the variance of a subspace, which is equivalent to find the eigenvalues from the covariance matrix. PCA can be used in combination with the so-called kernel methods [10]. The basic idea of the kernel PCA method is to first preprocess the data by some non-linear mapping and then to apply the same linear PCA. In kernel PCA (KPCA), each vector \mathbf{x} is projected from the input space, \mathbb{R}^n , to a high-dimensional feature space, \mathbb{R}^f , by a non-linear mapping function: $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^f$, $f > n$. Note that the dimensionality of the feature space can be arbitrarily large. When we use the KPCA trick framework, the original LMNN (Large Margin Nearest Neighbour, explained in the following section) can be immediately used as Kernel-LMNN (also called KLMNN) [11]. At the end of the extraction, the reduction of features r_2 obtained with KPCA is used as the input features for the LMNN classifier.

4 Large Margin Nearest Neighbor Classification

In this paper, we show how to learn a Mahalanobis distance for kNN classification. Previous approaches minimize the pairwise distances between all similarity labeled examples [12]. This metric is aimed at the organisation of the k-nearest neighbors to the same class, while examples from different classes are separated by a large margin. This technique is called *large margin nearest neighbor* (LMNN) classification. LMNN can be considered as the logical counterpart to SVMs in which KNN classification replaces linear classification. In addition, this approach is largely inspired by recent work on neighborhood component analysis [13] and metric learning by energy-based inspired [14]. Large Margin principle [15] was used to design classification algorithms. There is a cost function [12] that favors distance metric in which different labeled inputs maintain a large margin of distance, that is, while the first term of it penalizes with the idea of margin, large distances between similarly labeled inputs (target neighbors), the second term penalizes small distances between differently labeled inputs.

4.1 Model

A training set of n labeled examples is denoted $\{(\mathbf{x}_i, y_i)\} i = 1, \dots, n$, with inputs $\mathbf{x}_i \in \mathbb{R}^d$ and class labels y_i . The goal is to learn a linear transformation, $\mathbf{L}: \mathbb{R}^d \rightarrow \mathbb{R}^d$ that optimizes kNN classification when squared distances are computed in this way:

$$D(\mathbf{x}_i, \mathbf{x}_j) = \|L(\mathbf{x}_i - \mathbf{x}_j)\|^2 \quad (2)$$

Given two data points $x_1 \in \mathbb{R}^n$ and $x_2 \in \mathbb{R}^n$, the Mahalanobis distance can be calculated as follows [5]:

$$d_M(x_1, x_2) = \sqrt{(x_1 - x_2)^T \cdot M \cdot (x_1 - x_2)} \quad (3)$$

where $M \in \mathbb{R}^*$ where \mathbb{R}^* is $n \times n$ dimensional and positively semi-definite. Using the eigenvalue decomposition, M can be decomposed into $M = L \cdot L^T$. Thus, L is a linear transformation [12] whose effect is shown in figure 2.

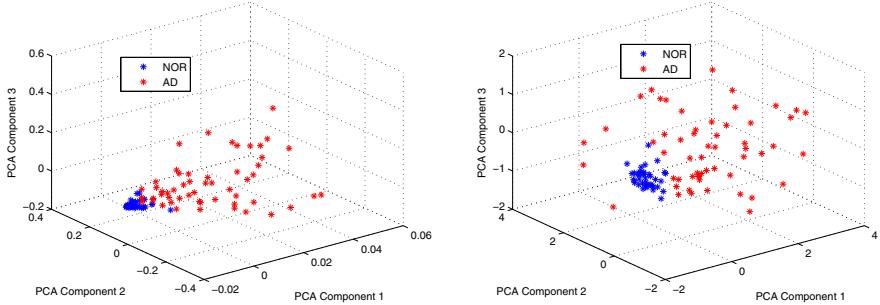


Fig. 2. coefficients after projecting the data onto the first three PCs a) using only kernel PCA (left) and b) using LMNN linear space transformation over PCs (right)

5 Results

Several experiments were conducted to evaluate the combination of VAF, NMSE and PCA feature extraction aiming to posterior Distance Metric Learning for brain image classification. The performance of the Distance Metric Learning-based classifier was evaluated in depth as a tool for the early detection of the AD in terms of Accuracy(Acc), Sensitivity (Sen) and Specificity (Spe), which are estimated by the leave-one-out cross-validation. The experiments considered an increasing number of kernel-PCA components for the comparision of the Euclidean, Mahalanobis and Energy-based distance metric learning respectively. Sensitivity and Specificity are defined as:

$$\text{Sensitivity} = \frac{TP}{TP+FN}; \quad \text{Specificity} = \frac{TN}{TN+FP}$$

respectively, where TP is the number of true positives: number of AD patients correctly classified; TN is the number of true negatives: number of controls

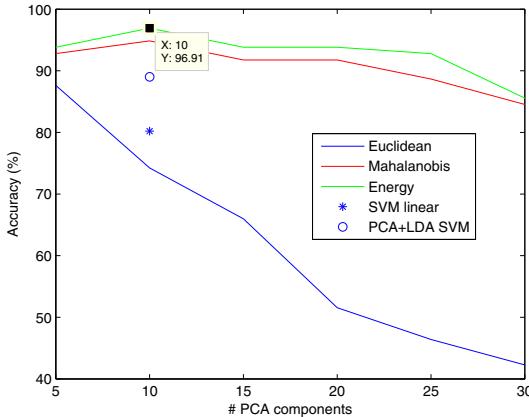


Fig. 3. Accuracy for an increasing number of PCs and Distance Metric Learning classification based on Euclidean, Mahalanobis and Energy techniques. Comparison to other reported methods.

Table 1. Sensitivity, Specificity, Accuracy using Euclidean, Mahalanobis and Energy distances using different kernel PCA types

Kernel Type	Euclidean	Mahalanobis	Energy
linear	69.072	87.62	92.78
sigmoid	67.01	85.56	88.65
poly	92.78	93.81	74.22
gaussian	74.23	94.85	96.91

correctly classified; FP is the number of false positives: number of controls classified as AD patient; FN is the number of false negatives: number of AD patients classified as control. Figure 2(b) is aimed to demonstrate that Distance Metric Learning is an effective way to obtain a large margin between AD and NORMAL subjects due to the product of the linear LMNN transformation (L) with the KPCA features, which clearly outperforms the separation of classes with regard to the figure 2(a), where only KPCA features were used. In Figure 3, it is shown the accuracy of Euclidean, Mahalanobis and Energy distance metric methods and other reported methods such as SVM linear or Linear PCA in combination with LDA and evaluated with SVM to compare with the technique of this work. It is observed that the best results were obtained for Energy with 10 PCA components. That is the reason why, in table II the results are summarized for 10 PCA components. The strategy to evaluate the method consists of differing, in terms of Acc, Sen, Spe the method which best discriminates Normal and AD subjects in terms of the distance metric learning, that is, the highest margin which separates the two considered classes. Firstly, NMSE features in blocks of $5 \times 5 \times 5$ were extracted. Secondly, the $r_1=100$ most discriminative

Table 2. Sensitivity, Specificity, Accuracy using Euclidean (Eu), Mahalanobis (Maha) and Energy (En) distances respectively and an increasing number of Gaussian Kernel PCA components

N	PCA	Sensitivity (%)			Specificity (%)			Accuracy (%)		
		Eu.	Maha.	En.	Eu.	Maha.	En.	Eu.	Maha.	En.
3		97.83	96.23	96.3	78.43	88.66	90.7	88.66	91.75	
4		97.67	97.96	98.04	74.07	83.33	86.96	91.75	93.81	92.78
5		93.73	98	98.11	75.47	85.11	90.91	87.63	92.78	93.81
6		97.5	100	100	70.18	91.11	92.72	84.54	90.72	92.78
7		97.44	100	98.18	68.97	89.13	95.24	85.57	91.75	94.84
8		96.97	100	100	62.5	89.13	93.18	81.44	95.88	96.91
9		96.77	100	100	60.61	85.42	87.23	80.41	94.85	96.91
10		100	100	100	56.94	87.23	95.35	74.23	94.85	96.91
11		100	100	96.3	56.16	87.23	90.7	72.16	92.78	93.81

NMSE ROIs were found by means of a t-test feature selection in which features highly correlated with already selected features are less likely to be included in the output list.

Thirdly, up to 11 PCA components were evaluated using gaussian Kernel (table 2) where the best results of Acc, Sen and Spe were reached for $r_2=10$ pca components with energy-based distance metric learning. A higher number of PCA components may lead to an increase in the computational cost of the CAD system, and a penalty in the Acc rate because no-discriminant features are being used too (see figure 3). Observing the table 2, we can conclude that energy-based distance metric learning method is the best, with results of Acc 96.91%, Sen 100% and Spe 95.35%, respectively, when the number of components for the gaussian Kernel PCA is set to 10. We have chosen the Gaussian Kernel in table 2 since as it is observed in table II, the best results of accuracy are reached for this type of kernel.

6 Conclusions

Kernel Distance Metric Learning Methods were investigated for SPECT images classification for the early AD's diagnosis, obtaining results of Acc of 96.91%, Sen of 100% and Spe of 95.35% for an energy-based method, outperforming other recently reported methods including Support Vector Machine (SVM) or PCA in conjunction with LDA evaluated with SVM [10]. As feature extraction is concerned, the combination of NMSE t-test selection and then gaussian kernel PCA helped reach the best results of classification.

References

1. Petrella, J.R., Coleman, R.E., Doraiswamy, P.M.: Neuroimaging and early diagnosis of alzheimer's disease: A look to the future. *Radiology* 226, 315–336 (2003)
2. English, R.J., Childs, J. (eds.): *SPECT: Single-Photon Emission Computed Tomography: A Primer*. Society of Nuclear Medicine (1996)
3. Fung, G., Stoeckel, J.: SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information. *Knowledge and Information Systems* 11, 243–258 (2007)
4. Derrac, J., García, S., Herrera, F.: A first study on the use of coevolutionary algorithms for instance and feature selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) *HAIS 2009*. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
5. Xiang, S., Nie, F., Zhang, C.: Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition* 41, 3600–3612 (2008)
6. Pérez, P., Chardin, A., Laferte, J.: Noniterative manipulation of discrete energy-based models for image analysis. *Pattern Recognition* 33, 573–586 (2000)
7. Saxena, P., Pavel, D., Quintana, J., Horwitz, B.: An automatic threshold-based scaling method for enhancing the usefulness of tc-hmpao spect in the diagnosis of alzheimer's disease. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) *MICCAI 1998*. LNCS, vol. 1496, pp. 623–630. Springer, Heidelberg (1998)
8. Chaves, R., Ramírez, J., Górriz, J., López, M., Salas-Gonzalez, D., Alvarez, I., Segovia, F.: Svm-based computer-aided diagnosis of the alzheimer's disease using t-test nmse feature selection with feature correlation weighting. *Neuroscience Letters* 461, 293–297 (2009)
9. Andersen, A., Gash, D.M., Avison, M.J.: Principal component analysis of the dynamic response measured by fmri: a generalized linear systems framework. *Journal of Magnetic Resonance Imaging* 17, 795–815 (1999)
10. López, M., Ramírez, J., Górriz, J.M., Salas-Gonzalez, D., Alvarez, I., Segovia, F., Puntonet, C.G.: Automatic tool for the Alzheimer's disease diagnosis using PCA and bayesian classification rules. *IET Electronics Letters* 45(8), 389–391 (2009)
11. Chatpatanasiri, R., Korsrilaibutr, T., Tangchanachaianan, P., Kijssirikul, B.: A new kernelization framework for mahalanobis distance learning algorithms. *Neurocomputing* 73, 1570–1579 (2010)
12. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10, 207–244 (2009)
13. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. *Advances in Neural Information Processing Systems* 17, 513–520 (2005)
14. Chopra, S., Hadsell, R., LeCun., Y.: Learning a similiarty metric discriminatively, with application to face verification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, CA (2005), doi:10.1109/CVPR.2005.202
15. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 6, 1453–1484 (2005)

Risk Estimation for Hierarchical Classifier

I.T. Podolak and A. Roman

Institute of Computer Science, Jagiellonian University
`{podolak,roman}@ii.uj.edu.pl`

Abstract. We describe the Hierarchical Classifier (HC), which is a hybrid architecture [1] built with the help of supervised training and unsupervised problem clustering. We prove a theorem giving the estimation \hat{R} of HC risk. The proof works because of an improved way of computing cluster weights, introduced in this paper. Experiments show that \hat{R} is correlated with HC real error. This allows us to use \hat{R} as the approximation of HC risk without evaluating HC subclusters. We also show how \hat{R} can be used in efficient clustering algorithms by comparing HC architectures with different methods of clustering.

1 Introduction

Given a finite data set $D = \{(x_i, c_i)\}_{i=1}^N$, with N example attribute vectors x_i , each labeled with a true class c_i , a classification algorithm is expected to find a hypothesis that is as consistent with the labelings, as possible. Several approaches exist: single models, like feed-forward neural networks, decision trees, clustering algorithms, SVMs [2,3,4] and machine committees, where several simple models are combined to obtain the final classification by way of, e.g., averaging [4,5,6].

In case of more complicated problems, when using single model algorithms we frequently need to search long for the right architecture and parameters. This may often result in solutions more complicated than the problem at hand. One of the solutions to overcome such problems is the *boosting* approach, where the machine committees are built in sequence. Each subsequent classifier actually profits from the inaccuracy of models built so far to drive the error down. Thanks to this, it is possible to use *simple* individual models, e.g. neural networks with small number of weights, and still attain a high accuracy.

The Hierarchical Classifier (HC) described in Sect. 2 is also a boosting algorithm, with structure different from the other well-known boosting algorithms like AdaBoost. The HC structure comes from the observation that simple models, when dealing with problems with several output classes that it cannot solve with low error, tend to group together classes with similar input vectors. Hence, it is possible to split the data set into several subproblems with limited number of output classes, each to be solved separately. This approach forms a tree of classifiers, with outputs combined in the final classification rule. Therefore, this is a boosting approach different from the one proposed in AdaBoost and similar algorithms.

The idea of boosting started in some papers of Valiant and Kearns [7], Schapire and Freund [8, 9]. Kearns and Valiant have introduced the notion of *Probably Approximately Correct* (PAC) learning, and the notion of *strong* and *weak learnability*. Later, this notion was extended by Schapire, who proved the equality of weak and strong learning. The HC model also uses weak classifiers in its nodes. Detailed discussion on this issue is given in [10], where the new notion of weakness for multiclass classification was introduced and analyzed.

The HC model was introduced earlier in [11, 12]. In this paper we introduce a new method for computing the risk estimation $\hat{R}(Cl^{HC})$ of HC. We check how this measure is correlated with some other parameters, like HC real error defined in terms of different loss functions. We also introduce some new clustering methods which use $\hat{R}(Cl^{HC})$, and compare them.

In Sect. 2 we shortly introduce the HC model and a new way for computing cluster weights. This method will play an essential role in the estimation of the HC risk, described in Sect. 3. In Sect. 4 we describe two experiments involving the $\hat{R}(Cl^{HC})$ measure. Finally, discussion follows with conclusions.

2 Hierarchical Classifier

First, we define the notion of a classifier. It will be used throughout the paper.

Definition 1. Let $D = \{(x_i, c_i), i = 1, \dots, N\}$, $x_i \in X$ be a finite set of training examples, such that $c_i \in \mathcal{C} = \{C_1, \dots, C_K\}$ is the true class of x_i . A classifier is a function

$$Cl : X \ni x \longrightarrow [0, 1]^K .$$

Slightly abusing the notation, sometimes we will refer to the class C_i as i for simplicity. A classifier Cl returns a K -element class probability vector $[Cl_1(x), \dots, Cl_K(x)]$. Then, we may use a selection function $S(x) = \arg \max_k Cl_k(x)$, treating $C_{S(x)}$ as the class predicted by Cl for an attribute vector x . The elements of $Cl(x)$ vector are used to compute cluster weights.

Hierarchical Classifier formal definition. Let $D \subset X \times \mathcal{C}$ be a set of N training examples for a K -class classification problem, $\mathcal{C} = \{C_i, i = 1, \dots, K\}$. Now we give the formal definition of HC – the notions and concepts used in it are explained in detail below.

Definition 2. A Hierarchical Classifier is a tuple $HC = (V, V^0, child)$, where $V = \{V^i = (Cl^i, F^i), i \in I\}$, $child : V \rightarrow 2^V$, $V^0 \in V$, $Cl^i : X \rightarrow [0, 1]^K$, $F^i = \begin{cases} K \times L^i \text{ matrix } (L^i = |child(V^i)|) & \text{if } |child(V^i)| > 0 \\ \emptyset & \text{otherwise} \end{cases}$, such that I is some finite set of indices, $sup(Cl^0(X)) = \mathcal{C}$ and

$$\forall i \in I : |child(V^i)| > 0 \quad \bigcup_{l: V^l \in child(V^i)} sup(Cl^l(X)) = sup(Cl^i(X)),$$

where $sup(Cl^i(X))$ denotes the support of Cl^i , that is, the set of all classes that can be recognized by Cl^i with positive probability.

HC structure. Hierarchical classifier is a tree with V as a set of nodes, $|V| = |I|$. Node V^0 is the *root* of this tree. For a given $V^i \in V$ the *child* function returns the set of child nodes of V^i . All V^i nodes with $|\text{child}(V^i)| > 0$ are called *internal*, and those for which $|\text{child}(V^i)| = 0$ are *leaf* nodes. Each internal node V^i consists of a classifier Cl^i and a *clustering matrix* F^i (defined below). Each leaf is composed of a classifier only, without a clustering matrix. Clustering matrix F^i represents the sets (clusters) of recognizable classes, i.e. subproblems, for classifiers being V^i 's children. A very important element of the proposed model is the fact that the clusters may *overlap* — we do not require that the family $\{Q_l^i\}_{l=1}^{L^i}$ forms the partition of $\text{sup}(Cl^i(X))$. This feature helps in increasing the accuracy of the whole model. The main idea for introducing the overlapping clusters is to allow the classifier to correct its classification on the lower levels of the tree in case the parent classifier made the wrong decision. The detailed discussion on that issue is given in [10].

Weights. For each $V^i \in V$ we define a *weight* function $w^i : X \rightarrow [0, 1]^{L^i}$, where $w^i(x) = [w_1^i(x), \dots, w_{L^i}^i(x)]$. The weights are used in combining the answers of all classifiers into one, final answer. The weights may be defined in many different ways [13] (below we introduce a new method). We only require that $w^i(x)$ is normalized for each x , that is $\sum_{l=1}^{L^i} w_l^i(x) = 1$.

Clustering matrix F . Each internal HC node Cl^i divides the set of recognizable classes into overlapping clusters, defining the subproblems for Cl^i child nodes. The clustering is defined with a $K \times L^i$ binary matrix $F^i = (f_{kl}^i)$, $k = 1, \dots, K$, $l = 1, \dots, L^i$, where $f_{kl}^i = 1$ if $C_k \in Q_l^i$, and $f_{kl}^i = 0$ otherwise. We say that each F^i defines a family of *clusters* $Q_l^i \subset \{C_1, \dots, C_K\}$, $l = 1, \dots, L^i$, such that $C_j \in Q_l^i \Leftrightarrow f_{jl}^i = 1$. Additionally we assume that F matrix must satisfy some properties: (P1) each cluster should have > 1 and $< K$ classes (otherwise, no improvement in accuracy will be obtained); (P2) each class must be included in some cluster (otherwise, some classes would not be recognized); (P3) at least two clusters must overlap (for HC to have a chance to correct a misclassification made in parent classifier); (P4) no cluster is a subset of the other (to increase the HC diversity [14]).

Evaluation of HC. For a given attribute vector $x \in X$ first the root classifier $Cl^0(x)$ is evaluated. Basing on Cl^0 answer, the weights w_l^0 are computed, $l = 1, \dots, L^0$. The final HC answer is $Cl^{HC}(x) = \sum_{l: V^l \in \text{child}(V^0)} w_l^0(x) Cl^l(x)$. All Cl^l subclassifiers are evaluated recursively in the same manner. For the leaf classifier Cl^t the answer is just $Cl^t(x)$.

Misclassification matrix. Misclassification matrix for a classifier Cl is a $K \times K$ matrix $M = (m_{ij})$ describing the way that classifier misclassifies the examples. It is built during the training process and used only to construct a clustering matrix F , but not during evaluation (therefore, M formally is not a part of HC structure). M can be used as an estimator of the conditional probability that Cl will classify an example as from class C_j under the condition that the true class is C_i : $m_{ij} = E[P(pr(x) = j | tr(x) = i)]$, where tr (resp. pr) stands for *true class*

(resp. *predicted class*). In our considerations we define m_{ij} in the following way: $m_{ij} = E[Cl_j(x)|x \in C_i]$. Such M matrix is computed from the training data T : when the classifier training is done, all examples from T are given again as the input to Cl and then we put $m_{ij} = \hat{m}_{ij}$, where $\hat{m}_{ij} = \frac{1}{|X_i|} \sum_{x \in X: tr(x)=i} Cl_j(x)$ is a sample estimator of m_{ij} .

Computing cluster weights. The weight function returns a cluster probability vector w . The l -th component of w^i can be viewed as an estimation of the probability that a given example class belongs to Q_l^i . The w_l^i values are used in the evaluation of HC for a given x . In [II] the weights were defined in the following way:

$$w_l^i(x) = \frac{\sum_{k=1}^K f_{kl}^i Cl_k^i(x)}{\sum_{l'=1}^{L^i} \sum_{k=1}^K f_{k'l'}^i Cl_k^i(x)}, \quad l = 1, \dots, L^i. \quad (1)$$

We will modify this formula using Bayes rule. This simple modification is important for two reasons. First, it takes into account that the root classifier misclassifies the answers, and second, that it allows us to compute the exact risk value for a given HC. If the old method of computing weights is used, in the proof we would obtain the component like $E[\frac{X}{Y}]$, where X and Y are dependent random variables and, what is more, we do not know their distributions. It is impossible to compute the exact value of this expression. Using the modified version of (II) we are able to provide the exact risk value.

Suppose that the root classifier classified an example x as being from class C_j . Using Bayes rule we have

$$\begin{aligned} P(tr(x) = t | pr(x) = j) \\ = \frac{P(pr(x) = j | tr(x) = t) P(tr(x) = t)}{\sum_{k=1}^K P(pr(x) = j | tr(x) = k) P(tr(x) = k)} = \frac{m_{tj} p_t}{\sum_{k=1}^K m_{kj} p_k}, \end{aligned} \quad (2)$$

where $P = [p_1, \dots, p_K]$ is the vector of prior probabilities for all classes. The p_t values are computed from the training dataset: $p_t = \frac{|\{(x, C_t) \in D\}|}{|D|}$. Let $\Gamma^i = \Gamma^i(M^i, P) = (\gamma_{jk}^i)$, $j, k = 1, \dots, K$ and $\gamma_{jk}^i = \frac{m_{jk}^i p_j}{\sum_{s=1}^K m_{sj}^i p_s}$. We modify now (II) in the following way:

$$w_l^i(x) = \frac{\sum_{k=1}^K f_{kl}^i \gamma_{kj}^i}{\sum_{l'=1}^{L^i} \sum_{k'=1}^K f_{k'l'}^i \gamma_{k'j}^i}, \quad l = 1, \dots, L^i, \quad (3)$$

where $j = \arg \max_v \{Cl_v^i(x)\}$ is the index of a class predicted by Cl^i for x . Let $pr(x)$ be the class predicted by Cl^i . In (II) all $Cl^i(x)$ vector values were used to compute $w_l^i(x)$. Equation (3) is based on the Bayes rule: it takes into account only the index j of the largest value from $Cl^i(x)$ vector. This is interpreted as the event that $pr(x) = j$. Using (2) we can compute $P(tr(x) = k | pr(x) = j)$ and replace Cl_k^i with γ_{kj}^i in (II). The $w_l^i(x)$ value gives now the measure of the a posteriori probability that the true class is in Q_l^i under the condition that Cl^i classified x as from C_j .

Ways of obtaining clustering matrix. In the following experiments we use 4 different clustering algorithms. Two of them ($Bayes_G$ and $Bayes_R$) are deterministic and are based on the Bayes rule. Two others (GA_1 , GA_2) use genetic algorithms with different fitness functions. Bayes algorithms start with identity $K \times K$ clustering matrix $F = I$. Then, they successively add to F elements $f_{ji} = 1$ corresponding to the largest γ_{ij} values. The addition is performed only if: a) some fitness function $fit = fit(M, F)$ will grow after potential addition, and b) cluster sizes do not exceed some predefined threshold (for computational reason it is taken to be $\lceil \frac{K}{2} \rceil$). For $Bayes_G$ we have [15]

$$fit(M, F) = \text{tr}(\text{diag}(\sum_{l=1}^L \sum_{k=1}^K m_{1k} f_{kl}, \dots, \sum_{l=1}^L \sum_{k=1}^K m_{Kk} f_{kl}) M F F^T). \quad (4)$$

For $Bayes_R$ we use $fit(M, F) = \hat{R}(Cl^{HC})$, the HC risk estimation defined in the next section. Genetic algorithms operate on populations of F matrices. They use two kinds of mutation: first one randomly changes single f_{ij} values and the second one randomly adds or removes clusters (F columns). They also use a crossover operator which changes some subsets of columns between two F matrices. The genetic algorithms use different fitness functions: $f_1 = 3f_R + f_F$ and $f_2 = 3f_R + 3f_D + f_F$, where: $f_R = \hat{R}(Cl^{HC})$ is defined in the next section; f_F evaluates the F matrix correctness in terms of properties (P1)-(P4) defined above; f_D measures the diversity of clusters [14].

3 Estimation of HC Risk

In this section we provide the formula which allows us to estimate the HC risk $R(Cl^{HC})$, given the misclassification matrix and clustering matrix for the root classifier. We denote this sample estimator as $\hat{R}(Cl^{HC})$. The proof is valid for HC with the modified way of computing weights, given by (3). Recall that the risk is the expected value of the loss function: $R(Cl^{HC}) = E[l(x, tr(x), Cl^{HC}(x))]$. We use the pseudo-loss function introduced in [9]. This function is useful in K -class classification problem, where $K > 2$. Pseudo-loss for a classification of x by Cl in a K -class classification problem is defined as $\ell(x, tr(x), Cl(x)) = \frac{1}{2} \left(1 - Cl_{tr(x)}(x) + \frac{1}{K-1} \sum_{k \neq tr(x)} Cl_k(x) \right)$, where $tr(x)$ is the true class of x . We will provide the formula assuming that HC is a 2-level classifier (denoted by Cl^{HC}) with root (denoted by Cl) and L subclassifiers Cl^1, \dots, Cl^L , but obviously the formula can be used recursively for all classifiers, therefore can be used for HC with any number of levels. Let M and F be the misclassification and clustering matrices for Cl , P be the prior probabilities vector and $\Gamma = \Gamma(M, P)$. To compute $R(Cl^{HC})$ it is enough to compute

$$E[Cl_{tr(x)}^{HC}(x)] - \frac{1}{K-1} \sum_{k \neq tr(x)} E[Cl_k^{HC}(x)]. \quad (5)$$

For the sake of simplicity, let us define some additional matrices. Put $\Delta = \text{diag}(1/s_1, \dots, 1/s_K) F^T \Gamma$, where $s_j = \sum_{l'=1}^L \sum_{k'=1}^K f_{k'l'} \gamma_{k'j}$. Notice that Δ is

just $F^T \Gamma$ normalized by columns. Let us also define θ as $K \times L$ matrix with $\theta_{il} = m_{ii}^l$, where m_{ii}^l , $i = 1, \dots, K$ are the trace elements of the misclassification matrix M^l for Cl^l . These trace elements can be estimated from the simulation of Cl^l by Cl (see [15]): $m_{ii}^l = m_{ii} + p_i \sum_{j=1}^K (1 - f_{ji}) \frac{m_{ij}^2}{\sum_{t=1}^K f_{ti} m_{tj} p_t}$. Moreover, let us define: $\varphi = \theta \Delta M^T$, $\mu = (1 - \theta) \Delta M^T$, $\hat{P} = \text{diag}(P)$, $\hat{\sigma} = \text{diag}(1/|Q_1|, \dots, 1/|Q_L|)$, $\chi = M \Delta^T \hat{\sigma}^T F^T$, $\hat{\xi} = \text{diag}(|Q_1| - 1, \dots, |Q_L| - 1)$, $\nu = \hat{\xi} F \hat{\sigma} \Delta M^T$. Now let us compute the first component in sum (5). The notion $pr(x)$ refers to the class predicted by Cl .

$$\begin{aligned}
E[Cl_{tr(x)}^{HC}(x)] &= \sum_{i=1}^K \sum_{j=1}^K E[Cl_i^{HC}(x) | tr(x) = i, pr(x) = j] m_{ij} p_i \\
&= \sum_{i=1}^K \sum_{j=1}^K \sum_{l=1}^L E[w_l(x) Cl_i^l(x) | tr(x) = i, pr(x) = j] m_{ij} p_i \\
&= \sum_{i=1}^K \sum_{j=1}^K \sum_{l=1}^L E[w_l(x) | tr(x) = i, pr(x) = j] E[Cl_i^l(x) | tr(x) = i, pr(x) = j] m_{ij} p_i \\
&= \sum_{i=1}^K \sum_{j=1}^K \sum_{l=1}^L E[w_l(x) | pr(x) = j] E[Cl_i^l(x) | tr(x) = i] m_{ij} p_i \\
&= \sum_{i=1}^K \sum_{l=1}^L \sum_{j=1}^K \Delta_{lj} m_{ii}^l m_{ij} p_i = \sum_{i=1}^K p_i \sum_{l=1}^L \sum_{j=1}^K \Delta_{lj} m_{ji}^T = P \cdot \text{diag } \varphi
\end{aligned}$$

The second term of (5) may be computed in a similar way. However, in this case we must compute values $E[Cl_k^l(x) | tr(x) = i]$ when $C_i \notin Q_l$ and $C_k \in Q_l$. This is the case in which x cannot be properly recognized by Cl^l . In this situation we assume that the expected value on each position of the $Cl^l(x)$ vector is $\frac{1}{|Q_l|}$.

$$\begin{aligned}
E[\sum_{k \neq tr(x)} Cl_k^{HC}(x)] &= \sum_{i=1}^K \sum_{j=1}^K \sum_{k \neq i} E[Cl_k^{HC}(x) | tr(x) = i, pr(x) = j] \\
&= \sum_{i=1}^K \sum_{j=1}^K p_i m_{ij} \sum_{k \neq i} \sum_{l=1}^L \Delta_{lj} E[Cl_k^l(x) | tr(x) = i] \\
&= \sum_{i=1}^K \sum_{j=1}^K p_i m_{ij} \sum_{k \neq i} \sum_{l=1}^L \Delta_{lj} \left(f_{il} f_{kl} m_{ik}^l + (1 - f_{il}) f_{kl} \frac{1}{|Q_l|} \right) \\
&= P \cdot \text{diag } \mu + \mathbf{1} \hat{P} \chi^T \mathbf{1}^T - \text{tr}(\hat{P} \chi^T) - P \cdot \text{diag } \nu.
\end{aligned}$$

Theorem 1

$$\hat{R}(Cl^{HC}) = \frac{1}{2} \left(1 - P \cdot \text{diag } \varphi + \frac{P \cdot \text{diag}(\mu - \nu) + \mathbf{1} \hat{P} \chi^T \mathbf{1}^T - \text{tr}(\hat{P} \chi^T)}{K - 1} \right).$$

4 Experiments and Conclusions

Experiments were performed for 5 well-known benchmark datasets [16]. For each dataset 100 bootstrap sets B_1, \dots, B_{100} were generated. For each B_i a single root classifier Cl^0 (ANN with 4 hidden neurons) was trained. For each Cl^0 4 different clustering methods and 2 different evaluations (using (1) and (3) marked in Tab. 1 as 'std' and ' Γ ') were used, hence 8 different HCs were obtained. Table 1 shows Spearman correlations between \hat{R} and other measures for HCs with all clustering types and evaluations. The other measures are: trG_{HC} given by (4) and $Err^{(.632)}$ bootstrap estimate (see [17]) of $1 - Cl_{tr(x)}^{HC}$ prediction error. Table 2 shows $Err^{(.632)}$ errors for all clustering methods and evaluations.

Table 1. Spearman ρ correlations

Correlation	Problem				
	vowel	tumor	arrythmia	audiology	zoo
$\hat{R} \sim Err^{(.632)}$	0.430	0.403	0.697	0.314	0.519
$trG_{HC} \sim Err^{(.632)}$	-0.467	-0.241	0.022	-0.478	-0.558
$\hat{R} \sim trG_{HC}$	-0.661	-0.701	-0.196	-0.567	-0.696

Table 2. Comparison of different clustering algorithms in terms of $Err^{(.632)}$ for HCs with different loss and clustering type

Loss type →	$Err^{(.632)}$ error							
	for $1 - Cl_{tr(x)}^{HC}$ loss				for binary $\ell_{0/1}$ loss			
Problem	$Bayes_G$	$Bayes_R$	GA_1	GA_2	$Bayes_G$	$Bayes_R$	GA_1	GA_2
Vowel Γ	0.366	0.390	0.345	0.360	0.246	0.240	0.226	0.267
Vowel std	0.331	0.353	0.338	0.339	0.240	0.232	0.222	0.264
Primary-tumor Γ	0.610	0.609	0.618	0.617	0.471	0.459	0.473	0.524
Primary-tumor std	0.599	0.601	0.616	0.609	0.467	0.455	0.471	0.517
Arrythmia Γ	0.619	0.614	0.547	0.594	0.461	0.460	0.466	0.459
Arrythmia std	0.619	0.613	0.547	0.596	0.460	0.459	0.465	0.458
Audiology Γ	0.439	0.434	0.480	0.496	0.277	0.253	0.294	0.362
Audiology std	0.431	0.425	0.477	0.493	0.274	0.246	0.293	0.363
Zoo Γ	0.103	0.114	0.096	0.111	0.068	0.072	0.067	0.085
Zoo std	0.102	0.113	0.099	0.110	0.068	0.070	0.066	0.084

In the experiments described above the most interesting to us were the relations between \hat{R} and other measures, and the effect of applying the \hat{R} measure in different clustering algorithms. Therefore, the minimization of the error value was not the most important aim and is not relevant here.

Conclusions. The \hat{R} value is positively correlated with true HC error, see Tab. 1 (for arrythmia it is even a very high correlation, 0.697). It seems that \hat{R} is also a more stable measure than trG_{HC} , which for some problems is positively and for others — negatively correlated with true HC error. This makes \hat{R} a good HC

real error estimation. The best algorithm using \hat{R} is better than $Bayes_G$ in 18 out of 20 cases. In 13 cases $Bayes_R$ is better than $Bayes_G$. Algorithm $Bayes_G$, which does not use \hat{R} , overcomes all 3 algorithms using \hat{R} estimation only in vowel 'std' and tumor 'std' problems, but in both cases the predominance over the second best algorithm is insignificant.

Acknowledgments. This research was partially funded by the Polish National Science Centre Grant No. N519 654840.

References

1. Corchado, E., Abraham, A., Carvalho, A.C.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)
2. Haykin, S.: Neural networks: a comprehensive foundation. Prentice-Hall, Englewood Cliffs (2009)
3. Christiani, N., Shawe-Taylor, J.: Support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
4. Bishop, C.: Pattern recognition and machine learning. Springer, Heidelberg (2006)
5. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, Heidelberg (2001)
6. Tresp, V.: Committee Machines. In: *Handbook for Neural network Signal Processing*. CRC Press, Boca Raton (2001)
7. Kearns, M., Valiant, L.: Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM* 41(1), 67–95 (1994)
8. Shapire, R.E.: The strength of weak learnability. *Machine Learning* 5, 197–227 (1990)
9. Freund, Y., Shapire, R.E.: A decision theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences* 55, 119–139 (1997)
10. Podolak, I.T., Roman, A.: Theoretical foundations and practical results for the hierarchical classifier. Submitted to *Computational Intelligence*
11. Podolak, I.T.: Hierarchical Classifier with Overlapping Class Groups. *Expert Systems with Applications* 34(1), 673–682 (2008)
12. Podolak, I.T.: Hierarchical rules for a hierarchical classifier. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 749–757. Springer, Heidelberg (2007)
13. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants – Evolutionary and Neural Methods of Training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS (LNAI), vol. 6076, pp. 590–597. Springer, Heidelberg (2010)
14. Shipp, C.A., Kuncheva, L.I.: Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion* 3, 135–148 (2002)
15. Podolak, I., Roman, A.: Fusion of supervised and unsupervised training methods for a multi-class classification problem. Accepted for publication in *Pattern Analysis and Applications*
16. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA,
<http://archive.ics.uci.edu/ml>
17. Efron, B.: Estimating the error rate of a prediction rule: some improvements on cross-validation. *Journal of the American Statistical Association* 78, 316–331 (1983)

Combining Meta-learning and Active Selection of Datasetoids for Algorithm Selection

Ricardo B.C. Prudêncio¹, Carlos Soares², and Teresa B. Ludermir¹

¹ Center of Informatics, Federal University of Pernambuco,
Cidade Universitária - CEP 50732-970 - Recife (PE) - Brazil
`{rbcp,tbl}@cin.ufpe.br`

² LIAAD-INESC Porto L.A., Faculdade de Economia, Universidade do Porto
Rua de Ceuta 118-6, 4050-190, Porto, Portugal
`csoares@fep.up.pt`

Abstract. Several meta-learning approaches have been developed for the problem of algorithm selection. In this context, it is of central importance to collect a sufficient number of datasets to be used as meta-examples in order to provide reliable results. Recently, some proposals to generate datasets have addressed this issue with successful results. These proposals include datasetoids, which is a simple manipulation method to obtain new datasets from existing ones. However, the increase in the number of datasets raises another issue: in order to generate meta-examples for training, it is necessary to estimate the performance of the algorithms on the datasets. This typically requires running all candidate algorithms on all datasets, which is computationally very expensive. One approach to address this problem is the use of active learning, termed active meta-learning. In this paper we investigate the combined use of active meta-learning and datasetoids. Our results show that it is possible to significantly reduce the computational cost of generating meta-examples not only without loss of meta-learning accuracy but with potential gains.

Keywords: Meta-learning, Active learning.

1 Introduction

A large number of learning algorithms are available for data analysis nowadays. For instance, decision trees, neural networks, support vector machines, among others, can be used in classification problems. After narrowing down the list of candidate algorithms taking into account problem-specific constraints (e.g., interpretability of the model), the goal of data analysts is to select the algorithm with higher chances to obtain the best performance on the problem at hand. The algorithm selection problem is addressed by *meta-learning* as a supervised learning task [3]. A learning algorithm is used to model the relation between the characteristics of learning problems (e.g., number of examples, proportion of symbolic attributes) and the relative performance of a set of algorithms.

An important issue in the development of meta-learning systems for algorithm recommendation is the computational cost of generating the meta-data [3]. This

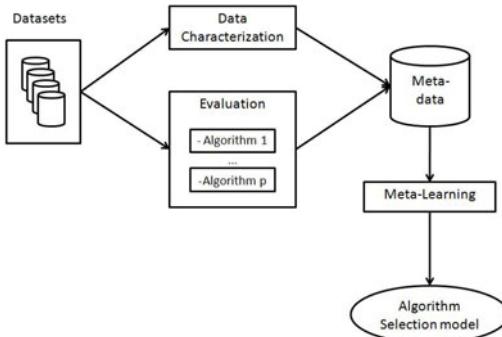


Fig. 1. The meta-learning process for algorithm selection (adapted from [3])

implies running the candidate algorithms on all the training datasets, which can be quite costly. In this paper, we address this problem using a hybrid approach, that combines two machine learning approaches [6]: an active learning approach with a dataset generation method, datasetoids [14], to guide the meta-data collection process. This combination is tested on an algorithm selection task. The contribution of this work is to show that these two methods simultaneously and successfully address two of the major issues in meta-learning: obtaining sufficient datasets for reliable meta-learning and reducing the computational cost of collecting meta-data.

We start by describing background information on meta-learning, including datasetoids, (Section 2) and active learning (Section 3). Next, we present the experimental setup used to evaluate the approach empirically (Section 4). We close the paper with conclusions and some ideas for future work (Section 5).

2 Meta-learning for Algorithm Selection

The meta-learning approach to algorithm selection is summarized in Figure 1. A database is created with meta-data descriptions of a set of datasets. These meta-data contain estimates of the performance of a set of candidate algorithms on those datasets as well as meta-features describing their characteristics (e.g., number of examples in the dataset, entropy of the class attribute, mean correlation between the numerical attributes). A machine learning algorithm (the so-called *meta-learner*) is applied to this database to induce a model that relates the values of the meta-features to the performance of the candidate algorithms (e.g., the best algorithm on the datasets). For more information on meta-learning for algorithm recommendation, the reader is referred to [3] and references therein.

An important issue in meta-learning is the availability of a sufficient number of datasets to enable reliable (meta-)induction. The UCI Repository [1] is the most common source of examples for meta-learning, however it contains slightly over 100 classification datasets. Given that each dataset represents one meta-example, most meta-learning research is based on approximately 100 meta-examples. This

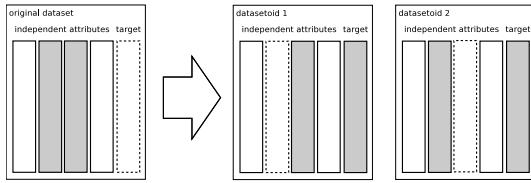


Fig. 2. Illustration of the generation of two classification datasetoids from a dataset with two symbolic attributes (reproduced from [14])

is a small number to obtain reliable models, particularly in such a complex application such as meta-learning. This problem is receiving an increasing amount of attention recently. Two common approaches are the generation of synthetic datasets and the manipulation of existing ones [7][9].

In this work we use *datasetoids*, a very simple data manipulation approach to generate new datasets which was recently proposed [14]. A datasetoid is generated from a given dataset by switching the target attribute with an independent attribute (Figure 2). Thus, the target attribute of the dataset becomes an independent attribute in the datasetoid and the independent attribute selected in the dataset becomes the target attribute in the datasetoid. To generate datasetoids for classification, the process is repeated for every symbolic attribute of the dataset, thus creating as many datasetoids as there are symbolic attributes in the dataset. Experiments on the problem of deciding whether to prune a decision tree using meta-data containing datasetoids obtained significant improvements when compared to meta-data that only contained datasets [14].

3 Active Learning and Meta-learning

Active learning is a paradigm of Machine Learning in which the learning algorithm has some control over the examples on which it trains [5]. It has been used in many tasks to reduce the number of training examples, while maintaining (or even improving) the learning performance [8][11][12][13]. Active learning is ideal for domains in which the acquisition of labeled examples is a costly process. The cost of acquiring labels for meta-learning is computationally expensive, as it is necessary to execute the candidate algorithms on the datasets used for training. This makes meta-learning a good candidate problem for active learning.

In [10], active learning is proposed to improve the generation of meta-examples. This proposal, termed as *Active Meta-learning*, is illustrated in Figure 3. The method starts with a small set of labeled examples and a large set of unlabeled ones. An active learning module receives these two sets as input and selects, from the latter, the next example to be labeled. The selection of unlabeled meta-examples is performed based on a pre-defined criterion which takes into account the meta-features of the problems and the current set of labeled examples. Labeling is done by evaluating the candidate algorithms on the selected problem and the best algorithm becomes the label of the corresponding meta-example. The

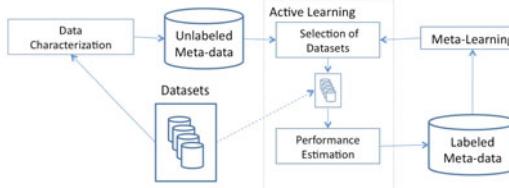


Fig. 3. Active meta-learning process

process iterates until some stopping condition. The algorithm selection model is then obtained by meta-learning on the labeled examples.

The Active Meta-learning was empirically evaluated in [10] by using an *uncertainty sampling method* originally proposed in [8] for the k-NN algorithm. This method selects unlabeled examples for which the current k-NN learner has high uncertainty in its prediction. The uncertainty of k-NN was defined in [8] as the ratio of: (1) the distance between the unlabeled example and its nearest labeled neighbor; and (2) the sum of the distances between the unlabeled example and its nearest labeled neighbor of every class. A high value of uncertainty indicates that the unlabeled example has nearest neighbors with similar distances but conflicting labeling.

In the context of meta-learning, let E be the set of labeled meta-examples. Let $\mathcal{C} = \{c_1, \dots, c_L\}$ be the domain of the class attribute C , with L possible class labels, representing the set of candidate algorithms. Each labeled meta-example e_i is represented as the pair $(\mathbf{x}_i, C(e_i))$ storing: (1) the description \mathbf{x}_i of the problem e_i , where $\mathbf{x}_i = (x_i^1, \dots, x_i^m)$ is a vector of m meta-features; and (2) the class attribute C associated to e_i , i.e., $C(e_i) = c_l$, where $c_l \in \mathcal{C}$.

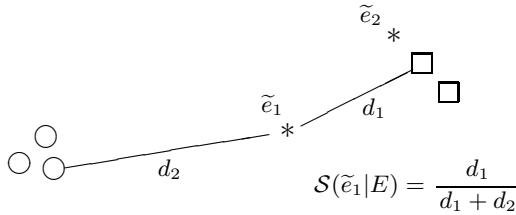
Let \tilde{E} be the set of unlabeled meta-examples. Let E_l be the subset of labeled meta-examples associated to the class label c_l , i.e., $E_l = \{e_i \in E | C(e_i) = c_l\}$. Given E , the classification uncertainty of k-NN for each $\tilde{e} \in \tilde{E}$ is defined as:

$$\mathcal{S}(\tilde{e}|E) = \frac{\min_{e_i \in E} dist(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\sum_{l=1}^L \min_{e_i \in E_l} dist(\tilde{\mathbf{x}}, \mathbf{x}_i)}. \quad (1)$$

where $\tilde{\mathbf{x}}$ is the description of the unlabeled meta-example \tilde{e} and $dist$ is the distance function adopted by the k-NN algorithm. The unlabeled meta-examples in \tilde{E} are selected according to the following probabilities:

$$p(\tilde{e}) = \frac{\mathcal{S}(\tilde{e}|E)}{\sum_{\tilde{e}_i \in \tilde{E}} \mathcal{S}(\tilde{e}_i|E)}. \quad (2)$$

The above probability is just a normalized value of the uncertainty. The roulette-wheel algorithm is often used to sample the unlabeled meta-examples according to their associated probabilities. In this method, the probability of a meta-example being sampled is proportional to its uncertainty. The selected meta-example is labeled (i.e., the class value $C(\tilde{e})$ is defined) by estimating the

**Fig. 4.** Illustration of Uncertainty Sampling

performance of the candidate algorithms on the corresponding dataset. Finally, the new labeled meta-example ($\tilde{\mathbf{x}}, C(\tilde{e})$) is then included in the meta-data.

Figure 4 illustrates the uncertainty sampling method. Circles and squares represent two classes of labeled examples. The stars named as \tilde{e}_1 and \tilde{e}_2 represent two unlabeled examples which are candidates to be labeled. The example \tilde{e}_2 would be less relevant since it is very close to the labeled examples of one specific class. In our method, the \tilde{e}_1 has a higher probability to be sampled since it is more equally distant from labeled examples of different classes.

4 Experiments and Results

Meta-data. We empirically evaluate the proposed approach exactly on the same meta-learning task used in [4]. It consists of predicting, a priori, if pruning a decision tree will improve the quality of the model or not. There are three classes, p, u or t, meaning, respectively, the winner is the pruned tree, the unpruned tree or that they are tied.

Concerning the meta-features to characterize datasets (and datasetoids), we used (1) the class entropy and (2) the average entropy of the attributes [4]. These meta-features are expected to contain some information about the behavior of decision trees because this algorithm uses the concept of entropy. But, most importantly, although there are certainly other meta-features that could contribute to improve the meta-learning results, the use of measures that previously obtained good results enables us to focus on the main goal of this paper, which is to test the combination of active meta-learning and datasetoids.

The set of problems used to generate meta-examples are 64 UCI classification datasets. By swapping the target attribute with every nominal attribute, 983 datasetoids were obtained. Table 1 presents the class distribution, both in the meta-data obtained from datasets as in the meta-data obtained from datasetoids.

Meta-level performance estimation. Given that the datasetoids are generated from datasets, they cannot be treated as independent problems. In other words,

Table 1. Class distribution (%) of the meta-data (for datasets and datasetoids)

metadata	pruned tree wins (p)	unpruned tree wins (u)	tie (t)
datasets	36	23	41
datasetoids	37	10	53

datasetoids generate meta-examples which are not independent from the meta-examples representing the corresponding datasets. Therefore, to estimate meta-learning performance, we adopted the same methodology as in [14], which is based on the following principles:

- predicting which algorithm is the best on the datasetoids is not relevant. As stated above, datasetoids are not interesting as applications *per se*. Therefore, only the original UCI datasets are used as test set.
- to ensure independence between the training and test sets, we must guarantee that the datasetoids generated from the test datasets are not included in the training set. Thus, the meta-examples corresponding to the datasetoids obtained from test datasets are removed from the training meta-data.

By removing the datasets which are obtained from test datasets, the amount of meta-data available for learning can reduce significantly. To minimize this, we use a leave-one-out (LOO) approach as in [14], which means that, at each iteration, a single dataset is used as test and the corresponding datasetoids are removed from the training set. The measure of meta-learning performance is the classification accuracy.

Active Meta-learning Setting and Baseline. Every experiment starts with a single labeled meta-example which is selected randomly. Then, we allow the active meta-learning method to sample and label up to 500 training meta-examples (about 50% of the available candidate problems). Given that the methods has two random components (selection of the first labelled example and roulette wheel), we repeat the experiments 100 times to reduce the variance of the results. The k-NN is run with a single neighbor, $k = 1$. Good results have been obtained in previous meta-learning experiments with the k-NN algorithms [4].

As a basis for comparison, we tested the Random Sampling method for selecting unlabeled problems (also repeated 100 times). Despite its simplicity, the random method has the advantage of performing a uniform exploration of the example space [8]. Another reference is the execution of the meta-learning method without active learning. In this case, we compare the active meta-learning with two results obtained by applying the meta-learning method on two different training sets: only datasets and the set of datasets and datasetoids together. The accuracies of those methods were 52% and 58%, respectively.

Results. Figure 5 presents the average curves of accuracy obtained by the two methods, uncertainty sampling and the random sampling, as well as the line representing the accuracy of meta-learning with all the meta-data. For both methods the accuracy of the k-NN meta-learner increases as the number of labeled meta-examples increases. However, the curve of the active meta-learning method is clearly above the random method. This means that the active learning method is able to identify the examples that are really helpful for the meta-learner.

When compared to the traditional meta-learning approach, the active meta-learning method is able to achieve the same level of accuracy that was obtained by meta-learning with all datasets and datasetoids (58%) using only 245 labeled meta-examples. This is less than 30% of the available problems, which represents

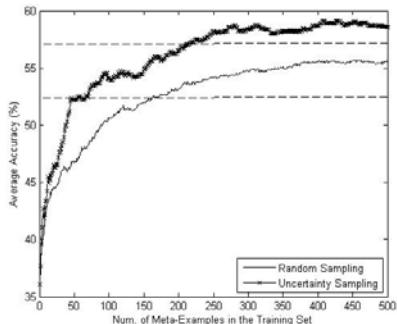


Fig. 5. Average accuracy obtained by uncertainty and random sampling. The dashed lines represent the accuracy obtained using all datasets and datasetoids (top) or just the datasets (bottom).

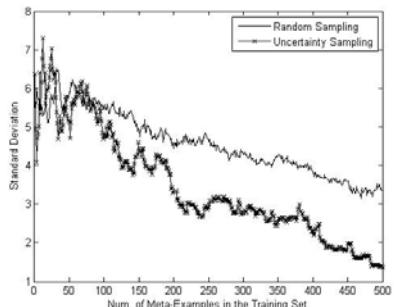


Fig. 6. Standard deviation of the accuracy obtained by uncertainty and random sampling

a significant gain in computational cost. Another interesting observation can be made. The active meta-learning can achieve an accuracy that is higher than the traditional meta-learning using all meta-examples. This can be explained by the fact that datasetoids are expected to have some noise [24]. These results indicate that the active meta-learning method may be also avoiding those examples.

To analyze the robustness of the methods, we computed the standard-deviation of their accuracies. Figure 6 shows that the standard-deviation of the two methods is similar until labeling approximately 100 meta-examples but then the uncertainty method has lower values. This indicates that, not only active meta-learning is more accurate than random sampling, as it is also more robust.

5 Conclusion

We proposed the combination of Active Meta-Learning and datasetoids to simultaneously address two important issues of meta-learning for algorithm selection: augmenting the number of datasets to produce meta-examples and reducing the computational cost of collecting meta-data. Our results show that it is possible to take advantage of the large number of meta-examples provided by datasetoids without incurring into significant extra computational costs. Additionally, if sufficient resources are available, it is even possible to achieve improvements, possibly due to the elimination of irrelevant and misleading meta-examples.

These results were obtained with a simple active learning method with known limitations (e.g., sensitivity to outliers). This opens up a number of possibilities for improvement, by using more complex active learning methods, possibly adapting them for meta-learning. Finally, we point out that we need to test the approach on other meta-learning problems, with different learning problems (e.g., regression), sets of base-level algorithms and meta-features.

Acknowledgements. The authors would like to thank CNPq, CAPES, FACEPE (Brazilian Agencies), FCT (Programa de Financiamento Plurianual de Unidades de I&D and project Rank! - PTDC/EIA/81178/2006) their financial support.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
2. Blockeel, H., Vanschoren, J.: Experiment databases: Towards an improved experimental methodology in machine learning. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 6–17. Springer, Heidelberg (2007)
3. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. Cognitive Technologies. Springer, Heidelberg (2009)
4. Brazdil, P., Soares, C., da Costa, J.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Mach. Learn.* 50(3), 251–277 (2003)
5. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. *Machine Learning* 15, 201–221 (1994)
6. Corchado, E., Abraham, A., de Carvalho, A.: Editorial: Hybrid intelligent algorithms and applications. *Inf. Sci.* 180, 2633–2634 (2010)
7. Hilario, M., Kalousis, A.: Quantifying the resilience of inductive classification algorithms. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 106–115. Springer, Heidelberg (2000)
8. Lindenbaum, M., Markovitch, S., Rusakov, D.: Selective sampling for nearest neighbor classifiers. *Machine Learning* 54, 125–152 (2004)
9. Macià, N., Orriols-Puig, A., Bernadó-Mansilla, E.: Genetic-based synthetic data sets for the analysis of classifiers behavior. In: Proceedings of 15th International Conference on Hybrid Intelligent Systems, pp. 507–512 (2008)
10. Prudêncio, R.B.C., Ludermir, T.B.: Selective generation of training examples in active meta-learning. *Int. J. of Hybrid Intelligent Systems* 5, 59–70 (2008)
11. Raghavan, H., Madani, O., Jones, R.: Active learning with feedback on both features and instances. *Pattern Recognition Letters* 7, 1655–1686 (2006)
12. Riccardi, G., Hakkani-Tur, D.: Active learning - theory and applications to automatic speech recognition. *IEEE Transactions on Speech and Audio Processing* 13(4), 504–511 (2005)
13. Sampaio, I., Ramalho, G., Corruble, V., Prudêncio, R.: Acquiring the preferences of new users in recommender systems - the role of item controversy. In: Proceedings of the ECAI 2006 Workshop on Recommender Systems, pp. 107–110 (2006)
14. Soares, C.: UCI++: Improved support for algorithm selection using datasetoids. In: Theeramunkong, T., Kijksirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 499–506. Springer, Heidelberg (2009)

A Parallel Genetic Programming Algorithm for Classification

Alberto Cano, Amelia Zafra, and Sebastián Ventura

Department of Computing and Numerical Analysis, University of Córdoba
14071 Córdoba, Spain
{i52caroa,azafra,sventura}@uco.es

Abstract. In this paper a Grammar Guided Genetic Programming-based method for the learning of rule-based classification systems is proposed. The method learns disjunctive normal form rules generated by means of a context-free grammar. The individual constitutes a rule based decision list that represents the full classifier. To overcome the problem of computational time of this system, it parallelizes the evaluation phase reducing significantly the computation time. Moreover, different operator genetics are designed to maintain the diversity of the population and get a compact set of rules. The results obtained have been validated by the use of non-parametric statistical tests, showing a good performance in terms of accuracy and interpretability.

Keywords: Genetic Programming, Classification.

1 Introduction

The general idea of discovering knowledge in large amounts of data is both appealing and intuitive, but technically it is significantly challenging and difficult, especially in the fields where really huge amounts of relational data have been collected over last decades. In this paper we focus on classification which is a well-known task in data mining.

The classification task has been overcomed with numerous computer techniques (rule learning, instance based learning, neural networks, support vector machines, statistical classifiers and so on). These include crisp rule learning [10,6], decision trees [15], evolutionary algorithms [3,4,13,20] and specifically Genetic Programming (GP) algorithms [7,11,18,19]. One of the best advantage of rule-based system is that they provide interpretable solutions to the user. When considering a rule-based learning system, the different genetic learning methods follow two approaches in order to encode rules within a population of individuals. The first one represents an individual as a rule set, this proposal is known as Pittsburgh approach [17]. The second one represents an individual as a single rule, and the whole rule set is provided by combining several individuals in a population (rule cooperation) or via different evolutionary runs (rule competition). In turn, within the individual as a single rule approach, there are three

generic proposals: Michigan, Iterative Rule Learning and Genetic Cooperative-Competitive Learning).

The main advantage of the Pittsburgh model compared to other approaches is that it allows to address the cooperation-competition problem, dealing the interaction among rules in the evolutionary process. However, its main problem is controlling the number of rules of the individuals, as the total number of rules in the population can grow quite, increasing the computational cost and becoming unmanageable problems. On the other hand, the other approaches provide good results but are inefficient methods and have addressed the cooperation-competition problem by not dealing the interaction between rules in the evolutionary process.

In this paper we propose a Grammar Guided GP (G3P) based algorithm that learns disjunctive normal form (DNF) rules generated by means of a context-free grammar, coded as one rule base decision list [16] per individual (Pittsburgh approach). This provides easily interpretable and understandable rules, and it also considers the interaction among the rules in the evolutionary process. The genetic operators are designed to work on two levels. On the one hand, it allows the optimization of particular rules by combining the rules to obtain the bests disjunction and conjunction of attribute-value comparisons. On the other hand, it address the problem of cooperation-competition, considering the interaction that occurs among the rules when these are combined to form the final classifiers.

Evolving full classifiers allows us to evaluate the relationship among rules but introduces greater complexity and computation time. To solve this problem and reduce the computation time, the evaluation phase for fitness computation is parallelized using the GPU or multiple CPU threads. Furthermore, the problem of controlling the number of rules is solved by setting a parameter with the maximum number of rules per class. This parameter allows the user to decide to obtain simpler or more complex classifiers by limiting the number of rules.

An experimental study involving 18 datasets and 11 well-known classification algorithms shows that the algorithm obtains accurate and comprehensible rules. Non-parametric statistical methods have been used to compare and analyze the accuracy of the experimental results. They show the good performance in terms of accuracy of our approach compared to other traditional methods analyzed. Moreover, the suitability of some components such as the use of specific genetic operators and the use of full classifier take advantages with respect to the rest of GP and G3P-based methods considered in the study.

The remainder of this paper is organized as follows. Section 2 presents the proposed GP classification algorithm and discusses the genetic operators, the fitness function and the generational model. Section 3 describes the experimental study. In Section 4, the results will be announced and finally the last section presents the final remarks of our investigation and outlines future research work.

2 Algorithm

This section details the algorithm proposed. It describes the encoding of the individual's genotype, the genetic operators, the fitness function, the initialization criterion and the generational model.

2.1 Individual Representation

One of the main advantages of GP is the flexibility to represent the solutions. Thus, GP can be employed to construct classifiers using different kinds of representations, e.g. decision trees, classification rules, discriminant functions, and many more [9]. In our case, the individuals in the Pittsburgh GP algorithm are classifiers where each individual is a set of classification rules generated by means of a context-free grammar and whose expression tree is composed by terminal and non-terminal nodes. A classifier can be expressed as a set of IF-antecedent-THEN-consequent rules in a decision rules list. The antecedent of the rule represents a disjunction and conjunction of attribute-value comparisons and the rule consequent specifies the class to be predicted for an instance that satisfies all the conditions of the rule antecedent. The terminals set consists of the attribute names and attribute values of the dataset being mined, logical operators (AND, OR, NOT), relational operators ($<$, $=$, $<>$, $>$) and the interval range operator (IN). We must ensure that the classifier contains at least one rule for each class.

2.2 Genetic Operators

This subsection describes the genetic crossover and mutation operators which modify the genotype of the individuals throughout the evolutionary process.

Crossover Operator

As mentioned, an individual represents a classifier as a set of rules. Taking advantage of this representation, the crossover operator is designed to optimize both the rules and the interaction among them in the classifier. So on the one hand, the crossover applied on specific rules operates on two rules from the individual and produces two new rules. Two random compatible nodes are selected from within each rule and then the resultant sub-trees are swapped, generating two child rules. These crossed rules become the decision list of a new individual. On the other hand, the crossover applied on classifiers acts over the individuals swapping their rules. Given two parent individuals, two crossing points are chosen (one by parent) so the rules are swapped from those points, building two new individuals different from their parents. Therefore, selected crossing points will ensure that at least one rule of a classifier will cross with the classification rules from the other, i.e., it is not allowed to swap all the rules of a classifier.

Mutation Operator

The mutation operator can also be applied to the rules and the individuals. The rules mutation operates either on a function node or a terminal node. It randomly selects a node in a sub-tree and replaces it with a new randomly created sub-tree. The mutation of an individual is determined by the random elimination of a rule of the rule set with a probability degree.

2.3 Fitness Function

The algorithm has two fitness functions. The first one evaluates the rules of each individual independently. The second one evaluates the individuals checking the success rates of the classifiers over the training set. It is necessary to evaluate the rules of each individual first and then evaluate the classifiers success rates.

Fitness Function Applied on Particular Rules

The fitness function we use on particular rules is the proposed by Bojarczuk et al. [11]. Specifically, each rule is evaluated over each instance and each class. This obtains the results of the quality of the predictions of the rules for each class. Thus, the consequent of a rule is reassigned to the class that has produced better results. The rules fitness function combines two indicators that are commonplace in the domain, namely the sensitivity (Se) and the specificity (Sp), which employ the true positive (t_p), false positive (f_p), true negative (t_n) and false negative (f_n) values from the match of the class of the instance and the predicted class.

$$Se = \frac{t_p}{t_p + f_n} \quad Sp = \frac{t_n}{t_n + f_p} \quad ruleFitness = Se * Sp \quad (1)$$

Fitness Function Applied on Rule Set (The Classifier)

The classifiers fitness function is performed once the best consequent for each rule is calculated. The fitness of a classifier is simply the success rate of the classifier over the training set. Each instance is submitted to the decision list to find the first rule that covers the instance. If the consequent of the rule matches the class of the instance it is a hit, otherwise it is a fail.

The activation process of the rules defines in which order the rules are evaluated to determine which rule classifies each instance. The activation method employed is a decision list. Thus, an instance is classified by the first rule of the classifier that covers the instance and whose order is defined in the individual's genotype. If no rule covers the instance, it is classified using the default class (most frequent class or defined by the user).

Parallel Evaluation Phase

Many studies have proved the evaluation phase is by far the most expensive [5] since it requires evaluating each rule over each instance. As mentioned, a disadvantage of Pittsburgh approaches is their high computational complexity. In

fact, if you analyze our proposal, the total number of rules is the addition of the number of rules for each individual. This number can be large and at least the total number of rules is the product of the number of classes and the population size. An evolutionary system that performs crossover, mutation, evaluation and selection on so many rules is really expensive.

To solve this problem, numerous studies have parallelized the evaluation phase of these algorithms. Among the references there are two main ways to perform the parallelization. One is the use of different threads and the second one in more recent works is the use of GPUs [4]. The most efficient approaches conclude they can speed up the evaluation phase more than 800 times [5]. Therefore, we will take advantage of its parallelization capability to solve the high computational cost problem of the evaluation phase, specifically in Pittsburgh approaches, by using the GPU.

Our system divides the evaluation phase into two functions: rules fitness function and individuals fitness function. The rules evaluation can be performed completely parallel using GPUs, but also the evaluation of individuals can be parallelized. Each individual can be tested as a rule over the training set independently. Therefore, our model will take advantage of both parallel fitness functions to reduce the computational cost. The full description of the parallel model would exceed the scope and the number of pages of this paper but it is detailed in [5].

2.4 Initialization

The initialization of the algorithm with a genetically diverse population is crucial for a successful outcome. The problem of the fitness function we use to evaluate the rules is that a minority class can be ignored. Therefore, we must ensure that individuals contain at least one rule for each class. One way to do this, once the individuals are created, is to complete with new rules of the classes that have not yet been covered by at least one rule. As it might be very expensive to get at least one rule for each and every one of the classes, the process of completing the individual is performed up to a number of times depending on the number of classes. This way, we try most of the classes to be represented in the classifier. Finally, among all the rules of all individuals we decide to keep the best rule for each class that will help us later to fill the classifiers with the rules of the classes that could be missing.

2.5 Generational Model

In this section we describe the generational model represented in Fig. 1. The left box represents the initialization of individuals described before. Once the population is initialized, the algorithm iteratively proceeds to apply genetic operators crossover and mutation described in the section 2.2. Specifically, the algorithm performs the individuals crossover swapping subsets of rules. The individuals can mutate eliminating some of its rules. The rules within each individual are crossed together and mutated, obtaining new classification rules. These new rules must be evaluated to get their consequents and fitness values.

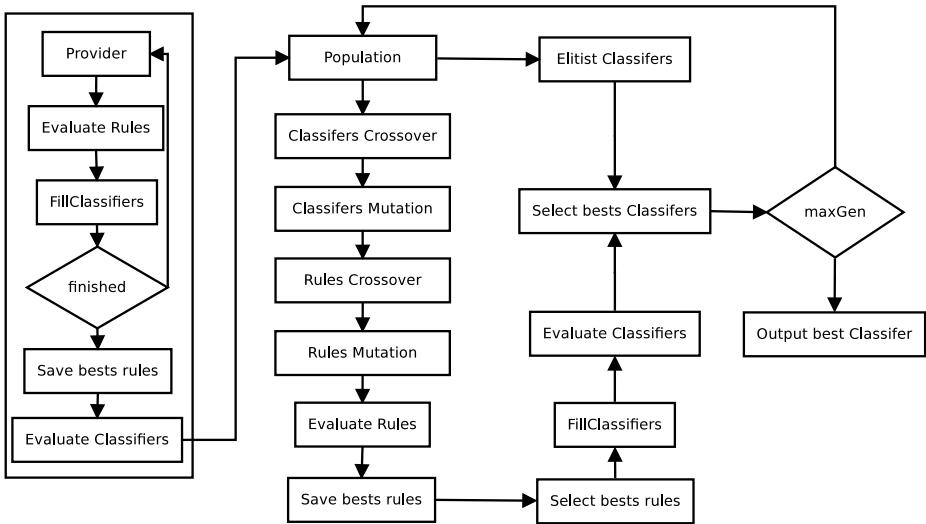


Fig. 1. Algorithm flowchart

To ensure the survival of the best rules, the algorithm checks in each generation and for each class if any new rule is better than the one stored for that class, if so the rule is replaced by the new one. As the crossover operator may have created individuals that exceed the maximum number of rules allowed, we can simplify by selecting the best rules subset.

The individual must be completed by rules from uncovered classes, taking the best rules from the pool to cover them. Once completed, each individual must be evaluated to get its fitness using the training data. We employ elitism to keep a subset of the best individuals in each generation to ensure the survival of the fittest individuals. The algorithm finishes when it has found an individual that correctly classifies all the training instances or when the algorithm has iterated a certain number of generations.

3 Experimental Study

This section describes the details of the experiments, discuss the application domains, the algorithms used and the settings of the tests.

The experiments performed compare the results of 11 different classification algorithms using 18 datasets. These algorithms are available on the JCLEC [21] website and KEEL [2] website. The datasets employed have been selected from the KEEL repository website [1]. These datasets are very varied considering different degrees of complexity, number of classes, number of features and number of instances. Thus, the number of classes ranges up to 10, the number of features ranges from 4 to 60 and the number of instances ranges from 101 to 58000.

To properly evaluate the performance of the algorithm proposed it is considered to make a comparative study with some evolutionary crisp rule learning

algorithms for classification (De Falco et al. [7], Bojarczuk et al. [11], Tan et al. [18][19], MPLCS [3], ILGA [13], CORE [20] and UCS [4]), two rule learning algorithms widely extended (PART [10] y RIPPER [6]) and a classic decision tree algorithm (C4.5 [15]). For each algorithm, the values of the parameters to be configured by the user were set to the default values provided by the authors. The population size and the number of generations for our proposal are both set to 200, i.e. the algorithm deals with 200 candidate full classifiers, and the maximum number of rules per class is set to 3. The results are validated using 10-fold cross validation and 10 runs with different seeds for stochastic methods. The results provided are the average of the 100 executions. A CPU time comparison analysis would exceed the number of pages of this paper and the execution time of the proposal is definitely higher than the other algorithms.

The experiments were executed in an Intel i7 quadcore machine with 12 GB DDR3-1600 and two NVIDIA GTX 480 GPUs, which are 3 billion transistors GPUs with 480 cores and 1.5 GB GDDR5. The operating system was Ubuntu Linux 10.10 64 bits, the GPU computing software was NVIDIA CUDA 3.1 and the compiler GCC 4.4.4.

4 Results

In this section we provide the results of the experiments and discuss the performance of the algorithms over the datasets. Table II shows the average results of the predictive accuracy obtained running each algorithm ten times in each of the datasets tests folds. The last but one row shows the average accuracy over all the datasets.

Table 1. Experiments results: predictive accuracy (%)

Dataset	Proposal	Falco	Bojarczuk	Tan [18]	Tan [19]	MPLCS	ILGA	CORE	UCS	C4.5	PART	RIPPER
Zoo	95.90%	61.73%	86.31%	93.43%	92.75%	95.50%	84.67%	94.58%	96.50%	92.80%	86.13%	93.41%
Iris	95.82%	76.20%	84.40%	89.67%	75.34%	96.00%	93.33%	94.67%	92.00%	96.67%	33.33%	94.00%
Hepatitis	88.51%	68.85%	74.44%	73.50%	83.43%	85.15%	77.92%	83.43%	80.74%	85.66%	84.17%	79.09%
Wine	94.58%	62.72%	79.33%	75.81%	66.24%	91.54%	89.28%	95.49%	91.57%	94.90%	63.98%	93.79%
Sonar	75.09%	61.90%	66.31%	60.57%	59.05%	76.81%	71.57%	53.38%	77.83%	70.54%	60.53%	72.45%
Glass	69.39%	31.79%	39.98%	36.88%	48.85%	65.88%	52.40%	50.97%	66.07%	68.86%	46.50%	63.47%
New-thyroid	94.40%	69.18%	75.54%	73.65%	78.66%	91.65%	90.71%	91.21%	93.55%	93.52%	77.22%	93.08%
Heart	81.85%	66.78%	74.37%	72.55%	76.30%	83.70%	64.44%	71.11%	80.37%	78.14%	57.77%	76.29%
Dermatology	94.96%	45.61%	73.59%	77.29%	76.56%	95.53%	60.81%	43.86%	96.36%	94.35%	73.12%	94.42%
Haberman	73.82%	65.92%	68.43%	52.70%	70.18%	72.17%	71.89%	70.88%	72.18%	71.19%	73.53%	46.72%
Ecoli	78.92%	51.72%	57.63%	50.48%	65.19%	80.67%	63.10%	65.78%	79.49%	77.37%	44.67%	72.63%
Australian	85.41%	72.42%	84.29%	75.77%	78.41%	86.96%	85.07%	83.33%	86.09%	85.21%	60.72%	81.44%
Pima	75.01%	69.82%	67.59%	68.98%	66.68%	74.60%	73.19%	72.28%	76.04%	72.53%	65.11%	69.53%
Vehicle	70.59%	31.11%	41.67%	36.04%	41.52%	71.15%	57.24%	40.05%	72.45%	66.66%	37.85%	70.44%
Contraceptive	55.61%	40.75%	42.68%	40.37%	44.00%	55.47%	43.59%	45.01%	49.49%	51.19%	42.90%	50.78%
Thyroid	99.22%	68.05%	51.39%	52.92%	92.43%	94.72%	94.10%	68.00%	96.99%	99.56%	92.58%	99.37%
Penbased	83.32%	25.54%	40.29%	35.76%	44.90%	91.80%	53.25%	15.69%	14.28%	94.89%	15.86%	96.15%
Shuttle	99.97%	61.16%	75.51%	63.55%	89.51%	99.60%	93.67%	91.60%	99.62%	99.96%	99.60%	99.96%
Average (%)	84.02%	57.34%	65.76%	62.44%	69.44%	83.82%	73.34%	68.40%	78.98%	83.00%	61.97%	80.38%
Ranking	2.22	10.56	8.72	9.72	8.08	3.03	7.00	7.25	3.56	3.58	9.19	5.14

Analyzing the results of the table notice that the algorithm proposed obtains better average accuracy results and ranking. Its accuracy is higher than most algorithms and is very close and slightly higher than the algorithms MPLCS, UCS and C4.5. In those datasets in which our algorithm does not achieve the best results, its accuracy is usually quite competitive. Moreover, the algorithm does not stand out negatively in any dataset.

In order to analyze the results and discover the existence of significant differences between the algorithms, a series of statistical tests [8][12] were carried out. We use the Iman and Davenport test to rank the k algorithms over the N datasets. The average rank according to the F-distribution throughout all the datasets is shown in the last row of the table. Notice that the best global position, the lowest ranking value, corresponds to the obtained by our proposal.

The Iman and Davenport statistic for average accuracy distributed according to F-distribution with $k - 1 = 11$ and $(k - 1)(N - 1) = 187$ degrees of freedom is 30.1460. This value does not belong to the critical interval $[0, F_{0.01,11,187} = 2.3439]$ for $p = 0.01$. Thus, we reject the null-hypothesis that all the algorithms perform equally well. In order to analyse if there are significant differences among the algorithms, we use the Bonferroni-Dunn test to reveal the difference in performance using a critical difference (CD) value for $p = 0.01$ equal to 3.9865.

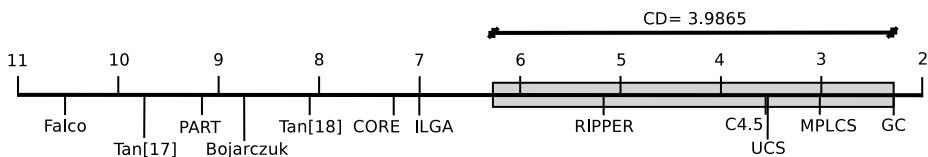


Fig. 2. Bonferroni-Dunn test. The noncritical range is shown shaded.

The results indicate that a significance level of $p = 0.01$ (i.e., with a probability of 99%), there are significant differences between our algorithm and De Falco et al., Bojarczuk et al., Tan et al. [18][19], PART, IGLA and CORE algorithms being our algorithm statistically better. These differences are shown in Fig. 2. The figure represents the algorithm's ranking and the critical distance interval. Regarding to the other algorithms, the test does not indicate significant differences. However, our proposal obtains the lowest ranking value, indicating that considering all datasets, it obtains better results in a greater number of them than other proposals.

5 Conclusions

In this paper we have proposed a G3P algorithm that optimizes the class prediction accuracy of a decision list as a set of interpretable DNF rules. The encoding of the individuals have allowed us to design specific crossover and mutation operators that consider the interaction among the rules in the evolutionary process. The algorithm overcomes the high computational complexity of Pittsburgh approaches by the parallelization of the evaluation phase. Nevertheless, in spite of this improvement, the complexity of the algorithm is still high. The experiments carried out compare the results of our proposal with other classic algorithms for solving classification problems considering 11 algorithms and 18 datasets. A statistical study on results validates the model showing that it provides the most accurate classifiers since obtaining the lowest ranking in the general comparative.

The algorithm design is complex and the execution time is high even using GPUs for the evaluation of the individuals. The individual representation is as a set of classification rules and it is necessary to evaluate both rules and classifiers every generation. Therefore, the algorithm performs slow regarding to the other algorithms from the experimental. Nevertheless, the idea of this work was to experiment a highly complex G3P Pittsburgh algorithm for classification accelerated using GPUs, since GPUs have been successfully used in Michigan approaches to speed up the evaluation phase [5]. Therefore, there is no a sequential implementation of the algorithm to compare with, since it would require excessive CPU time.

Currently, databases are moving toward sparse data, unbalanced, multiple labels, instances. It would be interesting to check the performance of the algorithm in these contexts and to improve the efficiency of the algorithm to perform faster.

Additional information of the paper such as the algorithm's code, the experimental setup and the datasets are published and available on the website: <http://www.uco.es/grupos/kdis/kdiswiki/HAIS11>

Acknowledgments. This work has been financed in part by the TIN2008-06681-C06-03 project of the Spanish Inter-Ministerial Commission of Science and Technology (CICYT), the P08-TIC-3720 project of the Andalusian Science and Technology Department and FEDER funds.

References

1. Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing* 17, 255–287 (2011)
2. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M., Ventura, S., Garrell, J., Otero, J., Romero, C., Bacardit, J., Rivas, V., Fernández, J., Herrera, F.: KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems. *Soft Computing - A Fusion of Foundations, Methodologies and Applications* 13, 307–318 (2009)
3. Bacardit, J., Krasnogor, N.: Performance and efficiency of memetic pittsburgh learning classifier systems. *Evolutionary Computation* 17(3), 307–342 (2009)
4. Bernadó-Mansilla, E., Garrell, J.M.: Accuracy-based learning classifier systems: Models and analysis and applications to classification tasks. *Evolutionary Computation* 11(3), 209–238 (2003)
5. Cano, A., Zafra, A., Ventura, S.: Solving classification problems using genetic programming algorithms on gPUs. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 17–26. Springer, Heidelberg (2010)
6. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the 12th International Conference on Machine Learning, pp. 115–123. Morgan Kaufmann, San Francisco (1995)
7. De Falco, I., Della Cioppa, A., Tarantino, E.: Discovering interesting classification rules with genetic programming. *Applied Soft Comput.* 1(4), 257–269 (2001)

8. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30 (2006)
9. Espejo, P.G., Ventura, S., Herrera, F.: A Survey on the Application of Genetic Programming to Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 40(2), 121–144 (2010)
10. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: *Proceedings of the 15th International Conference on Machine Learning*, pp. 144–151 (1998)
11. Freitas, A.A.: *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag New York, Inc., Secaucus (2002)
12. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. *Journal of Heuristics* 15, 617–644 (2009)
13. Guan, S.U., Zhu, F.: An incremental approach to genetic-algorithms-based classification. *IEEE Transactions on Systems and Man and Cybernetics and Part B* 35(2), 227–239 (2005)
14. Harding, S.: Genetic programming on graphics processing units bibliography, <http://www.gpgpu.com/>
15. Quinlan, J.: *C4.5: Programs for Machine Learning* (1993)
16. Rivest, R.L.: Learning decision lists. *Mach. Learn.* 2, 229–246 (1987)
17. Smith, S.F.: *A Learning System Based on Genetic Adaptive Algorithms*. Phd thesis, University of Pittsburgh (1980)
18. Tan, K.C., Tay, A., Lee, T.H., Heng, C.M.: Mining multiple comprehensible classification rules using genetic programming. In: *Proceedings of the Evolutionary Computation CEC 2002*, pp. 1302–1307. IEEE Computer Society, Washington, DC, USA (2002)
19. Tan, K.C., Yu, Q., Heng, C.M., Lee, T.H.: Evolutionary computing for knowledge discovery in medical diagnosis. *Artificial Intelligence in Medicine* 27(2), 129–154 (2003)
20. Tan, K.C., Yu, Q., Ang, J.H.: A coevolutionary algorithm for rules discovery in data mining. *International Journal of Systems Science* 37(12), 835–864 (2006)
21. Ventura, S., Romero, C., Zafra, A., Delgado, J.A., Hervás, C.: JCLEC: a Java framework for evolutionary computation. *Soft. Comput.* 12, 381–392 (2007)

Evolutionary Algorithm for P2P Multicasting Network Design Problem

Michał Wiśniewski and Krzysztof Walkowiak

Department of Systems and Computer Networks, Faculty of Electronics,
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Krzysztof.Walkowiak@pwr.wroc.pl

Abstract. P2P multicasting is a novel approach to provide cost effective streaming services over the Internet. This paper addresses the overlay network design for P2P multicasting. We assume that the P2P multicasting system is static with low membership change rate (e.g., corporate videoconferencing, distance learning, delivery of important messages, IPTV with STB). The problem consists in joint optimization of multicast flows and access link capacity. The objective is to minimize the network cost calculated as the cost of access links. An effective evolutionary algorithm is proposed. We report results of the algorithm with comparison against optimal results and other heuristic algorithms including Lagrangean relaxation algorithm and constructive algorithm. The proposed algorithm can be used to various problems related to overlay networks, e.g., streaming services, computing and storage systems.

Keywords: evolutionary algorithm, P2P multicasting, optimization, overlay.

1 Introduction

Recently, numerous streaming services have been gaining much popularity in the Internet. The most efficient way to distribute the streaming content to a large number of requesting users is multicast transmission. Since the IP multicast produces many technological and business challenges (e.g., difficulties with multi-domain multicasting – construction of multicast is limited to only one operator), new ideas are examined to assure effective streaming. One of the most promising approaches is P2P multicasting (called also application-layer multicasting) that provides an universal framework to develop various streaming services including IPTV, VoD (Video on Demand), IPTV, distance learning, podcasting, etc. The P2P multicasting is arranged on the top of an overlay network. The streaming content is delivered to requesting users by the use of multicast trees constructed among peers (end hosts). The main idea of P2P multicasting is that each participant of the system can both: receive (download) the streaming signal and transmit (upload) the same signal to subsequent users. In contrast to traditional IP multicast realized in the network layer, the P2P multicasting assumes that uploading (non-leaf) nodes can be normal end hosts [1-3].

We assume that the P2P multicasting is applied in relatively static streaming systems with low membership change rate, e.g., corporate videoconferencing, personal

video broadcast in small groups, collaborated workgroup, delivery of important messages (weather forecast, emergency alerts) to local governments, distance learning [1]. This means that users of the system remain stable connected and there are not dynamic changes of the structure as in file-sharing P2P systems. Since most of previous papers on optimization of P2P multicasting systems focus on optimization only of network flows, we noticed the problem of joint optimization of link capacity and P2P flows and formulated a problem of network design for P2P multicasting [4]. The optimization objective is to minimize the network cost and providing the streaming to all participants. Our first work on this subject [4] includes the sketch formulation of the problem and Lagrangean relaxation (LR) technique. In the next paper we proposed a construction heuristic algorithm [5]. The main contribution of this work is an evolutionary algorithm designed for the same problem. According to experiments, the evolutionary algorithm provides results very close to optimal (the optimality gap is less than 1.5%) and outperforms previous methods.

2 P2P Multicasting Network Design Problem

In this section we formulate an ILP model of the P2P multicasting network design problem [4-5]. Assumptions of the model follow from real P2P systems. The system consists of n nodes (peers) indexed by $v = 1, 2, \dots, V$, which are to participate in the streaming system. One of the peers is assigned to be the source of streaming (root node) and is denoted by $r_v = 1$. Each peer can communicate with another peer through an overlay network (e.g., Internet). Zhu and Li show in [6] that nodes' capacity constraints are typically sufficient in overlay networks. In addition, usually the physical network underlying overlay is considered to be overprovisioned and the only bottlenecks of network transmissions are access links, while the connection inside the overlay are dimensioned to avoid congestion [1-3], [7]. The goal of the optimization is to select the access link for each peer. We assume that each peer can select the access link from the price list of one (or more) Internet Service Provider (ISP). Let y_{vk} denote a binary decision variable, which is 1 if node v is connected to the overlay using access link of type k ; 0 otherwise. For each access link type available for peer v we are given the download capacity (denoted as d_{vk}), the upload capacity (denoted as u_{vk}) and cost (denoted as ξ_{vk}) expressed in euro/month.

The streaming is provided by P2P multicasting, i.e., a peer connected to the system downloads the streaming data from its parent node and additionally it can upload the stream to other subsequent nodes. An important issues related to P2P multicasting is fairness, i.e., each participant of the system should transmit similar amount of the volume as it receives. Thus, in place of a single tree, multiple trees are constructed indexed $t = 1, 2, \dots, T$. Consequently, each peer receives portions of the stream via different routes. This can be provided by special coding approaches that have been developed for multiple P2P trees transmissions, e.g., [1-2]. [7]. According to [8], another motivation to use multiple streaming trees is to improve the system resilience.

The multicast trees constructed to provide streaming can be decomposed into levels indexed $l = 1, 2, \dots, L$ – the root node is located on level 1, children of the root (peers that have a direct overlay link from the root) are on level 2, etc. The proposed notation enables us to set the value of L as a limit on the maximal depth of the tree.

Smaller number of tree levels usually means lower latency in network delivery of the streaming content. To model P2P multicast flows we use binary variable x_{vwlt} that is 1 if in tree t there is a link from node (peer) v to node w and v is located on level l of tree t ; 0 otherwise (binary). Peers – besides participating in overlay trees – can also use other network services and resources. Therefore, for each peer we are given constants a_v and b_v denoting download and upload background traffic.

Indices

$v, w = 1, 2, \dots, V$	overlay nodes (peers)
$k = 1, 2, \dots, K_v$	access link types for node v
$t = 1, 2, \dots, T$	multicast trees
$l = 1, 2, \dots, L$	levels of the multicast tree

Constants

a_v	download background transfer of node v
b_v	upload background transfer of node v
ξ_{vk}	lease cost of link of type k for node v (euro/month)
d_{vk}	download capacity of link of type k for node v (bps)
u_{vk}	upload capacity of link of type k for node v (bps)
r_v	= 1, if node v is the root of all trees; 0, otherwise
q_t	streaming rate of tree t (bps)
M	large number

Variables

x_{vwlt}	= 1, if in multicast tree t there is a link from node v to node w and v is located on level l of tree t ; 0, otherwise (binary)
y_{vk}	= 1, if node v is connected to the overlay network by a link of type k ; 0, otherwise (binary)

Objective

It is to find assignment of access link capacity for each peer and P2P multicast trees satisfying problem constraints (see below) and minimizing the linear cost:

$$F = \sum_v \sum_k y_{vk} \xi_{vk}. \quad (1)$$

Constraints

a) Each node $w = 1, 2, \dots, V$ – except the root node of the tree ($r_w = 1$) – must have exactly one parent node regarding each tree $t = 1, 2, \dots, T$:

$$\sum_{v \neq w} \sum_l x_{vwlt} = (1 - r_w) \quad \text{for each } w = 1, 2, \dots, V \quad t = 1, 2, \dots, T. \quad (2)$$

b) Node v can be the parent of the first level link, only if it is the root node:

$$\sum_{w \neq v} \sum_l x_{vwlt} \leq M r_v \quad \text{for each } v = 1, 2, \dots, V. \quad (3)$$

c) Node v cannot be a parent node on level $(l + 1)$, if it is not a child on the level l :

$$\sum_{w \neq v} x_{vwlt(l+1)} \leq M \sum_{w \neq v} x_{vwlt} \quad (4)$$

for each $v = 1, 2, \dots, V$, $t = 1, 2, \dots, T$ and $l = 1, 2, \dots, L - 1$.

d) Exactly one access link type is to be selected for each overlay node:

$$\sum_k y_{vk} = 1 \quad \text{for each } v = 1, 2, \dots, V. \quad (5)$$

e) Download capacity constraint – background traffic and streaming traffic (left-hand side) cannot exceed the link capacity (right-hand side):

$$a_v + \sum_l q_l \leq \sum_k y_{vk} d_{vk} \quad \text{for each } v = 1, 2, \dots, V. \quad (6)$$

f) Upload capacity constraint – background traffic and uploaded streaming traffic (left-hand side) cannot exceed the link capacity (right-hand side):

$$b_v + \sum_{w \neq v} \sum_l \sum_k x_{vwkl} q_l \leq \sum_k y_{vk} u_{vk} \quad \text{for each } v = 1, 2, \dots, V. \quad (7)$$

3 Evolutionary Algorithm

The problem (1)-(7) is NP-complete, since it can be reduced to the Hop-Constrained Minimum Spanning Tree Problem (HMSTP) [9]. Therefore, to solve this problem (1)-(7) in optimal way branch-and-bound or branch-and-cut methods must be applied. However, due to complexity of the problem, only for relatively small networks optimal results can be found in reasonable time. In [4] we presented optimal solutions for networks consisting of 20 nodes obtained in execution times up to hours. Thus, effective heuristics are required to solve larger problem instances, which arise in real scenarios. In this section we introduce an evolutionary algorithm [10] to tackle problem (1)-(7). Various optimization problems have been successfully attacked by hybrid intelligence including evolutionary methods, e.g. [11-13]. Note that in [14] the authors propose an evolutionary algorithms including local search for the minimum routing cost spanning tree problem, which is related to problem (1)-(7).

3.1 Chromosome Coding

First, we present the chromosome coding. We assume that the chromosome includes information related only to multicast routing (denoted by variables x in the model). The information defining access link capacity (denoted by variables y in the model) is determined later, according to the obtained multicast routing. The chromosome has the size of VT genes, i.e., for each tree $t = 1, 2, \dots, T$ there is one gene for each node $v = 1, 2, \dots, V$. The value of the gene denotes the number of children nodes of the particular node defined as the number of outgoing links in the multicast tree leaving the particular node. For instance, let's consider a network consisting of $V = 5$ nodes and one tree ($T = 1$). The chromosome 21100 denotes a tree, in which node $v = 1$ has 2 children. Nodes $v = 2$ and $v = 3$ have one child. Finally, nodes $v = 4$ and $v = 5$ are leaf nodes and do not have outgoing links. The sum of all genes related to one tree must be $(V - 1)$, as all nodes except the root node must be connected to the multicast tree. The proposed coding significantly reduces the solution space. For instance in the case of 20 nodes and 1 tree the solution space of the proposed coding is about 10^{10} . In contrast, if we use coding when each gene denotes just the index of the parent node, the corresponding size of the solution space is about 10^{26} .

3.2 Fitness Function

Recall that the coding used in our algorithm does not define directly the precise configuration of the multicast tree as well as the link capacity. Therefore, the following procedure is applied to find the objective value of a particular chromosome denoting one unique solution. For each tree $t = 1, 2, \dots, T$ the same procedure is repeated. First, we find in the chromosome a gene (except the gene denoting the root node that is fixed and located on level $l = 1$) with the highest value that is equivalent to a node with the largest number of children. This node is connected to the root node of the current tree and is located on the level $l = 2$. We repeat the procedure for subsequent nodes. When the root node is saturated, i.e., no more nodes can be connected to the root due to the limited number of outgoing connections, we start a next level of the tree, and assign subsequent nodes to nodes already connected to the tree satisfying the limit of outgoing connections. Continuing this procedure each tree is constructed. If the level limit is violated, all nodes that are located on non-feasible levels are re-connected randomly to nodes located close to the root node (i.e., node with low value of the level). Next, for each node $v = 1, 2, \dots, V$ we select the cheapest access link that satisfies the requested capacity according to constructed trees.

3.3 Operators

The algorithm starts with a population of randomly generated chromosomes (Fig. 1). Next, to find a new population we apply classical operators (crossover and mutation) as well as we use the elite promotion, i.e., the best chromosome is copied from the old population to the next one. The roulette mechanism is applied to select chromosomes for crossover operation. However, we modify the roulette and use exponential function modifier to increase the fitness of each chromosome and promote the best chromosomes, which is especially important in the case when the differences between fitness function of chromosomes are small. The crossover operator combines two parent chromosomes $C1$ and $C2$ into a one, new individual $C3$. The following procedure is repeated for each tree $t = 1, 2, \dots, T$. We take the first gene of $C1$ and put as the first gene of $C3$. Then, we take the last gene from $C2$ and put as the last gene of $C3$. Next, we take the second gene of $C1$ and copy as the second gene of $C3$, and the gene one before the last one of $C2$ and copy in the same place in $C3$, etc. We carry on the process until the sum of genes in $C3$ exceeds the required number of connections in the tree given by $(V - 1)$. If it is needed, the obtained chromosome is adjusted to yield a feasible solution – if the number of outgoing links is too high, some randomly selected genes are decreased. The mutation – according to the given probability – decreases one randomly selected gene. Next, to compensate the reduced number of outgoing connections, a randomly chosen gene is increased.

3.4 Performance Improvements

Moreover, to improve performance of the algorithm we use the following operators:

- Elite promotion – the best chromosomes from the current solution are copied to the next population.
- Global mutation – some chromosomes are generated randomly and added to the new population.

- Roulette selection limit – in the roulette process we omit chromosomes with the fitness function larger than $\alpha \text{AVG_FIT}$, where α is the selection limit and AVG_FIT denotes the average value of the fitness function in the population.

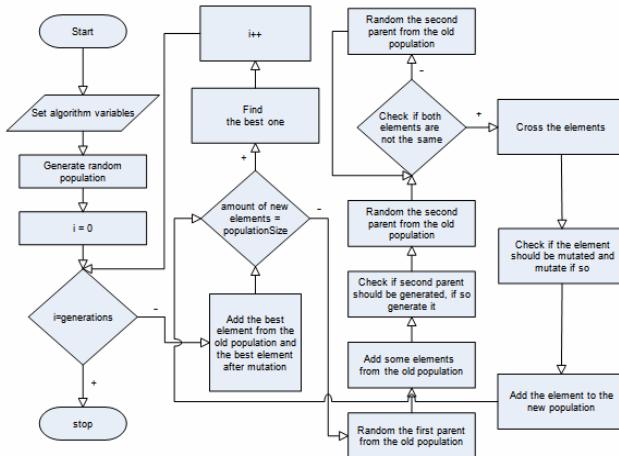


Fig. 1. Evolutionary algorithm

4 Results

The evolutionary algorithm introduced in the previous section was implemented in C++. In numerical experiments we applied DSL price lists of four ISPs: two of them operate in Poland (TP and Dialog) and two other operate in Germany (DT and Arcor). Each node was randomly assigned to one of ISPs. The values of download and background transfer were chosen at random (512 kbps or 1024 kbps). Analogously, the values of upload background transfer were selected randomly (64 kbps or 128 kbps). We generated tree sets of networks: 20 nodes, 100 nodes and 300 nodes. In each case the overall streaming rate (sum over all q_i) was set to 360 kbps. The number of trees was in the range 1-6, the number of levels was in the range 2-9.

The first objective of numerical experiments was tuning of the algorithm to examine performance of the heuristic against optimal results and other approaches. In Table 1 we present tuning related to 300 nodes networks, for each parameter we show the best value (column 2) and all tested values (column 3).

In Table 2 we report comparison of the following four algorithms made on 20-nodes networks: branch-and-cut method included CPLEX 11.0 solver [15] with optimality guarantees (OPT), Lagrangean relaxation (LR) algorithm presented in [4], MinCost heuristic (MC) proposed in [5] and evolutionary algorithm (GEN) presented in this paper. On average the evolutionary approach gives results 1.46% worse than optimal ones (OPT). On the other hand, our algorithm outperforms other heuristics on average by 2.43% (MC) and 2.36% (LR). Experiments were made on a PC with Windows Vista, 2GHz processor and 4GB RAM. The average execution times for optimal, LR, MC and evolutionary approaches were: 393 s, 12s, 0.08 s and 120 s,

Table 1. Algorithm's tuning – best values of parameters obtained for 300 nodes networks

Parameter name	Best value	Tested values
First population size	40000	1000, 5000, 10000, 15000, 20000, 40000, 50000, 60000, 80000, 100000
Population size	25000	1000, 5000, 10000, 15000, 20000, 25000, 40000, 55000
Roulette modifier	60	1, 10, 30, 50, 60, 70, 90, 120, 150
Mutation probability	0.1%	0%, 0.1%, 0.2%, 0.3%, 0.5%, 0.7%, 1%, 5%, 10%
Selection limit	0.95	0.8, 0.9, 0.95, 1, 1.05, 1.1, 1.2, 5, 10
Elite promotion	10%	0%, 1%, 2%, 3%, 5%, 7%, 10%, 15%, 20%
Global mutation	5%	0%, 1%, 2%, 5%, 10%, 15%

Table 2. Comparison of various algorithms for 20 nodes networks

Trees	Levels	OPT	LR	MC	GEN	GEN to OPT	LR to GEN	MC to GEN
1	2	1232	1257	1262	1232	0.0%	2.0%	2.4%
1	3	819	869	869	824	0.6%	5.2%	5.2%
1	4	635	670	670	635	0.0%	5.2%	5.2%
1	5	635	660	645	635	0.0%	3.8%	1.6%
2	2	809	854	869	809	0.0%	5.3%	6.9%
2	3	620	670	829	620	0.0%	7.5%	25.2%
2	4	620	625	625	620	0.0%	0.8%	0.8%
2	5	620	630	625	620	0.0%	1.6%	0.8%
4	2	660	854	854	809	22.6%	5.3%	5.3%
4	3	605	625	625	620	2.5%	0.8%	0.8%
4	4	605	625	625	615	1.7%	1.6%	1.6%
4	5	605	620	605	615	1.7%	0.8%	-1.7%

Table 3. Comparison of heuristic algorithms for various networks

Network size	# of tests	LR to GEN	GEN>LE	MC to GEN	GEN>MC
20	28	2.36%	23	2.43%	22
100	15	2.62%	12	3.56%	11
300	25	N/A	N/A	4.22%	11

respectively. Concluding, for small network the evolutionary algorithm provides the best heuristic results, however it requires the largest running time among heuristics.

Comparison of heuristic algorithms for various networks is shown in Table 3. We report the average distance between evolutionary algorithm and other heuristics LR and MinCost (columns 3 and 5, respectively) as well as the number of tests where the evolutionary approach outperforms other algorithms (columns 4 and 6). Considering the execution time for medium (100 nodes) and large (300 nodes) networks, the relation between tested heuristics was comparable to the values reported above in the context of 20 nodes networks. Due to large complexity, in the case of large networks (300 nodes), the LR algorithm cannot provide results in reasonable time of 1 hour.

5 Concluding Remarks

We have proposed a new evolutionary algorithm for the P2P multicasting network design problem. Intensive tuning of various parameters has been performed to find the

most effective combination providing the best results. Next, we have run numerical experiments to compare the proposed method against benchmark algorithms: branch-and-cut method (CPLEX), Lagrangean relaxation algorithm and constructive algorithm. The obtained results prove effectiveness of the evolutionary approach – the distance to optimal results is less than 1.5%, and other heuristics are outperformed.

Acknowledgments. This work is supported in part by The Polish Ministry of Science and Higher Education.

References

1. Buford, J., Yu, H., Lua, E.: P2P Networking and Applications. Morgan Kaufmann, San Francisco (2009)
2. Shen, X., Yu, H., Buford, J., Akon, M. (eds.): Handbook of Peer-to-Peer Networking. Springer, Heidelberg (2009)
3. Tarkoma, S.: Overlay Networks: Toward Information Networking. Auerbac (2010)
4. Walkowiak, K.: Network Design Problem for P2P Multicasting. In: International Network Optimization Conference INOC (2009)
5. Walkowiak, K.: P2P Multicasting Network Design Problem - Heuristic Approach. In: 1st IEEE Workshop on Pervasive Group Communication, Globecom, pp. 1553–1557 (2010)
6. Zhu, Y., Li, B.: Overlay Networks with Linear Capacity Constraints. IEEE Transactions on Parallel and Distributed Systems 19(2), 159–173 (2008)
7. Akbari, B., Rabiee, H., Ghanbari, M.: An optimal discrete rate allocation for overlay video multicasting. Computer Communications 31(3), 551–562 (2008)
8. Birrer, S., Bustamante, F.: Resilience in Overlay Multicast Protocols. In: IEEE Sym. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (2006)
9. Gouveia, L., Simonetti, L., Uchoa, E.: Modelling Hop-Constrained and Diameter-Constrained Minimum Spanning Tree Problems as Steiner Tree Problems over Layered Graphs. Mathematical Programming (2009)
10. Michalewicz, Z.: Evolutionary Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1999)
11. Pióro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufmann, San Francisco (2004)
12. Woźniak, M.: Evolutionary approach to produce classifier ensemble based on weighted voting. In: World Con. on Nature & Biologically Inspired Computing, pp. 648–653 (2009)
13. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
14. Merz, P., Wolf, S.: Evolutionary Local Search for Designing Peer-to-Peer Overlay Topologies Based on Minimum Routing Cost Spanning Trees. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 272–281. Springer, Heidelberg (2006)
15. ILOG AMPL/CPLEX software, <http://www.ilog.com/products/cplex/>

A Focused Wave Front Algorithm for Mobile Robot Path Planning

Anshika Pal, Ritu Tiwari, and Anupam Shukla

Soft computing and Expert System Laboratory
ABV-Indian Institute of Information Technology and Management, Gwalior, India
`{anshika, ritutiwari, anupamshukla}@iiitm.ac.in`

Abstract. Robot path planning is about finding a collision free motion from one position to another. The wave front expansion is commonly used for path planning tasks and appreciated for its efficiency. However, this approach requires full wave expansion, which takes considerable amount of time and process, in large scale environments. This paper presents a new method for motion planning of mobile robots based on wave expansion approach which avoids full wave expansion. This method imposes a cost function, that focuses on some of the waves for expansion instead of trying to expand the entire waves, as so-called focused wave front expansion algorithm. The proposed approach has been tested through computer simulation, with a set of environments with different levels of complexity depending on the density of the obstacles. Finally, we compare the results of our proposed method to that of the existing wave front method.

Keywords: Path planning, wave front, mobile robot.

1 Introduction

Robotic Path Planning is one of the problems in the field of robotics that tries to find and optimize the path from the initial position to the final position. Commonly, there are many paths for robot to accomplish the task, but in fact the best path is selected according to some guide line. These guide lines are : shortest path, least energy consuming or shortest time. So, the robot path planning is a constrained optimization problem [1]. Besides optimization, it needs to be ensured that the robot moves without any collision in the entire path it follows from the source to the destination. This would mean that the algorithm avoids all obstacles and reaches the destination starting from the source. This is also referred to as the navigation plan of the robot.

The algorithms for path planning must be evaluated in terms of completeness and computational complexity. An algorithm is said to be complete if it returns a valid solution to the path-planning problem if one exists and returns failure if and only if the problem is not feasible. This is what we will call a correct termination for a path-planning algorithm. The computational complexity of the algorithm is formulated by various problem specific performance metrics that are used for evaluation purpose [2].

The problem has been solved using numerous statistical, soft computing and hybrid approaches [3, 10, 14]. In this paper, a new path planning method using wave front

expansion, for mobile robots is presented. The algorithm returns the complete path if one exists between the source and the destination. If however, no path is possible between the source and the destination, the algorithm returns null. The proposed model algorithm is easy to implement and computationally efficient. Simulation results show that the proposed method is capable of planning collision-free path in various environments.

The paper is organized as follows. Section 2 summarizes the related work. The proposed approach is described in section 3. Experimental results are presented in section 4. Finally conclusion is given in section 5.

2 Related Work

The problem of path planning has been a very active area of research. There are so many algorithms something more Robot Path finding in static and dynamic environments [4] such as A*, artificial potential field and wave expansion also several methods exist for improving performance and increasing execution speed those [5], [6] and [7].

The research on robot path planning can be categorized into two models, based on different assumptions about the information available for planning: (1) path planning with complete information; and (2) path planning with incomplete information. The first model assumes that a robot has perfect information about itself and its environment. Information that fully describes the sizes, shapes, positions, and orientations of all obstacles in two dimensional (2D) or three-dimensional (3D) space is known. Because full information is assumed, the path planning is a one-time and off-line operation [8, 9].

In the second model, an element of uncertainty is present, and the missing data is typically provided in real time by some source of local information through sensory feedback using an ultrasound range or a vision module. A robot has no information on its environment except a start position and a target position. The sensory information is used to build a global model for path planning in real time. The path planning is a continuous on-line process. The concept of path planning algorithms in this category mostly comes from [11, 12, 13].

In static environment, many algorithms have been implemented and results verified [15, 16, 17]. In planning dynamic environment the steps are a little different, as the environment continuously changes [18, 19].

In this paper we have modified the wave front algorithm for fast path planning.

3 Proposed Approach

3.1 Environment Representation

The environment is a 2D plane and is occupied by stationary obstacles. The environment includes a Start (S) and a Target (T) point. It is assumed that there are only a finite number of obstacles in the environment. The working space is partitioned into a grid of square cells, and a M X N board is gotten. If there is no obstacle in a cell, the cell is called free, otherwise called obstacle cell (Fig.1).

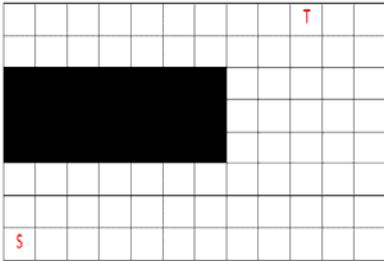


Fig. 1. Different state values. Obstacles are colored black, free space remains white. ‘S’ is start and ‘T’ is target.

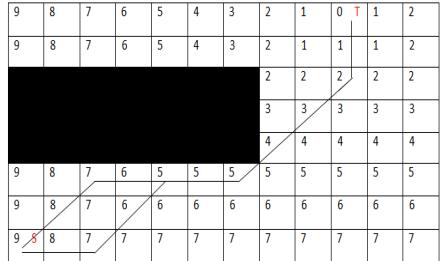


Fig. 2. Simplified Wave Front Planning (two of the possible path are shown by Black line segments)

3.2 Wave Front Based Path Planning Algorithm

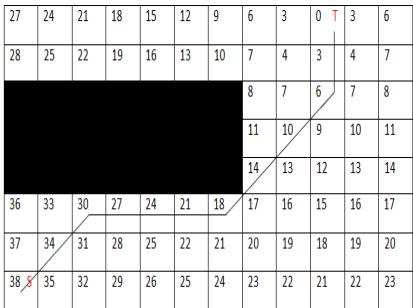
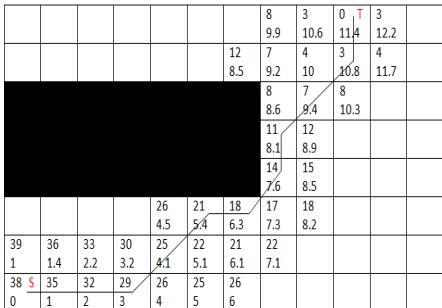
The wave front based path-planning method considers the task of path planning as finding paths from the goal location back to the start location. This planning procedure works well in the environment with a uniform grid by propagating the waves through free space from the goal cell. Generally, from the goal cell, the values are spread to the start cell with adding one. A cell will be assigned with a value when it is not an obstacle and not assigned a given value yet. The value of each cell of the grid map will be measured based on this formula [20]:

$$Map(i,j) = \begin{cases} \text{Min}(neighborhood(i,j)) + 1 & \text{Empty} \\ \text{Nothing} & \text{Fill} \end{cases} \quad (1)$$

Where i, j are the row and the column number of the grid map and the neighborhood (i, j) is the vicinities around the (i, j) cell. The neighboring can be in 4 or 8 directions around the (i, j) cell. The Algorithm, then, will calculate the value of each cell by adding +1 to it when there is no obstacle there (according to the least value of neighboring cells) and assign it to the cell. In each stage only the cell which hasn't any value will get a value and those cells got a value in previous stage will not change. The process terminates when reaching the start point. After finishing the wave expansion, the robot initiates its movement toward the target point. The robot chooses from its 8 surrounding directions the cell with least value and moves toward that cell until it has reached the target point.

A simple illustration of wavefront planning is shown in Fig.2 using eightfold neighboring (i.e robot has the ability of moving in 8 directions). In this example, the number of transitions from the start to the goal are shown in the grid. The robot could simply follow the decreasing values in order to reach the goal, and could do so along a number of different paths in this example.

In case of adding one to the value of all neighbor cells (as shown in fig.2), there is the selection problem for the shortest path because of the same value in neighborhood. This problem can be solved in [21] by assigning the different value according to the cell location. The values of orthogonal and diagonal neighbor cells are spread from the goal cell to the start cell, with adding three and four, respectively. Then, there are no cells having the same value in the planned path. Fig. 3 displays an example of this modified wave expansion planning method.

**Fig. 3.** Modified Wave Front Planning**Fig. 4.** Focused Wave Front Planning

As shown from the figures 2 and 3, these methods require that every cell in the map be filled by values to determine the path. This works well when there are a small number of cells. However, when planning a path using a regular grid map over a large area this becomes very time expensive.

Our main idea is, during expansion only focus on the region which provides navigations towards the source in early stage without full expansions. Therefore, it only updates the values of all cells which are nearer to the source so that it can reach the source very quickly, and the expansion process terminates. Our idea is simple and time efficient. For every cell ‘c’ encountered during the wave expansion, it maintains two values ‘weight’ and ‘cost’:

Weight: this value is assigned according to the cell location. The values of orthogonal and diagonal neighbor cells are spread from the goal cell to the start cell, with adding three and four, respectively as proposed by the author [21].

Cost: this is the cost of getting from cell ‘c’ to start point. Here cost is the Euclidean distance of two 2D points. The working steps are as follows:

1. Initialize target with weight zero and cost with $d(T,S)$, d is Euclidean cost.
2. While source not reached do,
3. Select a cell ‘c’ with minimum cost
4. Find those neighbors of ‘c’ that hasn’t got any value in previous stages.
5. Assign them weight and cost values accordingly.

Hence this algorithm only focuses on the cells from where the distance of the source is minimized. Once this is complete, we can simply follow the first value in reverse order until to reach the goal, avoiding any cell that is a obstacle. The process is illustrated in fig.4 (upper value in the cell represents the weight and lower is the cost). However, the path planned by this algorithm is not most optimal but time efficient.

4 Experiments

The aim of this part is to compare the capability of the waveform (WF) expansion proposed by [21] for rectangular map and the focused waveform (FWF) expansion to

plan a valid path. To do this, a Java based self-developed simulator is built. The workspace was taken as an input in form of an image. The image depicted the obstacles as black regions and the accessible area as the white region. Fig. 5 shows four different workspace on which the experiments have been carried out. The size of all the maps is 100X100. The robot was supposed to move from the top left corner to the bottom right corner. The two algorithms are run on all four maps. Through experiments we can find out the influence of the cost function on the planned path. Performance of each of the algorithm is measured in terms of (a) number of movement steps; (b) path length; (c) number of turns; (d) number of cells processed during wave expansion; and (e) time. One movement step is the movement from one cell to one of its neighbor cells. Processed cells are the cells that are filled with values during wave expansion.

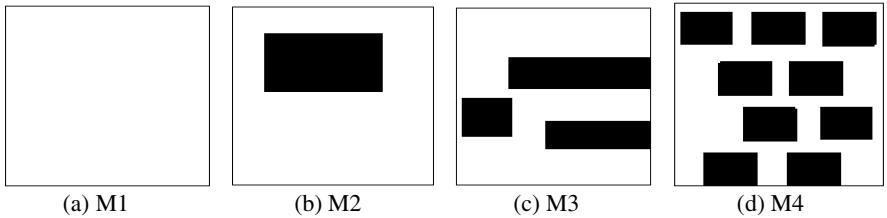


Fig. 5. Simulation environments

4.1 Case I: Obstacle Free Environment

In this case experiments are performed on map M1, which is an obstacle free environment. Both WF and FWF algorithm was made to run on a completely blank map. We observed that both algorithms traced the path from the source to the destination following a straight line path. This was the possible optimal path. Figure 6 shows the snapshots of the planned path. Table 1 presents the computed results. The analysis shows that the behavior of both algorithm is same for the first three performance metric. However time and number of processed cells are much reduced.

4.2 Case II: Environment with Obstacles

In this case experiments are performed on map M2, M3, and M4. Figure 7, 8 and 9 shows the planned trajectories. Computed results are presented in table 2, 3 and 4 respectively. The results presented in table 2, 3, and 4 shows that, the time and the number of processed cells in FWF algorithm is 88.4% and 93% less respectively on average basis compared to WF algorithm. The reason is that FWF algorithm is focuses only on the desired region instead of the entire workspace. However, the number of turns is 34.1% more. The reason is that the search is performed only in the focused region that's why the movement choice is limited and turns increases. Although, number of movement steps and path length is same in all the maps.

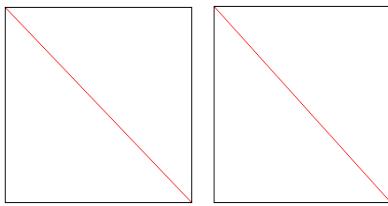


Fig. 6. Planned path using WF (left) and FWF (right) in map M1

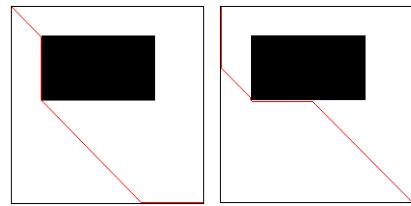


Fig. 7. Planned path using WF (left) and FWF (right) in map M2

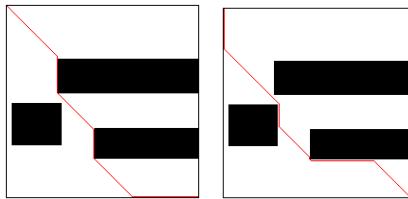


Fig. 8. Planned path using WF (left) and FWF (right) in map M3

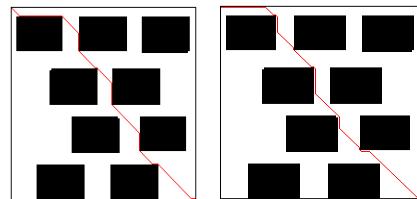


Fig. 9. Planned path using WF (left) and FWF (right) in map M4

Table 1. Computed Results of Map M1

Performance Metric	WF	FWF	Analysis (FWF vs. WF)
# Movement Steps	100	100	same
Path Length	140.01	140.01	same
# Turns	1	1	same
Time (in msec.)	1592	31	98% less
# Processed Cells (out of 10000)	10000	494	95.1% less

Table 2. Computed Results of Map M2

Performance Metric	WF	FWF	Analysis (FWF vs. WF)
# Movement Steps	132	132	same
Path Length	158.75	158.75	same
# Turns	4	7	42.8% more
Time (in msec.)	640	47	92.7% less
# Processed Cells (out of 8053)	8053	459	94.3% less

Table 3. Computed Results of Map M3

Performance Metric	WF	FWF	Analysis (WF vs. FWF)
# Movement Steps	134	134	same
Path Length	159.92	159.92	same
# Turns	6	9	33.3% more
Time (in msec.)	327	47	85.6% less
# Processed Cells (out of 7250)	6076	454	92.5% less

Table 4. Computed Results of Map M4

Performance Metric	WF	FWF	Analysis (WF vs. FWF)
# Movement Steps	129	129	same
Path Length	157	157	same
# Turns	14	19	26.3% more
Time (in msec.)	359	47	86.9% less
# Processed Cells (out of 5723)	5723	448	92.2% less

Table 5. Results' Summary (FWF vs. WF)

Performance Metric	Conclusion (on average basis)
Time	88.4% less
# Processed Cells	93% less
# Turns	34.1% more
# Movement Steps & Path Length	same

5 Conclusion and Future Work

In this paper, we showed that, in simple wave front algorithm, the addition of one to the value of all neighbor cells, creates the selection problem for the path because of the same value in neighborhood. This problem is resolved by assigning the different value according to the cell location. These methods continuously find the proper path but it requires that every cell in the map be filled by values to determine the path in almost cases, and this makes the algorithm execution more sluggish in big scale environment. In response, we proposed a new extension, called the focused wave front expansion, in which we focus only on the region that leads towards the source point by the use of the cost function. As a result algorithm execution in the big scale environments increases. However the planned path is not most optimal but time efficient. This algorithm can be applied in the applications where optimality sacrifices for fast operation. We execute and compare that algorithms. Table 5 summarizes the analysis that have been done. Further works will concern a deeper study on the properties of algorithm.

References

1. Liu, G., et al.: The Ant Algorithm for Solving Robot Path Planning Problem. In: Third International Conference on Information Technology and Applications (ICITA), pp. 25–27 (2005)
2. Frazzoli, E., et al.: Real-Time Motion Planning for Agile Autonomous Vehicles. Journal of Guidance, Control and Dynamics 25(1) (2002)
3. Chen, C., Han-Xiong, H., Dong, D.: Hybrid Control for Robot Navigation. IEEE Robotics & Automation Magazine, 37–47 (2008)
4. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Boston (1991)

5. Nooralie, A., Mostafa, S.A.: Robot path planning using wavefront approach WEFO. In: ICACTE 2009, Cairo, Egypt, vol. 1, pp. 963–972 (2009)
6. Amir Nooralie, R.: iraji: Robot path planning using wavefront approach wall-following. In: ICCSIT 2009, Beijing, China, vol. 1, pp. 417–420 (2009)
7. Nooralie, A., Nooralie, H.: Robot path planning using wavefront approach virtual wave hill. In: ICACTE 2009, Cairo, Egypt, vol. 1, pp. 695–704 (2009)
8. Lumelsky, V.J., Skewis, T.: A paradigm for incorporating vision in the robot navigation function. In: Proc., IEEE Int. Conf. on Robotic Automation, Institute of Electrical and Electronics Engineers, New York, pp. 734–739 (1988)
9. Lumelsky, V.J., Stepanov, A.A.: Path-planning strategies for a point mobile automation moving amidst unknown obstacles of arbitrary shape. *Algorithmica* 2(4), 403–430 (1987)
10. Pradhan, S.K., Parhi, D.R., Panda, A.K.: Navigation of Multiple Mobile Robots using Rule-based-Neuro-Fuzzy Technique. *International Journal of Computational Intelligence* 3(2), 142–152 (2007)
11. Lee, S., Adams, T.M., Byoo, B.: A fuzzy navigation system for mobile construction robot. *Autom. Constr.* 6(2), 97–107 (1997)
12. Lee, S.: Spatial model and decentralized path planning for construction automation. PhD thesis, University of Wisconsin–Madison, Madison, Wis (2000)
13. Kamon, I., Rivlin, E.: Sensory-based motion planning with global proofs. *IEEE Trans. Rob. Autom.* 13(6), 812–814 (1997)
14. kala, R., shukla, A., Tiwari, R.: Fusion of probabilistic A* algorithm and fuzzy inference system for robotic path planning. *Artificial Intelligence Review* 33, 307–327 (2010)
15. Manikas, T.W., Ashenayi, K., Wainwright, R.L.: Genetic Algorithms for Autonomous Robot Navigation. *IEEE Instrumentation & Measurement Magazine* (2007)
16. Du, X., Chen, H.-h., Gu, W.-k.: Neural network and genetic algorithm based global path planning in a static environment. *Journal of Zhejiang University SCIENCE* 6, 549–554 (2005)
17. Behnke, S.: Local Multiresolution Path Planning. Preliminary version. in Proc. of 7th RoboCup Int. Symposium, Padua, Italy, pp. 332–343 (2003)
18. Lebedev, D.V., Steil, J.J., Ritter, H.J.: The dynamic wave expansion neural network model for robot motion planning in time-varying environments. Elsevier Science Ltd 18, 267–285 (2005)
19. Nooralie, A., Mostafa, S.A.: Robot path planning using wave expansion approach Virtual target. In: ICCTD 2009, Kinabalu, Malaysia (2009)
20. Nooralie, A., Nooralie, H.: Path planning using wave front's improvement methods. In: International Conference on Computer Technology and Development (2009)
21. Oh, J.S., et al.: Complete Coverage Navigation of Cleaning Robots using Triangular Cell Based Map. *IEEE Transactions on Industrial Electronics* 51(3) (2004)

Evolving Temporal Fuzzy Association Rules from Quantitative Data with a Multi-Objective Evolutionary Algorithm

Stephen G. Matthews, Mario A. Gongora, and Adrian A. Hopgood

Centre for Computational Intelligence,
De Montfort University, Leicester, UK
{sgm,mgongora,aah}@dmu.ac.uk
<http://www.cci.dmu.ac.uk/>

Abstract. A novel method for mining association rules that are both quantitative and temporal using a multi-objective evolutionary algorithm is presented. This method successfully identifies numerous temporal association rules that occur more frequently in areas of a dataset with specific quantitative values represented with fuzzy sets. The novelty of this research lies in exploring the composition of quantitative and temporal fuzzy association rules and the approach of using a hybridisation of a multi-objective evolutionary algorithm with fuzzy sets. Results show the ability of a multi-objective evolutionary algorithm (NSGA-II) to evolve multiple target itemsets that have been augmented into synthetic datasets.

Keywords: multi-objective evolutionary algorithm, fuzzy association rules, temporal association rules, NSGA-II, hyrbid.

1 Introduction

Association rule mining is a well established method of data mining that identifies significant correlations between Boolean items in transactional data [1]. This paper extends the classical problem by exploring the composition of two variants of association rule mining with a hyrbid approach.

It is often assumed in classical association rule mining that the dataset is static, meaning that discovered rules hold across the entire period of the dataset. However, real-world datasets can have underlying temporal patterns. For example, an increase in association rule frequency may occur before a large sports event or when an unforeseen events occur, such as hurricanes (e.g., [2]). Quantitative association rule mining [3] discovers rules that express associations between intervals of item attributes (e.g. height, pressure), but common approaches of discretisation can lead to a loss of information. Evolutionary Computing (EC) has been used to remove the requirement for prior discretisation and the synergy of hybridising EC with fuzzy sets has become popular for data mining tasks [4,5] such as classification and association rule mining.

The composition of temporal association rule mining and quantitative association rule mining is treated as a multi-objective optimisation problem. The aim is to extract temporal association rules from quantitative data using fuzzy sets. The temporal association rules sought are those that occur more frequently over an interval of the dataset, which are seen as an area of greater density. The advantages of fuzzy sets are they allow a linguistic interpretation, a smoother transition between boundaries, and better handle uncertainty. The itemset/association rule space, temporal space and quantitative space are simultaneously searched and optimised. This paper extends our previous work in [6] by including a quantitative element, mining multiple occurrences of association rules and by directly mining association rules.

This paper is organised as follows: Section 2 presents an overview of related works on association rule mining; Section 3 describes the multi-objective evolutionary algorithm for mining temporal fuzzy association rules from quantitative data; Section 4 presents results; and conclusions are drawn in Section 5.

2 Quantitative and Temporal Association Rule Mining

A disadvantage of classical quantitative association rule mining is the crisp boundaries of discretised values that potentially hide rules and lose information [8]. Soft computing techniques can overcome this issue, for example, in [8], a genetic algorithm evolves attribute intervals for a fixed number of attributes. Fuzzy association rules deal with inaccuracies in physical measurements and better handle unnatural boundaries found in crisp partitions. They provide a linguistic interpretation of numerical values for interfacing with experts. Evolving fuzzy association rules [9] enhances the interpretability of quantitative association rules.

There are two common approaches to mining quantitative association rules. One approach is to tune membership functions and use a deterministic method to induce rules afterwards (e.g., [10]). Membership functions are tuned to produce maximum support for 1-itemsets before exhaustively mining rules. Another approach is to extract association rules whilst defining attribute intervals [8] or membership functions [9]. The latter approach is adopted in this paper.

A key issue of classical methods, based on the support-confidence framework, is that temporal patterns with low support can escape below the minimum support threshold. For example, supermarket items may be sold only during particular seasonal periods, resulting in annual support values dropping below a minimum threshold, despite having sufficient support values in a seasonal period. The *lifespan* property [11] is an extension on the Apriori algorithm [19] that incorporates temporal information. This measure of support is relative to the lifespan of the itemset defined by a time interval, known as temporal support, corresponding to the first and last occurrences of the itemset. But this does not consider datasets where the frequency of rules may be skewed towards particular areas whilst still occurring throughout the entire dataset.

A step towards analysing areas of a dataset where rules occur more frequently is cyclic association rule mining [12]. Cyclic rules are induced from user-defined

partitions of regular periods and pattern matching is performed on binary sequences. Other temporal patterns that can potentially be extracted with our method are partially periodic rules [13] and calendar-based schemas [14].

Our previous work [6] has demonstrated mining association rules that occur more frequently over single areas of a dataset with a single objective genetic algorithm. A multi-objective evolutionary algorithm (MOEA) is used in [7] and extended here to include association rules and multiple targets.

3 Multi-Objective Evolutionary Search and Optimisation

Extracting a set of fuzzy association rules from areas of the dataset where the occurrence is greater is treated as a multi-objective problem. This is the optimisation of two or more functions, whilst satisfying optional constraints [15]. Optimal solutions found with a MOEA are compromises between objectives and such trade-offs are often managed with the concept of Pareto optimality. A solution is said to be Pareto optimal when no change in the solution will improve one objective without degrading another objective.

A Pareto based MOEA is capable of producing multiple association rules from a single run through utilising a maintained set of maximally-spread Pareto-optimal solutions. This is desirable when the cardinality of the optimal set may be more than one, for instance in the case of multiple temporal patterns. This improves our previous work [7] which mines single temporal patterns. From the plethora of MOEAs, we selected NSGA-II [16] for its popularity and ability to maintain a diverse set of solutions suitable for extracting multiple patterns. Previous works have used NSGA-II for Subgroup Discovery [17], a closely related area, and motif sequence discovery [18], a different form of temporal mining.

3.1 Representation

A Michigan approach and mixed coding scheme is used to represent the temporal interval and fuzzy association rules as

$$C = (t_0, t_1, i_0, a_0, b_0, c_0, A_0, \dots, i_k, a_k, b_k, c_k, A_k) \quad (1)$$

where the temporal interval is defined with t_0 and t_1 as integers. The items are integers denoted with i and the basic parameters of the triangular membership functions are real numbers indicated with a , b and c for association rules with k distinct items. A binary value in A_k determines if this item belongs to the antecedent or consequent.

3.2 Objectives

Temporal Support: The temporal support objective, ts , guides the MOEA to find itemsets that occur more frequently in areas of the dataset. Modified from [11], this is redefined as a minimisation function

$$ts(X, l_X) = 1 - \frac{\sigma(X)}{l_X} \quad (2)$$

with l denoting a time interval i.e. $l_X = t_1 - t_0$ where t_0 is the lower endpoint, t_1 is the upper endpoint and $\sigma(X)$ is the itemset support. A minimum temporal support is used to prevent solutions evolving towards the smallest time interval of length 1, which would produce a 100% temporal support.

Temporal Rule Confidence: Temporal confidence, tc , helps extract association rules from itemsets. This aims to identify specific association rules that have a temporal occurrence based on temporal support.

$$tc(X \Rightarrow Y, l_{X \cup Y}) = \frac{ts(X \cup Y, l_{X \cup Y})}{ts(X, l_X)} \quad (3)$$

Fuzzy Rule Support: This objective optimises the membership function parameters of matching association rules. The quantitative values are modelled with triangular fuzzy sets and the objective's optimal solution is one where the fuzzy sets support the quantitative values associated with the association rule to the highest degree of membership. Fuzzy rule support, fs , is the sum of the degrees of memberships, $sum(\mu(x^{(i)}))$, for a chromosome itemset, $x^{(i)}$, in the i th transaction.

$$fs = (k \cdot (t_1 - t_0)) - \sum_{i=t_0}^{t_1} sum(\mu(x^{(i)})) \quad (4)$$

$$sum(\mu(x^{(i)})) = \sum_{j=0}^k \begin{cases} \mu(x_j^{(i)}), & \text{dataset item matches gene item} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$\mu(x_j^{(i)}) = \begin{cases} \frac{x_j^{(i)} - a}{b - a}, & \text{if } a \leq x_j^{(i)} < b \\ \frac{c - x_j^{(i)}}{c - b}, & \text{if } b \leq x_j^{(i)} \leq c \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Equation 4 subtracts the sum of the actual degrees of memberships from the maximum possible sum if all items in a transaction match those in the chromosome. Equation 5 performs the summation of actual degrees of memberships for chromosome items matching dataset transaction items. Equation 6 is the triangular membership function.

Membership Function Widths: The aim of this objective is to prevent the membership function parameters evolving to cover the entire range of values i.e. the feet of the membership function (a and c) nearing the limits of the attribute values. Without this objective solutions evolve to cover the entire range of attribute values because this yields higher support values as it includes more items.

$$mf_widths = \begin{cases} \sum_{j=0}^k c_j - a_j, & \text{if } c_j - a_j > 0 \\ nitems, & \text{otherwise} \end{cases} \quad (7)$$

3.3 Initialisation and Genetic Operators

The initial population is randomly generated with lower and upper endpoints being within proximity to the first and last transactions. An endpoint range is defined for two purposes: limit the range for creating endpoints and also for mutating endpoints. Time endpoints initialised near dataset boundaries provide starting solutions with large temporal coverages of the dataset. Without the endpoint range, random sampling of time intervals occurs. This may lead to some potentially strong itemsets being lost, so an initial large temporal coverage, combined with the mutation operator, provides more opportunity for solutions with great potential that initially may be weak.

Crossover is adapted to handle quantitative data from the method proposed in [6]. For mutating genes that form the time interval endpoints, the values are generated within the endpoint range (epr) where the midpoint is the value of the current gene (g), such that the mutated value is a member of the set $\{-epr/2, \dots, g, \dots, epr/2\}$. This reduces the effect of randomly sampling the dataset. The endpoint range is decremented every generation until reaching 10, to allow further mutations.

4 Experimental Study

4.1 Methodology

The IBM Quest Synthetic Data Generator [19] has been extended to include quantitative attributes. The dataset has these features: 1000 transactions, 50 items, an average transaction size of 10 and a maximum size for quantitative values of 20. The Apriori algorithm is used to identify relatively low (0.2%), medium (1.7%) and high support (3.5%) association rules that are augmented into areas of the dataset to produce temporal patterns. This is based on the process defined in [6] that creates target temporal patterns with varying levels of difficulty and is extended to include multiple temporal patterns. The minimum temporal support was set to 30. Figure 1 depicts the frequency of a quantitative itemset augmented into the first half and second half of a dataset to demonstrate

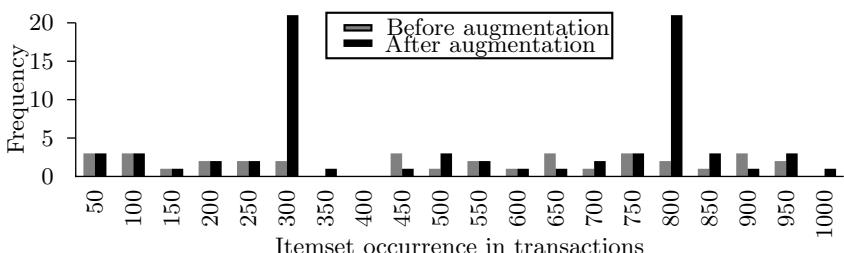


Fig. 1. Histogram of itemset {8, 12, 21, 45} with high support (3.5%) (Bins 250 and 750 have one extra itemset that does not contain the quantitative values)

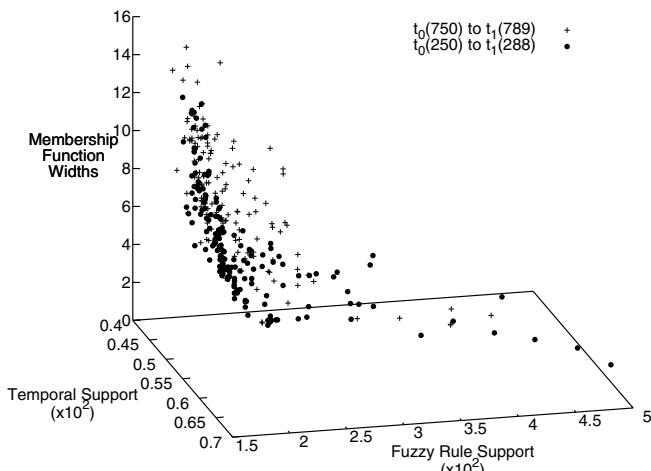
Table 1. Results of augmenting same quantitative itemset, or temporal patterns (TP), in two locations

Endpoint	t_0	t_1	Itemset	Aug. Sup.	TP identified	Qty. of TP identified	
						1	2
250	289	24	(3) 31 (7) 32 (12) 38 (16)	0.2%	8		
750	788	24	(3) 31 (7) 32 (12) 38 (16)	0.2%	1	7	1
250	289	12	(3) 31 (7) 41 (12) 48 (16)	1.7%	19		
750	788	12	(3) 31 (7) 41 (12) 48 (16)	1.7%	12	15	8
250	289	8	(3) 12 (7) 21 (12) 45 (16)	3.5%	47		
750	788	8	(3) 12 (7) 21 (12) 45 (16)	3.5%	41	12	38

the increased occurrence of the same pattern in two areas. Table 1 shows the itemsets used for augmentation. Augmentation is based on itemset support because this is used to extract fuzzy association rules.

4.2 Results

The augmented itemsets were identified with 50 runs of NSGA-II on each dataset, although with varying success for different levels of difficulty. The results are summarised in Table 1. The itemsets were deemed to be successfully identified if the entire itemset matched that of the augmented itemset and it was in proximity of the endpoints, t_0 and t_1 . The number of temporal patterns identified increases with the support level of the augmented itemset. For each level of difficulty there is one area of the dataset that is more likely to be identified as a temporal pattern. For example, the high support (3.5%) dataset identified the 1st temporal pattern (transactions 250–289) in 12 runs while identifying the 2nd (transactions

**Fig. 2.** Three objectives for best solutions in a portion of a final population augmented with a high support (3.5%) itemset

750–788) in 38 runs. Also, with a higher support value of augmented itemsets there is an increase in identifying both temporal patterns. The correct identification of the quantitative attributes with fuzzy sets varies greatly and not all attributes were correctly identified in a solution.

Three of the objectives are plotted in Figure 2, both augmented itemsets in the final solution are distinguished here. This graph can be used to view the trade-offs between fuzzy association rules, which is of particular use for knowledge discovery in real-world applications. This figure demonstrates how the objectives conflict, particularly for membership function widths and fuzzy rule support.

5 Conclusions

We have used a hybrid approach of a MOEA (NSGA-II) and fuzzy sets to evolve multiple temporal association rules from quantitative transaction data. This demonstrates the ability to find association rules that occur more frequently in numerous areas of a dataset. A MOEA maintains diversity and so allows for numerous temporal patterns to evolve. The advantages of the proposed approach is that it does not exhaustively search the various spaces, it requires no discretisation and yields numerous diverse association rules.

Future work will explore enhancing the robustness of identifying quantitative attributes and evolving low support itemsets. Real-world datasets will be used as these are crucial to demonstrating the impact of this research. We will compare statistical methods, such as temporal based Apriori methods, and other MOEAs with this approach to explore its suitability.

Acknowledgements. Supported by an Engineering and Physical Sciences Research Council Doctoral Training Account.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD ICDM, Washington, DC, USA, pp. 207–216 (1993)
2. Leonard, D.: After Katrina: Crisis Management, the Only Lifeline Was the Wal-Mart. *FORTUNE Magazine* (2005)
3. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In: ACM SIGMOD ICDM, Montreal, Quebec, Canada, pp. 1–12 (1996)
4. Ishibuchi, H.: Multiobjective Genetic Fuzzy Systems: Review and Future Research Directions Fuzzy Systems Conference. In: FUZZ-IEEE, London, UK, pp. 1–6 (2007)
5. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. *Information Sciences* 180(14), 2633–2634 (2010)
6. Matthews, S.G., Gongora, M.A., Hopgood, A.A.: Evolving Temporal Association Rules with Genetic Algorithms. In: Bramer, M., Petridis, M., Hopgood, A. (eds.) *Research and Development in Intelligent Systems XXVII*, pp. 107–120. Springer, London (2010)

7. Matthews, S.G., Gongora, M.A., Hopgood, A.A.: Evolving Temporal Fuzzy Itemsets from Quantitative Data with a Multi-Objective Evolutionary Algorithm. In: IEEE SSCI, Paris, France, (accepted for publication 2011)
8. Mata, J., Alvarez, J.L., Riquelme, J.C.: An evolutionary algorithm to discover numeric association rules. In: ACM SAC, New York, NY, USA, pp. 590–594 (2002)
9. Kaya, M.: Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules. *Soft Computing - A Fusion of Foundations, Methodologies and Applications* 10(7), 578–586 (2006)
10. Hong, T.-P., Chen, C.-H., Lee, Y.-C., Wu, Y.-L.: Genetic-Fuzzy Data Mining With Divide-and-Conquer Strategy. *IEEE Transactions on Evolutionary Computation* 12(2), 252–265 (2008)
11. Ale, J.M., Rossi, G.H.: An approach to discovering temporal association rules. In: ACM SAC, Como, Italy, pp. 294–300 (2000)
12. Özden, B., Ramaswamy, S., Silberschatz, A.: Cyclic Association Rules. In: ICDE, Washington, DC, USA, pp. 412–421 (1998)
13. Han, J., Gong, W., Yin, Y.: Mining segment-wise periodic patterns in time-related databases. In: KDD, New York, NY, USA, pp. 214–218 (1998)
14. Li, Y., Ning, P., Wang, X.S., Jajodia, S.: Discovering calendar-based temporal association rules. *Data & Knowledge Engineering* 44(2), 193–218 (2003)
15. Coello, C.A.C., Lamont, G.B., van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, Heidelberg (2007)
16. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
17. Carmona, C., Gonzalez, P., del Jesus, M., Herrera, F.: NMEEF-SD: Non-dominated Multiobjective Evolutionary Algorithm for Extracting Fuzzy Rules in Subgroup Discovery. *IEEE Transactions on Fuzzy Systems* 18(5), 958–970 (2010)
18. Kaya, M.: MOGAMOD: Multi-objective genetic algorithm for motif discovery. *Expert Systems with Applications* 36(2, Part 1), 1039–1047 (2009)
19. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB, Santiago, Chile, pp. 487–499 (1994)

Stereovision-Based Obstacle Avoidance Procedure for Autonomous Mobile Platforms

Maciej Polańczyk, Agnieszka Owczarek, Michał Strzelecki, and Krzysztof Ślot

Institute of Electronics, Technical University of Łódź, Wolczanska Street 211/215,
90-924 Łódź, Poland
maciej.polanczyk@gmail.com,
{agnieszka.owczarek,mstrzel,kslot}@p.lodz.pl

Abstract. The paper presents a procedure for collision-free guidance of autonomous mobile platforms through unknown environments. Stereovision images are used to determine and keep track of an obstacle map, which provides a basis for path-planning. The main problem that has to be solved in obstacle map derivation is elimination of artifacts resulting from depth estimation. We propose a two-step artifact filtering procedure, which exploits both within-frame spatial correlations as well as temporal, between-frame correlations to do this task. Experiment results prove efficiency of the adopted approach for platform operation in real environments, where both static and moving obstacles are present.

Keywords: mobile platform motion system, camera motion estimation, stereovision, path planning, map of obstacles.

1 Introduction

The capability of collision-free navigation in an unknown environment is a basic requirement that majority of the autonomous robotic mobile platforms have to satisfy. In many systems, e.g. robots aimed for environment exploration, there is no way to provide up-to-date detailed obstacle map beforehand. In such systems the ‘missing’ knowledge has to be discovered by the system in real time. To cope with problems of mobile platform operation in dynamically changing and partly-known environments, the concept of the Simultaneous Localization And Mapping (SLAM) was introduced.

SLAM methods build an up-to-date map while robotic devices estimate their own position. While moving, the map is updated every time the new obstacles are encountered. Not only new obstacles are marked on the map, but also the old ones’ positions are updated according to new platform localization. This has to be done in order to preserve spatial relationships between objects. To succeed it is vital to ensure the real time computational complexity together with localization reliability.

In this paper, a procedure for collision avoidance, aimed for autonomous mobile platforms equipped with stereovision camera, is presented. In such systems obstacle localization can be determined from depth image. Unfortunately, errors introduced during disparity calculation can significantly impair correct identification and localization of obstacles. Therefore there is a need to improve process of building map of

the obstacles by eliminating artifacts introduced during stereovision depth image formation. To cope with this problem a novel approach, based on spatio-temporal filtering, is proposed. In motion control module a fuzzy logic algorithm was used [1,2], resulting in an overall hybrid, fuzzy-crisp computational architecture of the proposed system. The experimental results show that the resulting method is able to reliably, in real time, navigate in unknown indoor environments such as corridors or office rooms.

2 Related Work

Several approaches that deal with the problem of building up-to-date obstacle maps can be found in literature. Most of the existing solutions use a laser rangefinder, augmented with monocular/stereo vision [3,4,5] or sonar sensors [6], as a source of information on the environment. Such complex systems are characterized by high precision and reliability but are costly. Therefore, many researches focus on stereovision-only based systems that are more error-prone but much cheaper [7].

In majority of stereovision-based systems imprecision of disparity calculation is compensated by fusion of utilizing a single camera image and depth image. For example in [8] the depth discontinuities in U-V-disparity domain is used to confirm the existence of obstacles. Other researchers have developed sub-pixel displacement methods [9] to enhance the accuracy of disparity.

The proposed approach for constructing reliable obstacle maps exploits only depth information. Disparity derivation errors are corrected by both spatial and temporal filtering. We use original depth information, with its insufficient precision to get a general environment perception, whereas appropriately filtered information, without artifacts, is used for path-planning. The proposed platform-guidance system provides reliable mapping of an environment, correct short-term estimation of the platform's current position and real-time operation.

3 Proposed Solution

The proposed algorithm consists of 3 stages – (I) camera motion estimation, (II) building map of obstacles and (III) path planning - as presented on the flowchart in Fig. 1. Detailed description of camera motion estimation algorithm, which is used in the proposed procedure, can be found in previous authors' work [10]. Path planning is based on the well-known Lee's algorithm [11].

3.1 Preliminary Obstacles Detection

Preliminary obstacle detection is performed on the basis of stereovision depth image (Fig.1d). In parallel with camera motion estimation (Fig.1e), ambient obstacles are searched for. First, the points that are located at distances larger than Z_{max} (10m) from the camera are discarded. This way certainty of point localization correctness in subsequent steps increases (Disparity for closer points can be estimated more accurately).

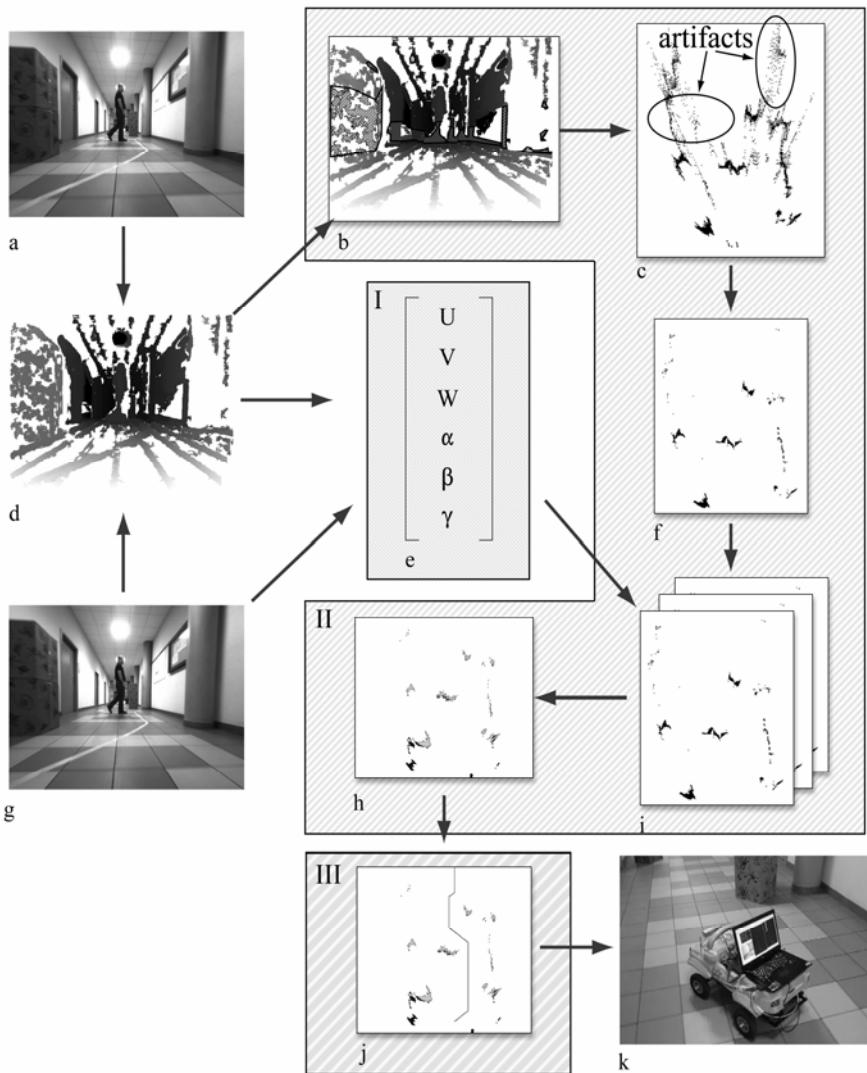


Fig. 1. The flowchart of the proposed algorithm. (a) Right image. (b) Disparity image with marked potential obstacles. (c) Potential obstacles' points transformed to 2D map. (d) Depth image. (e) Motion vector. (f) Preliminary map of obstacles (c) after erosion. (g) Left image. (h) Final map of obstacles. (i) Time filtering. (j) Path. (k) Control mobile platform. (I) Camera motion vector estimation. (II) Building map of obstacles. (III) Path planning.

Points which Z coordinate is in the range $0 \div H_{max}$ (H_{max} – platform's height, 0.8m) from the camera are considered as belonging to obstacles and are taken into account for further map construction (Fig.1b). Such assumption enables eliminating points lying on the surface on which mobile platform is moving and located on hanging

obstacles that have no impact on safety platform motion (note that to adopt this assumption the camera needs to be mounted in parallel with the ground).

Next, points that belong to the considered volume are transformed from camera coordinates to 2D map of obstacles (Fig.1c)). The transformation is done according to the formula (1).

$$\begin{bmatrix} Xm \\ Zm \end{bmatrix} = \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} Z \\ Zc \end{bmatrix} + \begin{bmatrix} Xc \\ Zc \end{bmatrix}. \quad (1)$$

where: Xm, Zm – point coordinates in map, Xc, Zc - point coordinates in real world, X, Z - camera current position in map, β – direction of current camera movement.

3.2 Building Reliable Up-to-Date Map of Obstacles

Preliminary map of obstacles created in previous step contains artifacts introduced by errors in disparity calculation (Fig.1c). These artifacts have to be filtered out as cause misjudgment what is an obstacle. Therefore before points preliminarily classified as belonging to an obstacle are added to the global map of obstacles they undergo spatio-temporal filtration (Fig. 1f, 1c). They are stored in temporary buffer and:

- the 3x3 erosion with cross-shaped structuring element is applied to eliminate single points (spatial filtering). The probability that only one point was detected on the obstacle is considered as very low thus such a point can be neglected (Fig. 1f).
- only points that are valid in three subsequent frames (points from each frame are stored in separate temporary buffers (Fig. 1i)) are left (temporal filtering) and added to global obstacle map (Fig. 1h).

While the camera is moving, a new region is explored and new obstacles can appear or some can change their position. Thus there is a continuous need to update the global map. Points from global obstacle map that are located within the camera visual ‘proximity’ (defined as the 10m-long rectangular area in front of the camera restricted by a 100-degree camera viewing angle) are taken into further consideration, as described in [12]. Firstly, it is checked if the analyzed point is present in all three temporary buffers. If it is, no update is needed. If not, the label “potentially to remove” is assigned with such a point and a follow-up procedure is done. In subsequent steps it is verified whether this point is missing from the temporary buffer because the obstacle is not valid anymore and should be removed or if it was occluded by another obstacle. In order to do this the disparities are compared. If the disparity is:

- Larger than the one resultant from the distance means that the obstacle was occluded by another one and should not be removed from global map (closer object has larger disparity).
- Otherwise the object changed its position and has to be removed from previous location on the map (some farther object is now within the camera viewing range).

Disparity calculation based on the data from global map is not a trivial task as the map is two-dimensional. Only information about X and Z coordinates is stored. To cope with this problem, disparities for all points in the range of Y : $0.0m \div H_{max}$ are computed and compared.

Finally, temporary buffers are updated by new data so that data associated with the latest three images were stored and analyzed. Such updated global map of obstacles is an entry point for the path planning algorithm.

4 Experimental Results

To validate the proposed solution the Bumblebee2 stereovision camera was used [13]. This camera was mounted on a test mobile platform (Fig. 1k) and several sequences of images, for indoor environment, were captured.

In order to increase computational efficiency, the proposed algorithm is implemented in four independent parallel threads. The first thread is responsible for capturing stereovision images and calculating disparity. In the second one the camera motion vector is estimated. Obstacles map derivation and path planning is performed by third thread. The fourth thread controls mobile platform motion.

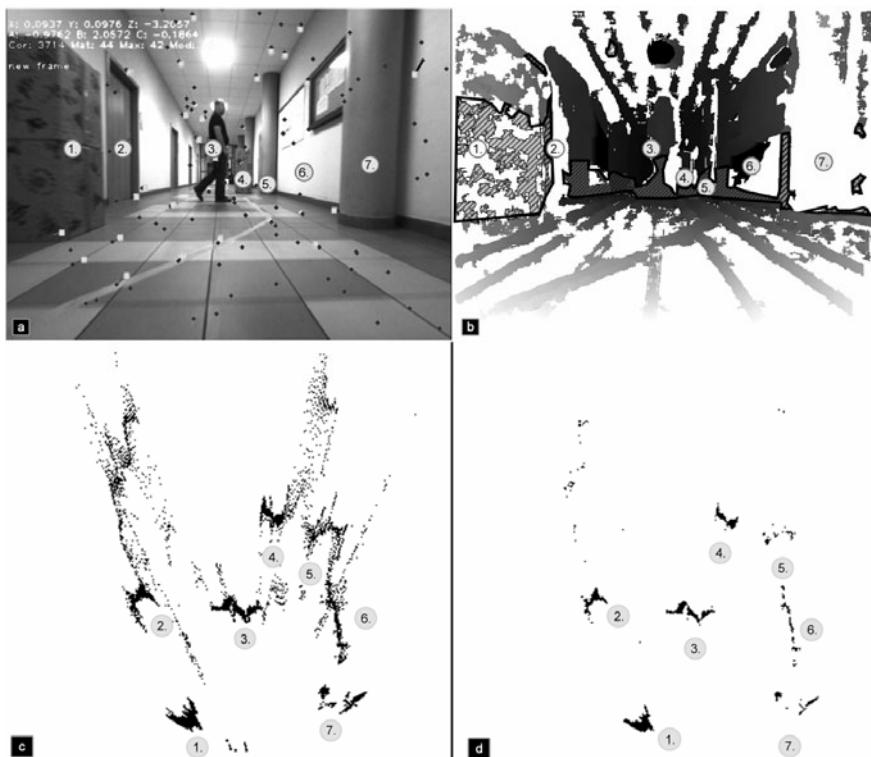


Fig. 2. The experimental results of proposed solution. (a) The frame captured from stereovision camera. (b) Disparity image; points being analyzed are marked with slanting lines; the darker color the further localized point; point for which disparity could not be calculated are marked white. (c) Potential obstacles points transformed to 2D map. Artifacts are visible. (d) Map of obstacles after spatial-time filtering.

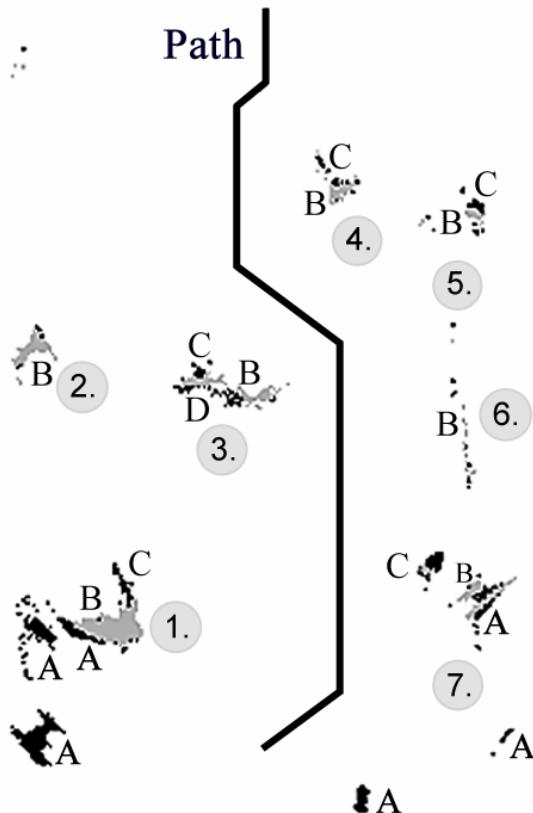


Fig. 3. Global map of obstacles with found path, A – obstacles outside the camera vision range, B – obstacles within camera vision range, C – obstacles hidden by other objects, D – obstacles to remove (not valid any more).

Fig. 2 presents the experimental results. In Fig. 2a the right image captured by stereovision camera is shown. Fig. 2b presents the stereovision disparity image being the base of all calculations - the darker color the farther localized point. Points for which disparity could not be calculated are marked in white. Regions marked with slanted lines correspond to potential obstacles. On Fig. 2c the preliminary map of obstacles is shown. Not only obstacle points (labeled 1-7) but also artifacts are visible. Most of the artifacts are removed after spatio-temporal filtering Fig. 2d. In Fig. 3 the global map of obstacles, built during the process of exploration, is presented. The thick dark line shows the optimal path. The A-D labels mean respectively: A – obstacles outside the camera vision range; B – obstacles within camera vision range; C – obstacles hidden by other objects; D – obstacles to remove (not valid any more).

Camera motion estimation error is about 5%. Such errors propagate and after long-term exploration a deviation between real camera position and estimated one can reach unacceptable level and results in wrong obstacles mapping. This is a common problem of majority of algorithms based on visual odometry, so there is a need of

periodical corrections of localization process, using some other input modality, such as e.g. GPS.

Also, as the process of detecting objects and building a map is based only on disparity image, homogenous obstacles can get missed. The possible solution to this problem is to consider additional information from some other type of sensors or introduce to the system some additional module of object detection based on raw image (not a depth-map) analysis.

In the Table 1 the average times of computation for every stage of the proposed algorithm are presented. All calculations were performed using 2.5GHz quad core computer. Resolution of images was 640x480. It shows that the path planning module is a weak point of the proposed solution. In order to cope with this problem some improvements were introduced, e.g. updating a path only in local areas. Also, one can consider entirely different, more computationally efficient path-planning algorithms, where sub-optimal path selection is sufficient. Comparing the performance with other solutions tested on similar equipment (the same camera, 2.4GHz processor), a significant improvement can be observed (average computational time for images with resolution 640x480 for proposed algorithm does not exceed 133ms whereas in [8] for images with resolution 320x240 it is 230ms). It results in more reliable system, being able to avoid obstacles in dynamically changing environment.

Table 1. Average times of computation

Stage	Time [ms]	Thread
Image acquisition and disparity calculation	60	1
Camera motion estimation	40	2
Obstacles detection	25	
Map building	3	
Map updating	5	3
Path planning	80 ¹	
Mobile platform control	<1	4

5 Conclusion

The presented algorithm correctly builds the obstacle map in majority of tested cases. Introduced spatio-temporal filtering of artifacts greatly improve quality of maps. The precision of obstacle localization allows to avoid collisions and reach the destination. Camera motion estimation is performed with ca. 17 frames/s and path planning with ca. 7 frames/s. The average delay between the path update is ca. 230ms.

Further research will focus on modifications of the path planning procedure, aimed at reducing its computational burden. Additionally, the way the map of obstacles is updated will be modified to prevent from removing of fast moving objects, which is the case for the current algorithm.

¹ Dependent on the distance to the destination and the size of cells (in this case 10m and 0.1x0.1m respectively).

References

1. Corchado, E., Abraham, A., de Carvalho, A.C.P.L.F.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)
2. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72(13–15), 2729–2730 (2009)
3. Castellanos, J.A., Neira, J., Tardos, J.D.: Multisensor fusion for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation* 17(6), 908–914 (2001)
4. Murarka, A., Modayil, J., Kuipers, B.: Building Local Safety Maps for a Wheelchair Robot using Vision and Lasers. In: 3rd Canadian Conference on Computer and Robot Vision, p. 25 (2006)
5. Labayrade, R., Royere, C., Gruyer, D., Aubert, D.: Cooperative Fusion for Multi-Obstacles Detection With Use of Stereovision and Laser Scanner. *Autonomous Robots* 19, 117–140 (2005)
6. Xue-Cheng, L., Cheong-Yeen, K., Shuzhi, S.G., Al Mamun, A.: Online map building for autonomous mobile robots by fusing laser and sonar data. *Mechatronics and Automation* 2, 993–998 (2005)
7. Agrawal, M., Konolige, K., Bolles, R.C.: Localization and Mapping for Autonomous Navigation in Outdoor Terrains: A Stereo Vision Approach. *Applications of Computer Vision* 7 (2007)
8. Bai, M., Zhuang, Y., Wang, W.: Stereovision based obstacle detection approach for mobile robot navigation. In: International Conference on Intelligent Control and Information Processing (ICICIP), pp. 328–333 (2010)
9. Young-Chul, L., Chung-Hee, L., Soon, K., Woo-Young, J.: Distance Estimation Algorithm for Both Long and Short Ranges Based on Stereo Vision System. In: Intelligent Vehicles Symposium, pp. 841–846 (2008)
10. Polańczyk, M., Barański, P., Strzelecki, M.: The application of Kalman filter in visual odometry for eliminating direction drift. In: International Conference on Signals and Electronic Systems, pp. 131–134 (2010)
11. Lee, C.Y.: An algorithm for path connection and its applications. *IRE Trans. on Electronic Computers* EC-10(3), 346–365 (1961)
12. Skulimowski, P., Strumiłło, P.: Surface detection in a stereo image sequence. *Image Process Techniques* (2006) (in polish)
13. BumbleBee2, <http://www.ptgrey.com/products/bumblebee2/index.asp>

Detecting Unknown Attacks in Wireless Sensor Networks Using Clustering Techniques

Z. Banković, J.M. Moya, J.C. Vallejo, and D. Fraga

Dep. Ingeniería Electrónica, Universidad Politécnica de Madrid, Av. Complutense 30,
28040 Madrid, Spain
`{zorana, josem, jcvallejo, dfraga}@die.upm.es`

Abstract. Wireless sensor networks are usually deployed in unattended environments. This is the main reason why the update of security policies upon identifying new attacks cannot be done in a timely fashion, which gives enough time to attackers to make significant damage. Thus, it is of great importance to provide protection from unknown attacks. However, existing solutions are mostly concentrated on known attacks. In order to tackle this issue, we propose a machine learning solution for anomaly detection along with the feature extraction process that tries to detect temporal and spatial inconsistencies in the sequences of sensed values and the routing paths used to forward these values to the base station. The data produced in the presence of an attacker are treated as outliers, and detected using clustering techniques. The techniques are coupled with a reputation system, isolating in this way the compromised nodes. The proposal exhibits good performances in detecting and confining previously unseen attacks.

Keywords: wireless sensor networks, unknown attacks, clustering, reputation system.

1 Introduction

Wireless sensor network (WSNs) exhibit few important issues that make their securing very challenging, among which the most important ones are limited resources, unreliable communication and unattended operation. Due to the last one, any human interaction cannot be done in a timely fashion, including the update of the security policies upon identifying new attacks. Current security solutions can protect the networks only from the known attacks. However, attackers are always capable of designing new attacks, and thus free to make significant damage until the appropriate countermeasure becomes available. Thus, it is of great importance to provide protection from the unknown attacks as well. For this reason, anomaly detection has gained lots of interest, as it is capable of detecting unknown attacks.

However, the existing anomaly detection solutions mainly look for the deviations in the values of the parameters that capture the properties of known attacks, which means that they use so-called numerical features. Hence, their possibilities to detect unknown attacks are limited, since it is hard to define the numerical features of the unknown attacks. In order to overcome this issue, in this work we propose a machine

learning solution for anomaly detection along with the feature extraction process that does not capture the properties of the attacks, but rather relies on the existing temporal and spatial redundancy in sensor networks and tries to detect temporal and spatial inconsistencies in the sequences of sensed values and the routing paths used to forward these values to the base station. The data produced in the presence of an attacker are treated as outliers, and they are detected using clustering techniques, which are traditionally used for tackling problems in security [1]. The techniques are further coupled with the reputation system, which provides implicit response to the attackers, as the compromised nodes get isolated from the network. The proposal has been tested on a number of the attacks on the core network protocols, exhibiting good performances in detecting and confining attacks, including the previously unseen ones.

The rest of the work is organized as follows. Section 2 gives more details of the state of the art solutions. Section 3 details the proposed solution, while Section 4 provides its evaluation. Finally, conclusions are drawn in Section 5.

2 Previous Work

A number of custom intrusion detection systems (IDS) [2] for sensor networks have been proposed. Some of the representative solutions are given in [3], [4]. However, they are mainly focused on misbehaving detection, hence are capable of detecting only limited number of attacks, i.e. known attacks and their variations. In order to detect new attacks, they need to be adjusted through human interaction.

Recently few solutions that deploy machine learning techniques appeared [5], [6]. Among these solutions we can also find a few anomaly based solutions [7], [8], that claim of having the possibility to detect unknown attacks. They uphold the idea that machine learning techniques offer higher level of flexibility and adaptability to the changes of the environment, as it only takes to retrain the algorithms with new data, which can be done automatically. Furthermore, in reality we often have to deal with incomplete and noisy information, and the security requirements themselves are often fuzzy and incomplete. Machine learning techniques are known to cope well with these sorts of problems [2].

However, the feature sets they deploy mostly include those features that capture the properties of known attacks, i.e. those that are known to change under the influence of an attacker, or are known to be weak spots. This is their major deficiency, as relying on these features only the known attacks or their variations can be detected. Furthermore, it assumes that an attacker can exploit only the known vulnerabilities, but general experience is that vulnerability is detected after being exploited by an adversary. Some of them assume that the feature sets can be expanded [3], yet this again has to be done through a human intervention.

Thus, we can say that the main deficiencies of the known solutions are: the scope of attacks they can detect is limited and their adaptation has to be performed through human interaction. Thus, our aim is to provide a machine learning based solution that does not suffer from these issues, i.e. a solution that would be capable of detecting wide range of attacks, including the previously unseen ones, which would also be adaptable automatically.

3 Proposed Solution

In order to provide uninterrupted network operation, core network protocols (aggregation, routing and time synchronization) have to be secured. Regarding the attacks on the aggregation protocol, we assume that they demonstrate themselves in skewed aggregated values, which can be the result of either a number of skewed sensed values, or a compromised aggregated node. The assumption is very reasonable, having in mind that the main objective of these attacks is to provide wrong picture of the observed phenomenon.

On the other hand, in time critical systems it is mandatory to receive information within certain time window. If the attacker manages to introduce delays or desynchronize clock signal in various nodes, the received critical information will not be up to date, which can destabilize the system. Also, if the received information is not up to date, the aggregated value will be skewed, as it will also be out of date. For these reasons, and given the existing redundancy in WSNs, we believe that these attacks can be detected as temporal and/or spatial inconsistencies of sensed values.

Regarding attacks on routing protocols [13], we assume that they will introduce new and different paths than those that have been seen before. Here we have attacks whose main objective is to compromise the routing protocol, and they usually do it by spoofing or altering the data stored in the routing tables of the nodes. Thus, the resulting routing paths will be different from those used in a normal situation. In the case of wormhole for example, two nodes that are not within each other's radio range result in consecutive routing hops in routing paths, which is not possible in a normal situation. From these examples we can see that the assumption about the attacks resulting in routing paths different from those that appear in normal situation is reasonable. Thus, in this case we want to detect temporal inconsistencies in paths used by each node.

3.1 Feature Extraction and Formation of Model

Following the idea of temporal and/or spatial inconsistency in the presence of attackers, we want to provide the model of the data that would capture these properties and allow us to deploy machine learning.

For the case of sensed values, we follow the idea presented in our previous work [9], [10] based on extracted n -grams and their frequencies within different time windows. For the purpose of illustration, we will give a short example for a sensor that detects presence. Let the sensor give the following output during the time window of size 20: 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0. If we fix the n -gram size on 3, we extract all the sequences of size 3 each time moving one position forward. In this way we can observe the following sequences and the number of their occurrences within the time window: 111 – occurs 6 times, 110 – 2, 100 – 2, 000 – 6, 001 – 1, 011 – 1. Thus, we can assign them the following sequences: 111 – 0.33, 110 – 0.11, 100 – 0.11, 000 – 0.33, 001 – 0.05, 011 – 0.05. In our model, the sequences are the features and their frequencies are the corresponding feature values. Thus, the sum of the feature values is always equal to 1. This characterization is performed in predefined moments of time and takes the established amount of previous data, e.g. we can perform the characterization after every 20 time periods based on previous 40 values.

In a similar fashion, we form features for spatial characterization. The first step is to establish vicinities of nodes that historically have been giving consistent information. In this way, an n -gram for spatial characterization in a moment of time is made of the sensor outputs from that very moment. For example, if sensors S1, S2, S3 that belong to the same group each give the following output: 1 1 1 0 during four time epochs, we characterize them with the following set of n -grams (each n -gram contains at the first position the value of S1, the value of S2 at the second and the value of S3 at the third at a certain time epoch): 111 – occurs 3 times, 000 – occurs once, thus the feature value of each n -gram is: 111 – 0.75, 000 – 0.25, i.e. the frequencies within the observed period of time.

The same principle is followed for characterizing routes that a node has been using to send its sensed data to the sink. Each routing hop adds its ID to the message that is further forwarded, so the sink gets the information about the routing path together with the message. Each sensor has its own model and each feature, i.e. n -gram in the model consists of a predefined number of successive hops used in routing information coming from the node. For example, if during the characterization time, the node has used the following paths for routing its data to the sink: A-B-C-S – 3 times, A-D-E-F-S – 2 times, A-B-E-F-S – 1 time (A – the node that is sending the data, B, C, ... - other nodes in the network, S- sink), we can characterize the routing with the following n -grams ($n=3$): ABC, BCS, ADE, DEF, EFS, ABE and BEF. In all of the routes, the n -gram ABC occurs 3 times, BCS – 3, ADE – 2, DEF - 2, EFS – 3, ABE – 1, BEF – 1. The total number of n -grams is 15, so dividing the values given above with 15, we get the frequencies of each n -gram which are the values that we assign to our features, i.e. n -grams.

It is important to notice that the extracted feature vectors will not be of the same size, so we are not able to use standard distance functions. For this reason, we calculate distance using the approach presented in [11], which calculates distance between sequences.

3.2 Detection of Attacks

As previously mentioned, we treat attacks as data outliers and deploy clustering techniques, SOM and unsupervised GA explained in more detail in our previous works [9], [10]. There are two possible approaches for detecting outliers using clustering techniques [11] depending on the following two possibilities: detecting outlying clusters or detecting outlying data that belong to non-outlying clusters. Since in this work we concentrate on unknown attacks which have not been seen during the training, the second case is the one that reveals outliers. For this reason, we calculate quantization error (QE) of each input as the distance from its group center.

The important points necessary for the understanding of the principle is the deployed distance function [11], which is equivalent to Manhattan distance after making the following assumption: the feature that does not exist in the first vector while exists in the second (and vice versa) actually exists with the value equal to 0, since we can say that it occurs with 0 frequency. In this way, we get two vectors of the same size and the distance between the centre and an input is between 0 (when they are formed of the same features with the same feature values) and 2 (when the features with the

values greater than 0 are completely different). In the same way, if the set of the features of one is the subset of the feature set of the other, the distance will be between 0 and 1.

In during the testing, different n -grams occur in an input, that can happen when the node starts sending data significantly different than before or starts using different routes to send the data, the distance, which is the QE value defined previously, between it and its corresponding centre will be greater than 1. This can serve as evidence of abnormal activities happening in the node or in its routing paths. It is also a typical case when the training is performed with clean data.

3.3 Recovery from Attacks

Every sensor node is being examined by agents that execute clustering algorithms and reside on nodes in its vicinity and listen to its communication. The agents are trained separately. The system of agents is coupled with a reputation system where each node has its reputation value that basically reflects the level of confidence that others have in it based on its previous behavior. In our proposal, the output of an agent affects on the reputation system in the way that it assigns lower reputation to the nodes where it detects abnormal activities and vice versa. We further advocate avoiding any kind of interaction with the low-reputation nodes: to discard any data or request coming from these nodes or to avoid taking them as a routing hop. In this way, compromised nodes remain isolated from the network and have no role in its further performance.

In this work the reputation is calculated in the following way. For the reasons explained in the previous chapter, the value (rep) for updating overall reputation based on QE is calculated in the following way:

```
if (QE<1) rep = 1; else rep=1-QE/2;
```

There are two functions for updating the overall reputation of the node, depending whether the current reputation is below or above the established threshold that distinguishes normal and anomalous behavior. If the current reputation is above the threshold and the node starts behaving suspiciously, its reputation will fall quickly. On the other hand, if the reputation is lower than the established threshold, and the node starts behaving properly, it will need to behave properly for some time until it reaches the threshold in order to “redeem” itself. In order to achieve this, we use the function $x+\log(1.2*x)$ because it provides what we want to accomplish: if x is higher than 0.5, the output rises quickly, so the reputation rises; if x is around 0.5, the output is around 0, so the reputation will not change its value significantly; if x is smaller than 0.4, the output falls below 0. Finally, the reputation is updated in the following way:

```
if (last_reputation[node]>threshold)
    new_reputation[node]=last_reputation[node]+rep+log(1.2*rep);
else
    new_reputation[node]=last_reputation[node]+0.05*(rep+log(1.2*rep));
```

If the final value falls out from the [0, 1] range, it is rounded to 0 if it is lower than 0 or to 1 in the opposite case.

However, if during the testing of temporal coherence, we get normal data different from those that the clustering algorithms saw during the training, it is possible to get high QE value as well. On the other hand, the spatial coherence should not detect any

anomalies. Thus, the final reputation will fall only if both spatial and temporal algorithms detect anomalies. This is implemented in the following way:

```
if (value_rep < threshold)
{
    if (space_rep < threshold) result = value_rep;
        else result = 1 - value_rep; }
else result = value_rep;
```

where `value_rep` is the reputation assigned by the algorithms for temporal characterization and `space_rep` is the reputation assigned by the algorithms for spatial characterization.

Concerning the detection of routing protocol anomalies, the explained approach can tell us if there is something suspicious in routing paths of a certain node. Yet, in order to find out the nodes that are the origin of the attack, we need to add one more step. In the second step, if the reputation of the routes calculated in the previous step is lower than the established threshold, the hops that participated in bad routes will be added to the global list of bad nodes, or if they already exist, the number of their appearance in bad routes is increased. The similar principle is performed for the correct nodes. For each node, let the number of its appearances in bad routes be $nBad$ and the number of its appearances in good routes be $nGood$. Finally, if $nGood$ is greater than $nBad$, the node keeps its reputation value, and in the opposite case, it is assigned the following reputation value: $nGood / (nGood + nBad)$. In this way, as the bad node spreads its malicious behavior, its reputation will gradually decrease.

4 Evaluation of the Proposed Approach

The proposed algorithm has been tested on a simulator of sensor networks developed by our research group and designed using the C++ programming language. Since the aim of this work is to prove the detection of unknown attacks, we will present only the results in the cases the attacks have not been seen during the training. The approach will be tested in the presence of the representative attacks on core network protocols [13]:

- *Sybil*. The compromised node pretends to have multiple IDs, either false, i.e. *fabricated*, or impersonated from other legitimate nodes, i.e. *stolen IDs*. The attacker can affect on many aspects of network (aggregation, routing, etc.)
- *Misrouting*. The compromised node sends data back to the network, instead of forwarding it towards the base station, thus introducing loops in routing paths.
- *Pulse-delay*. The data from attacked node(s) have much higher latencies than in the normal case. The attack affects on time critical systems.

In the first experiment we have a scenario where 200 nodes can occupy 2000 possible positions. The attacker launches the Sybil attack from the node situated at position 800 that has stolen 21 existing IDs from the nodes situated from position 258 to position 432. The reputation evolution in time is presented in Fig. 1.a., where we can clearly observe the low reputation of the compromised nodes, i.e. the node at position 800, and the nodes whose IDs have been stolen. Thus, we can say that in this case we have detected and completely confined the Sybil attack.

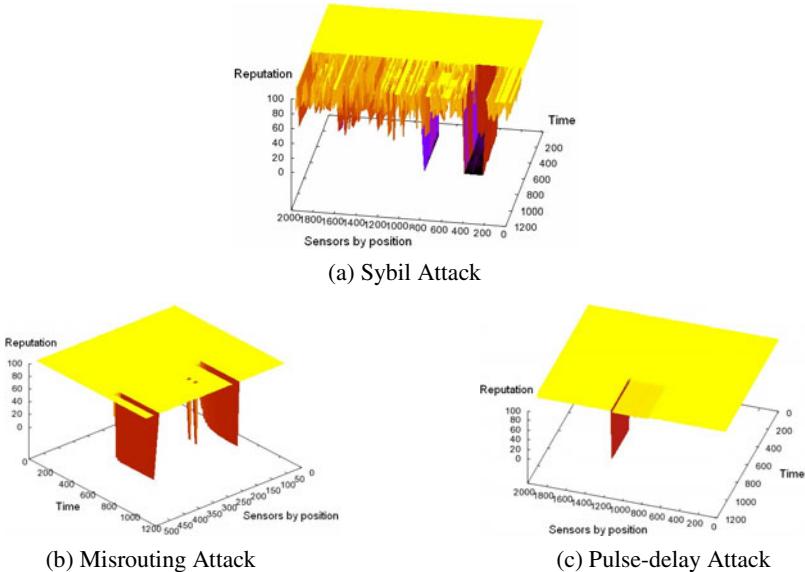


Fig. 1. Reputation Evolution

In the second experiment we have a scenario where 50 nodes can occupy one of 500 positions, and misrouting attack is launched from two nodes, situated at positions 162 and 447. The reputation evolution is given in Fig.1.b, where we can clearly see that the reputation of the malicious node is lowered to 0, which means that the attack is detected and completely confined. We can also see that one more node has its reputation falsely lowered for some time, yet it is recovered after some time.

Finally, we test the approach in the presence of pulse-delay attack, where the node at position 1196 introduces delays in sending its data to the base station. The reputation evolution is presented in Fig. 1.c. We can observe that the only node whose reputation is lowered is the malicious node, which means that we have detected and confined the attack with no false positives.

5 Conclusions

In this work we have proposed a machine learning based anomaly detection approach for detecting unknown attacks in wireless sensor networks. The attacks are treated as data outliers, and we have designed clustering algorithms for outlier detection. The algorithms are further coupled with a reputation system, which provides implicit response to attackers, as low reputation nodes remain isolated from the network. Our experiments confirm that the approach is capable of detecting and completely confining attacks on core network protocols with no false positives.

In the future we plan on reducing the resource consumption of the approach, which would allow its total or partial implementation in sensor nodes. On the other hand, the approach itself after can be deployed in other distributed systems, such as mesh or social networks, which are known to exhibit low level of security.

Acknowledgments. This work was funded by the Spanish Ministry of Industry, Tourism and Trade, under Research Grant TSI-020301-2009-18 (eCID), the Spanish Ministry of Science and Innovation, under Research Grant TEC2009-14595-C02-01, and the CENIT Project Segur@.

References

1. Corral, G., Armengol, E., Fornells, A., Golobardes, E.: Explanations of unsupervised learning clustering applied to data security analysis. *Neurocomputing* 72(13-15), 2754–2762 (2009)
2. Herrero, A., Corchado, E., Pellicer, M.A., Abraham, A.: MOVIH-IDS: A mobile-visualization hybrid intrusion detection system. *Neurocomputing* 72(13-15), 2775–2784 (2009)
3. Krontiris, I., Giannetsos, T., Dimitriou, T.: LIDeA: A Distributed Lightweight Intrusion Detection Architecture for Sensor Networks. In: 4th International Conference on Security and Privacy for Communication Networks. ACM, New York (2008)
4. Hai, T.H., Khan, F., Huh, E.-n.: Hybrid Intrusion Detection System for Wireless Sensor Networks. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part II. LNCS, vol. 4706, pp. 383–396. Springer, Heidelberg (2007)
5. Onat, I., Miri, A.: A Real-Time Node-Based Traffic Anomaly Detection Algorithm for Wireless Sensor Networks. In: Systems Communications, pp. 422–427. IEEE Press, Los Alamitos (2005)
6. Wallenta, C., Kim, J., Bentley, P.J., Hailes, S.: Detecting Interest Cache Poisoning in Sensor Networks using an Artificial Immune Algorithm. *Appl. Intell.* 32, 1–26 (2010)
7. Kaplantzis, S., Shilton, A., Mani, N., Sekercioglu, Y.A.: Detecting Selective Forwarding Attacks in WSNs using Support Vector Machines. In: Int. Sensors, Sensor Networks and Inf. Proc. Conf., pp. 335–340. IEEE Press, Los Alamitos (2007)
8. Loo, C.E., Ng, M.Y., Leckie, C., Palaniswami, M.: Intrusion Detection for Routing Attacks in Sensor Networks. *Int. J. of Dist. Sens. Net.* 2(4), 313–332 (2006)
9. Moya, J.M., Araujo, A., Bankovic, Z., de Goyeneche, J.M., Vallejo, J.C., Malagon, P., Villanueva, D., Fraga, D., Romero, E., Blesa, J.: Improving Security for SCADA Sensor Networks with Reputation Systems and SOMs. *Sensors* 9, 9380–9397 (2009)
10. Banković, Z., Moya, J.M., Araujo, A., Fraga, D., Vallejo, J.C., de Goyeneche, J.M.: Distributed Intrusion Detection System for WSNs based on a Reputation System coupled with Kernel Self-Organizing Maps. *Int. Comp. Aided Design* 17(2), 87–102 (2010)
11. Rieck, K., Laskov, P.: Linear Time Computation of Similarity for Sequential Data. *J. Mach. Learn. Res.* 9, 23–48 (2008)
12. Muñoz, A., Muruzábal, J.: Self-Organizing Maps for Outlier Detection. *Neurocomputing* 18(1-3), 33–60 (1998)
13. Roosta, T.G.: Attacks and Defenses on Ubiquitous Sensor Networks, Ph. D. Dissertation, University of California at Berkeley (2008)

A Hybrid System with Regression Trees in Steel-Making Process

Mirosław Kordos¹, Marcin Blachnik², Marcin Perzyk³, Jacek Kozłowski³,
Orestes Bystrzycki¹, Mateusz Gródek¹, Adrian Byrdziak¹, and Zenon Motyka¹

¹ University of Bielsko-Biala, Department of Mathematics and Informatics,
Bielsko-Biala, Willowa 2, Poland
mkordos@ath.bielsko.pl

² Silesian University of Technology, Department of Management and Informatics,
Katowice, Krasinskiego 8, Poland
marcin.blachnik@polsl.pl

³ Warsaw University of Technology, Faculty of Production Engineering,
Warsaw, Pl. Politechniki 1, Poland

Abstract. The paper presents a hybrid regresseion model with the main emphasis put on the regression tree unit. It discusses input and output variable transformation, determining the final decision of hybrid models and node split optimization of regression trees. Because of the ability to generate logical rules, a regression tree maybe the preferred module if it produces comparable results to other modules, therefore the optimization of node split in regression trees is discussed in more detail. A set of split criteria based on different forms of variance reduction is analyzed and guidelines for the choice of the criterion are discussed, including the trade-off between the accuracy of the tree, its size and balance between minimizing the node variance and keeping a symmetric structure of the tree. The presented approach found practical applications in the metallurgical industry.

1 Introduction

There are several issues concerning regression tasks in data mining. One of the problems is caused by the fact, that there is usually no clear rule about which data point is an outlier or wrong point and thus should be excluded from the training set. Another problem is the choice of the appropriate regression model (as neural network, support vector machine for regression (SVR), decision tree, etc.) Most of nonlinear models, as neural networks, kernel methods or even k-NN are very powerful and are able to model any shape of decision surface as they are universal approximators. However, using them we lose the possibility to understand in a simple way what they have learned. Another issue is human verification of the models. It is not only required to provide the user with extracted rules, but also the user must be given a chance to provide feedback deciding whether extracted statistical knowledge is in accordance to the specific expert knowledge. The authors faced that kind of problems while building models for optimizing steel production process in two of polish steelworks [1][2]. The control of the process was critical for the safety and economical reasons. Since hybrid systems are known to

perform usually better than single-model systems [3][4][5], we used a hybrid system built of several parallel modules: MLP neural network, support vector regression, multivariate linear regression and decision tree. But the best results were obtained by combining outputs of all the modules as follows:

$$y = \frac{\sum_i y_i w_i}{\sum_i w_i} \quad (1)$$

where y_i is the output predicted by the $i - th$ module and w_i is the weight assigned to this module. The better the module performs on the learning data, the higher weight is assigned to it. A reasonable first choice is to assign the weights that are inversely proportional to the mean squared error of the modules, but in general that is another parameter to optimize. However, the comprehensibility of the model was not less important than the model accuracy; if the operator of the process doesn't understand the predictive system decisions, he is not likely to apply them to the process. Therefore it was necessary to provide decision rules explaining the rationale behind predicting a given value.

Two common groups of models that can address the comprehensibility problems [6] are:

- Rule extraction by sequential covering
- Decision tree induction

Sequential covering is based on extracting decision rules in such a way that each new rule covers new examples not covered by previously defined rules. In other words rules are created starting from the most general to the most specific one. Decision tree induction is based on the divide and conquer paradigm, which defines the hierarchical structure of the learning problem. Decision trees, depending on how tree nodes get split, can be defined as one dimensional, multidimensional (oblique) or mixed. In the first type of trees the nodes of the tree split the dataset based on the value of a single variable, comparing it against a threshold or "belong to" operation in case of symbolic values. The choice of the variables and thresholds is discussed in the following chapters. In multidimensional trees the nodes split the dataset based on a combination of several variables. That can be a linear or quadratic model, or in general any decision making algorithm (LMT [7], NBtree [8], etc.). Mixed trees constitute a combination of one- and multi-dimensional trees [9], in which for each node the best node splitting function is selected. Although mixed and multidimensional trees are more flexible, they lose their comprehensibility, making obtained rules less readable.

One of the features of decision trees is the ability to extract decision rules, which converts the tree structure into a flat list of rules. So each branch of the tree is converted into an independent decision rule. Decision tree algorithms use the divide and conquer concept, making the tree induction process very fast. The complexity of sequential covering algorithms is much higher, so that it becomes one of limitations of the method. On the other hand rules obtained from decision tree tend to be more complex because they all have a common part (at least one common premise defined by the root node) as the tree is a hierarchical structure. This makes the rules less readable, and may require further pruning as in C4.5 rules [10].

Another undesirable property of decision trees is the possible insignificance of the root node and other top level for the interpretation of obtained results, because at the top nodes the criterion function may be highly impure.

As a remedy to that problem we can redefine the criterion function. For that purpose many different criteria have been defined such as entropy based functions: Shannon, Renyi or Tsallis [11] and others. On the other hand instead of defining new criterion functions, typical criteria may be parameterized such that the promise of the top level nodes will become more significant. This can be obtained by not splitting the node symmetrically (i.e. with equal error rate in the two subsets), but rather at the top level nodes the split would be asymmetric, and travelling further down the tree the split would be getting more and more symmetric.

The paper is divided into two theoretic sections. The first section presents a simple data transformation for reducing the influence of outliers for the system and the second section discusses parameterized split criteria function for the regression tree module. The next sections presents empirical results obtained on various real-word datasets and discuss the influence of different parameters on the quality of obtained results, including final results obtained for the metallurgical problem. The last section concludes the paper.

2 Outliers and Data Transformation

The presence of outliers and wrong data may dramatically reduce generalization abilities and limit the convergence of training process of any learning algorithm. Proper data preprocessing is helpful not only to deal with outliers but also to achieve variable sensitivity of the model in different ranges of input variables. A good practice is to standardize the data before the training, e.g. according to the following formula:

$$x_{std} = \frac{x - \bar{x}}{\sigma} \quad \sigma = \sqrt{\frac{1}{k} \sum_{i=1}^k (x_i - \bar{x})^2} \quad (2)$$

to make the influence of particular inputs independent of their physical range. One dimensional regression trees do not need the standardization of inputs, because they consider only a single attribute at a time. However, they can still benefit from the standardization of the output variable, which can be performed as the first step of outlier removal and sensitivity improvements. In practical problems, it is frequently desired to obtain a model with higher sensitivity in the intervals with more dense data (as it was in our case) or in other intervals of special interests.

To address the problem, we transfer the data through a hyperbolic tangent function (fig. (2), eq. (3)). The other advantage of the transformation is the automatic reduction of the outliers' influence on the model. We do not consider the outliers as erroneous values and thus do not reject them [11], but rather reduce their influence on the final model, because it is frequently not clear whether a given value is already an outlier or a wrong value or is still correct. The hyperbolic tangent transformation allows for a smooth reduction of the outliers, because no matter how big the value is, after the transformation it will never be greater than one or smaller than minus one. That approach

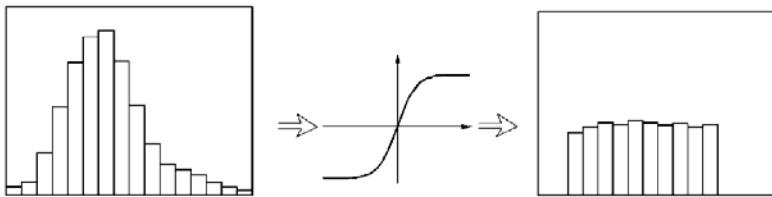


Fig. 1. The idea of transforming output data to uniform distribution

does not work well in the case of multimodal data distribution. That data must be first divided into several single-mode distribution datasets or a more complex transformation function must be used.

After the input and output attributes are standardized, we transform them by tanh function, as shown in fig. ② using the formula

$$y = \frac{1 - \exp(-\beta \cdot y + \theta)}{1 + \exp(-\beta \cdot y + \theta)} \quad (3)$$

In practice the θ value could be estimated using the mean value of the output variable y (that is 0 after standardization).

3 Parameterization of Split Criteria

The parameterization function used to split the data into two children of every node can take various forms. The following pseudo-code shows how the split points are determined: The purpose here is to find such a split point s_0 and a feature (variable) f_0 over

Algorithm 1. Tree optimization pseudocode

```

Require:  $F = [f_1, f_2, \dots, f_s]$ 
Ensure:  $\forall_{i=1:s} \text{ sizeof}(f_i) \leftarrow p$ 
for  $i = 1 \dots s$  do
     $f_i = SortFeatureElements(f_i)$ 
    for  $j = 1 \dots p$  do
         $p_L = j/p$ 
         $p_R = (p - j)/p$ 
         $v = v_0 - p_L^m \cdot v_L^n - p_R^m \cdot v_R^n$ 
        if  $v \geq q$  then
             $q = v$ 
             $s_0 = j$ 
             $f_0 = f_i$ 
        end if
    end for
end for
return  $s_0, f_0$ 

```

all possible features f_i , which maximize the variance reduction v for each tree node. v_0 is the node variance, i.e. the variance of all vector output values Y in the node. We search for the optimal split point (s_0) iterating over each input feature f_j and each value of that feature j . For that reason the vectors must be sorted in the increasing order of each feature separately, before the search for the optimal split is attempted. v_L is the variance of the left side of the node (the potential left child) and v_R of the right side. The simplest statement for v could be:

$$v = v_0 - (v_L + v_R) \quad (4)$$

Although, this maximizes variance reduction in a single node, it turns out to be only a local minimum and the entire tree created with this split criterion tends to be large and with poor generalization ability. That happens, because this split usually divides the node in two children with very different number of vectors. One child comprises very few vectors, frequently only one and the other one very many.

The solution is to multiply the variances of each child node by the number of vectors in the child node to enforce the split into two sets with more equal number of vectors.

$$v = v_0 - (p_L^n \cdot v_L^m - p_R^n \cdot v_R^m) \quad (5)$$

Where p is the number of vectors in the given node, p_L is the ratio of the number of vectors in the left child node and the total number of vectors in that node (p) and respectively p_R is the ratio in the right mode. We raised these values to different powers m and n and examine the influence of the powers on the complexity and accuracy of the decision tree. The results are discussed in the experimental result section.

4 Experimental Results

4.1 Experimental Methodology

The purpose of the experiments was to find out how the split criteria influence the structure and accuracy of regression trees. We conducted the experiments on about 10 different datasets. However, here we present results only on three of them, because the results on the other datasets showed similar dependencies.

The source code of the software is available from [12].

The experiments described in this chapter were performed with the following parameters:

- minimum node variance: $0.002/\beta^2$
- minimum number of instances in the current node: 2% of the number of instances in the training data
- minimum number of instances in a child node: 0.5% of the number of instances in the training data
- maximum number of levels in the trees: 24

4.2 Results

Concrete Compressive Strength. There are 8 input attributes (variables) in the dataset reflecting the amount of particular substances in the concrete mixture, such as cement, slag, water, etc. The task is to predict the concrete compressive strength. There are 1030 instances in the database. We used 687 instances for the training data and 343 instances for the test data. The dataset is available from the UCI Machine Learning Repository. The results are shown in Fig. 1.

Communities and Crime. There are 120 input attributes in the data set, describing various social, economical and criminal factors. The attribute to predict is per capita violent crime. After removing the instances with missing attributes, 121 instances were left. We used 81 instances for the training data and 40 instances for the test data. The dataset is available from the UCI Machine Learning Repository. The results are shown in Fig. 2.

Metallurgical problem. The dataset comes from a real metallurgical process at the phase of refining the melted steel in a ladle arc furnace to achieve desired steel properties. The inputs variables represent various measured parameters, such as temperature, energy, amount of particular elements in the steel etc. The amount of carbon that should be added to the steel refinement process is represented by variable to precipit (C). The data was standardized and the names of 12 input attributes were changed to $x_1 \dots x_{12}$. There are 1440 instances in the data set. We used 960 instances for training and 480 instances for the test dataset. The dataset is available at [13]. The results are shown in Fig. 3.

Output Transformations. As discussed in the previous chapter, we transformed the output column by hyperbolic tangent and sought for the optimal parameter " β " in the tanh equation (3).

The tree was constructed on the transformed data, however, the MSE was measured on the original test data in order to obtain a direct comparison with the results obtained on the original, untransformed data. In order to keep the algorithm properties unchanged, the parameters describing the minimum variance in the current and child nodes were changed so to keep a constant ratio of the mean deviation (square root of the current and child node variance) to the derivative of tanh function at the zero point (for $y=0$). It was found that the optimal β was close to one in most cases (table 1.). However, it significantly improved the results only in the case of a single mode distribution. For example, the distribution of the steel data was similar to a sum of three Gaussian distributions with different mean values and therefore this transformation didn't work well in this case. It was found that the transformation didn't have a significant influence on the tree size.

Forest of Decision Trees. There are ways to improve the prediction ability of a single tree. One of the methods is to create a forest of trees. In our experiments we used a forest of 10 trees. The whole training set consisted each time of a 90% of the total set and the whole model was tested in 10-fold crossvalidation, each time on the remaining 10% of vectors. One third of training vectors were randomly chosen to build each of

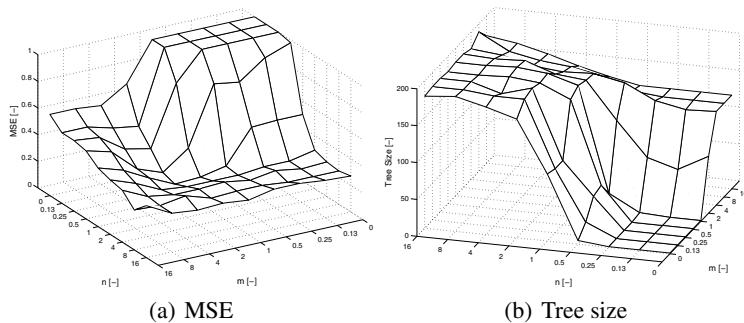


Fig. 2. n and m influence on the MSE (a) and on the tree size (b) for Concrete Compressive Strength dataset

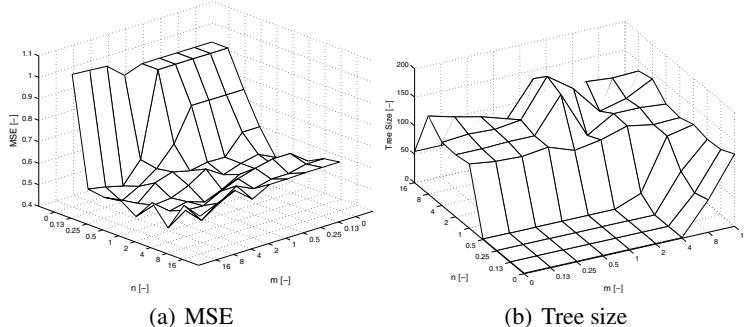


Fig. 3. n and m influence on the MSE (a) and on the tree size (b) for Communities and Crime dataset

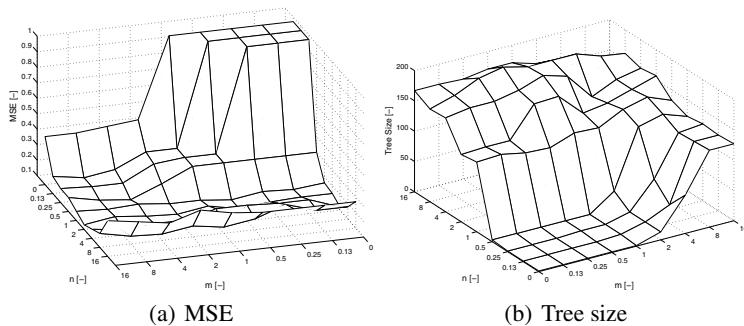


Fig. 4. n and m influence on the MSE (a) and on the tree size (b) for Steel dataset

Table 1. Influence of β for the accuracy (for $n = m = 1$)

dataset	Desc.	none	$\beta = 0.05$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.5$	$\beta = 1$	$\beta = 2$	$\beta = 5$
Concrete	MSE	0.20	0.20	0.18	0.16	0.15	0.15	1.16	0.21
	Tree size	167	164	157	167	169	177	168	149
Crime	MSE	0.58	0.58	0.56	0.54	0.46	0.35	0.58	0.71
	Tree size	95	95	85	69	85	95	95	65
Steel	MSE	0.14	0.14	0.15	0.14	0.14	0.15	0.23	0.35
	Tree size	121	121	77	95	105	111	107	59

thee trees. Each tree was then tested on the remaining 2/3 of the training vectors and the inverse of MSE achieved on this set was the quality measure of the tree. Then the final decision y was taken based a weighted average of the values predicted by the 10 trees:

$$y = \left(\sum_{i=1}^k \frac{1}{MSE_i} \right)^{-1} \sum_{i=1}^k \frac{y}{MSE_i} \quad (6)$$

5 Conclusions

The output variable transformation as discussed in section 2 allowed reducing the MSE for most datasets. In standardized data with Gaussian distribution of the output variable the assumption $\theta = 0$ is correct. For non Gaussian distributions s θ and β should be optimized simultaneously. The decision tree module, although not always achieves the highest accuracy in the whole hybrid system, is very usable because of it easily generates comprehensive logical rules. The goal of the proposed split criteria was to control the influence of variance reduction and the position of the splitting threshold/cut-off. According to the formula (5) increasing n value increases the role of the position in the split. In other words it forces the split threshold to be placed in the middle such that $p_L \approx p_R$. Reducing the value n below m $n < m$ also reduces the importance of that factor and focuses more on the pure variance reduction. To improve the decision tree accuracy the best m and n values could be derived from the formula $m = 2 - n$ where the value of n were in ranges $n \in [0.5, 2]$. Our experiments showed that the most optimal m and n values oscillate around 1 in (4). The results can be further improved with output variable transformation and with tree forest was presented as well as with post-training optimization of the tree.

Acknowledgment

The work was sponsored by the Polish Ministry of Science and Higher Education, projects No. 4866/B/T02/2010/38 and 4421/B/T02/2010/38.

References

1. Kordos, M.: Neural Network Regression for LHF Process Optimization. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008. LNCS, vol. 5506, pp. 453–460. Springer, Heidelberg (2009)
2. Blachnik, M., Mączka, K., Wieczorek, T.: A model for temperature prediction of melted steel in the electric arc furnace(EAF). LNCS, vol. 6614 (2010)
3. Corchado, E., et al.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
4. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13–15), 2729–2730 (2009)
5. Wozniak, M., Zmyslony, M.: Designing fusers on the basis of discriminants – evolutionary and neural methods of training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 590–597. Springer, Heidelberg (2010)
6. Duch, W., Setiono, R., Zurada, J.: Computational intelligence methods for understanding of data. Proceedings of the IEEE 92(5), 771–805 (2008)
7. Landwehr, N., Hall, M., Frank, E.: Logistic Model Trees. Machine Learning 95 (2005)
8. Kohavi, R.: Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In: 2nd Int. Conf. on Knowledge Discovery and Data Mining, pp. 202–207 (1996)
9. Grabczewski, K., Duch, W.: Heterogeneous forests of decision trees. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 504–509. Springer, Heidelberg (2002)
10. Quinlan, J.R.: Simplifying decision trees. Int. Journal of Man-Machine Studies 27(3) (1987)
11. Maszczyk, T., Duch, W.: Comparison of Shannon, Renyi and Tsallis Entropy Used in Decision Trees. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 643–651. Springer, Heidelberg (2008)
12. <http://www.kordos.com/tree-source.zip>
13. <http://www.kordos.com/datasets/steel.zip>

Interval Type-2 Fuzzy Modelling and Simulated Annealing for Real-World Inventory Management

Simon Miller¹, Mario Gongora², and Robert John²

¹ Horizon Digital Economy Research / Intelligent Modelling and Analysis,
University of Nottingham, Nottingham, UK

² Centre for Computational Intelligence,
De Montfort University, Leicester, UK
s.miller@nottingham.ac.uk,
{mgongora,rij}@dmu.ac.uk

Abstract. The modelling of real-world complex systems is an area of ongoing interest for the research community. Real-world systems present a variety of challenges not least of which is the problem of uncertainty inherent in their operation. In this research the problem of inventory management was chosen. The goal was to discover a suitable configuration for a Simulated Annealing search with a fuzzy inventory management problem. A hybrid of a series of Simulated Annealing configurations and an Interval Type-2 Fuzzy Logic model were used to identify suitable inventory plans for a large-scale real-world problem supplied by collaborators on a Technology Strategy Board research project (ref: H0254E).

Keywords: Interval Type-2 Fuzzy Logic, Simulated Annealing, Real-world problems, Inventory Management.

1 Introduction

Optimising inventory levels within a supply chain is an area of ongoing interest for supply chain managers. Planning the allocation of resources within a supply chain has been critical to the success of manufacturers, warehouses and retailers for many years. Poorly managed resources result in two main problems: stock outs and surplus stock. The consequence of stock outs is lost sales, and potentially lost customers. Surplus stock results in added holding cost and the possibility of stock losing value as it becomes obsolete. Holding some surplus stock is advantageous however; safety stock can be used in the event of an unexpected increase in demand or to cover lost productivity.

Various degrees of uncertainty are present in the different data sources used in supply chain management (SCM). This uncertainty is further amplified in demand forecasts by applying methods of analysis which have varying degrees of inherent uncertainty within themselves. Furthermore, other data that is often used in resource planning such as transportation and other costs, customer satisfaction information, etc. is also uncertain. Therefore, Fuzzy Logic (FL) and

especially Type-2 Fuzzy Logic (T2FL) are particularly appropriate for this problem. While traditional (or Type-1) FL (T1FL) has successfully been used many times for modelling supply chain operation (e.g., [16] and [1]), T2FL has been shown to offer a better representation of uncertainty on a number of problems (e.g., [4] and [8]). In this research an Interval Type-2 Fuzzy Logic (IT2FL) model is used, as it benefits from some of the advantages of T2FL, while incurring considerably less computation. Section 2 provides details of the model.

The search spaces involved in inventory management are often very large even for a relatively simple problem. As such, it is not possible to find a resource plan using an exhaustive search, a more efficient method needs to be selected. In this research a hybrid of a series of Simulated Annealing (SA) configurations and the IT2FL model mentioned previously were evaluated for this purpose. Section 3 describes SA in more detail, sections 4 and 5 give the test scenario used and the results respectively. Section 6 considers the conclusions that can be drawn from the work, and what form future work might take.

2 Model

The proposed model represents the interaction of nodes within a multi-tier supply chain. Figure 1 provides an example of a typical supply chain. In each tier there are one or more nodes that supply the subsequent tier with one or more products, and receive stock from the preceding tier. The first tier receives goods from an external supplier which is assumed to have infinite capacity, the final tier supplies the customer. Below the first tier, capacity is limited by node and product.

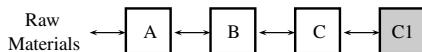


Fig. 1. A typical supply chain

Customer demand is provided by a fuzzy forecast which is given to the model at run-time. This forecast represents the demand placed upon the final tier in the SC. Tiers above this can see their own demand by looking at the suggested inventory levels at the succeeding tier, as they will be required to supply these items.

In order to use the model the following information must be provided: the number of *tiers* not including the end customer, number of *nodes* in each tier, number of end *customers*, number of *products*, number of *periods*, *service level* required as a percentage of orders filled completely, *capacities* for each product at each node (amount that can be produced in one period), *lead time* in periods for production/supply of each product at each node, *minimum order* and *unit of order* quantities for each product at each node, *initial stock* levels for each node, *distance* between nodes in successive echelons, *forecast* of customer demand, suggested *inventory levels* and costs including: *batch cost* by node, *production*

cost by product, *transport cost*, *holding cost* as a percentage of purchase price and *purchase price* by product.

Using this information the model will calculate the cost of the given resource plan, the total cost of a plan is made up of the following:

Batch cost represents the cost of administration, setting up any machines that are required, and picking the items for dispatch. *Production cost* is the cost of producing an item of a particular product. *Transport cost* is a fixed value representative of the cost per mile of transporting a batch of goods from a node to another node in the succeeding echelon. *Holding cost* is the cost of storing items. *Stock out cost* is charged for the shortfall of a product in a particular period. *Service penalties* are added to the cost of sub-optimal solutions that do not meet service level requirements.

2.1 Interval Type-2 Fuzzy Logic

As stated previously T1FL has been used to tackle the resource planning problem. However Type-1 (T1) fuzzy sets represent the fuzziness of the particular problem using a ‘non-fuzzy’ (or crisp) representation - a number in $[0, 1]$.

As Klir and Folger [10] point out:

“...it may seem problematical, if not paradoxical, that a representation of fuzziness is made using membership grades that are themselves precise real numbers.”

This paradox leads us to consider the role of Type-2 (T2) fuzzy sets as an alternative to the T1 paradigm. T2 fuzzy sets [13] represent membership grades not as numbers in $[0, 1]$, but as T1 fuzzy sets. T2 fuzzy sets have been widely used in a number of applications (see [6] and [11] for examples), and on a number of problems T2FL has been shown to outperform T1FL (e.g., [4] and [8]). Some work has been done regarding the use of optimisation methods to design T2 Fuzzy sets (e.g., [18]) however, in this work we do not optimise the sets; we use an Interval Type-2 (IT2) fuzzy model as the means to evaluate resource plans, as this work focuses on optimising the latter. Resource plans naturally take the format of a matrix of values detailing inventory by time period, node and product.

In previous work the authors have shown that Interval Type-2 Fuzzy Logic (IT2FL) [12] is an appropriate method of modelling a multi-echelon supply chain [14]. IT2FL has been used because it is computationally cheaper than general T2FL as it restricts the additional dimension, referred to as the secondary membership function, to only take the values 0 or 1. We believe that the extra degree of freedom offered over a T1FL model will allow a better representation of the uncertain and vague nature of data used in SCM.

Fuzzy arithmetic is used to calculate costs. In this model, fuzzy sets are represented using a series of α -cuts, each set is an array of pairs of intervals. Each pair shows the area of values in x covered at a particular value of μ , the first interval is the left hand side of the set, and the second the right. Operations on

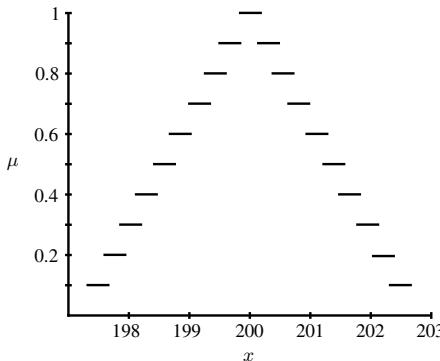


Fig. 2. Interval representation of IT2 fuzzy set ‘about 200’

the IT2 fuzzy sets are performed at the interval level, corresponding intervals (at the same μ) are taken from two sets, the operation performed and the result stored in a third fuzzy set.

Forecast demand, inventory level, transportation distance, transportation cost, stock out level, stock out cost, carry over and holding cost are represented with IT2 fuzzy numbers. For each of these values we can use the linguistic term ‘about n ’, e.g., forecast demand of product a for customer b in period c may be ‘about 200’. Figure 2 shows how the set ‘about 200’ may look with the α -cut representation used, where x is the scale of values being represented.

In order to produce an output that can be applied to a real-world supply chain, some of the IT2 fuzzy numbers need to be defuzzified. Defuzzification is the process of taking a fuzzy set and deriving a crisp value from it. To do this, the Karnik-Mendel method proposed in [7] is used. This is a widely used method that finds an interval representing the centroid of an IT2 fuzzy set. The interval can then be used to obtain a crisp number by finding its centre.

3 Optimisation

The purpose of these experiments is to evaluate the performance of a set of optimisation configurations. For the purposes of these experiments, Simulated Annealing has been chosen to conduct the search as it has been shown to work well in previous experiments (see [15]), and does not require the maintenance of a large population as is the case with some other methods (e.g., a Genetic Algorithm); for this large-scale problem, this is critical.

3.1 Simulated Annealing

Simulated Annealing (SA) [9] is inspired by a real-world phenomenon, in this case the process of heating and cooling (annealing) of metals to reduce defects. An initial solution is created, then a neighbouring solution is selected and compared

with it. The probability of the algorithm accepting the neighbour as the current solution is based upon a temperature value and the difference in quality between the two solutions. The higher the temperature value, the more likely it is that the algorithm will accept an inferior solution. The process is then repeated using the selected solution as a starting point. Over the course of a run the temperature is gradually decreased, making inferior choices less likely. SA has been successfully applied to a number of optimisation problems. For example, in [17] a series of experiments are described that show SA is an appropriate choice for optimising a production/inventory system. In [2] SA is used to solve production scheduling problems.

By using SA and IT2FL together the advantages that both methods offer can be exploited. In [5] a number of real-world problems closely related to inventory management are described including: process planning, assembly line management and dynamic scheduling that have been tackled using hybrid artificial intelligence approaches. T1FL and SA have previously been used for operations management in [3] where SA is used with a fuzzy job shop scheduling problem to find suitable schedules.

4 Test Scenario

The data set is part of a large-scale real-world scenario provided by a UK consultancy that design and manage supply chain operations for other organisations. An overview of the test scenario used can be seen in Table 1

Table 1. Real-world scenario test supply chain setup

Tiers	3
Nodes	1,2,100
Products	100
Periods	13 months
Batch Cost	100
Distance	100
Stock out Multiplier	25
Holding Cost	10% of purch. price
Production Cost	1.2
Purchase Cost	1.2
Transport Cost	0.1
Transport Distance	100 miles
Service Level	100%

In the first tier is a manufacturing warehouse based in Newquay, the second tier contains warehouses in Stockport England, Dublin Ireland and Orléans France. The final tier contains customers spread over the UK, Ireland, France and Spain. Forecast demand, capacities, leadtimes, minimum order quantities, unit of order quantities and initial stock levels were produced using information supplied by the UK consultancy.

5 Results

With the test scenario described, a number of tests were performed. The configurations can be seen in Table 2 along with the results. For these exploratory tests, time was a factor; only a single test was executed for each configuration, and fewer evaluations are performed than in previous experiments (e.g., [15]). Configurations were chosen to result in a comparable number of evaluations for each test. The intention was to use the results of this study to conduct further research with longer experiments, taking advantage of multiple runs. Even with these restrictions, a typical test took approximately 4 hours and 20 minutes.

Table 2. Results of real-world data SA tests

Stage No.	Test No.	Temp.	Dec.	Stalls	Iterations	Cost
				w/out Improvement	w/ Improvement	
1	1	100,000	1,000	50	100	£608,914,048
	2	50,000	500	50	100	£601,184,960
	3	25,000	250	50	100	£598,187,840
2	4	12,500	125	100	50	£602,834,688
	5	125,000	1250	100	50	£606,111,104
	6	150,000	1500	100	50	£610,011,648
3	7	10,000	100	100	50	£619027008
	8	15,000	150	100	50	£599,733,632
	9	20,000	200	100	50	£612,790,464
	10	30,000	300	100	50	£606,846,080
4	11	40,000	400	100	50	£604,292,352
	12	60,000	700	100	50	£604,731,008
	13	70,000	700	100	50	£600,298,880
	14	80,000	800	100	50	£608,938,368
	15	90,000	900	100	50	£619,253,760
5	16	21,000	210	100	50	£607,766,656
	17	22,000	220	100	50	£606,997,952
	18	23,000	230	100	50	£600,488,576
	19	24,000	240	100	50	£601,860,608
	20	26,000	260	100	50	£600,713,984
	21	27,000	270	100	50	£606,076,096
	22	28,000	280	100	50	£618,071,104
	23	29,000	290	100	50	£609,204,032

In stage 1 the intent was to identify an area of the search space to focus on in further tests, however the results were inconclusive. The best result came in test 3 when a starting temperature of 25,000 was used, Table 3 shows the attributes of the plan found. Less than 1% of the total cost was incurred through stockouts, as almost all (99.9%) customer demand was satisfied. Holding cost, however, was a concern, in this plan as 29.4% of the total cost came from holding stock. This suggested that too much stock was being allocated. In subsequent tests (stages 2 – 4), a wider area of the configuration space was explored, before returning the

Table 3. Real-world scenario - Test 3 - Resource plan attributes

Total Cost	Service Level	Batch Cost	Production Cost
£598,187,840	99.9%	£9,067,901 (1.5%)	£407,605,024 (68.1%)
Transport Cost	Stockout Cost	Holding Cost	
£899,522.19 (0.2%)	£4,886,310.50 (0.8%)	£175,729,056 (29.4%)	

local area (stage 5) around the temperature of 25,000. In all tests, none bettered the result found in the test 3.

6 Conclusion

A large-scale real-world case study was collected from a UK supply chain design and management consultancy, and a series of experiments was conducted to discover a suitable configuration of SA in a hybrid with an IT2FL supply chain model to determine near-optimal solutions. The tests showed that using the model, SA was able to find realistic solutions to the real-world scenario, satisfying 99.9% of customer demand. As SA was guided purely by the model, this suggests that the IT2FL model is a valid representation of the problem. However, all of the tests gave results with high holding costs, showing that there are some limitations that could be considered in additional research.

Other future work that could be undertaken includes addressing the scalability of the model. Lack of scalability limited the extent to which SA was able to operate. In these experiments solutions were represented by a 3 dimensional array representing time periods, destinations and products respectively. This caused the computational effort required to increase exponentially as scenarios became more complex, limiting the amount of solutions that could be evaluated in a reasonable time frame. Future work could focus on re-designing the solution representation so that it does not experience exponential increase in size as complexity increases; perhaps by encoding solutions or rules.

Another area of interest is a comparison of the IT2FL model with an equivalent T1 model. Using the best settings found here more extensive tests are to be conducted with each type of model, and the results analysed to determine whether one can be said to be ‘better’ than the other.

Acknowledgments. The research reported here has been funded by the Technology Strategy Board (Grant No. H0254E).

References

1. Aliev, R.A., Fazlollahi, B., Guirimov, B.G., Aliev, R.R.: Fuzzy-genetic approach to aggregate production-distribution planning in supply chain management. *Information Sciences* 177, 4241–4255 (2007)

2. Bouleimen, K., Lecocq, H.: A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research* 149(2), 268–281 (2003)
3. Fortemps, P.: Jobshop scheduling with imprecise durations: a fuzzy approach. *IEEE Transactions on Fuzzy Systems* 5(4), 557–569 (1997)
4. Hagras, H.A.: A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems* 12(4), 524–539 (2004)
5. Ibáñez, O., Cordón, O., Damas, S., Magdalena, L.: A review on the application of hybrid artificial intelligence systems to optimization problems in operations management. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 360–367. Springer, Heidelberg (2009),
http://dx.doi.org/10.1007/978-3-642-02319-4_43
6. John, R.I., Coupland, S.: Type-2 fuzzy logic a historical view. *IEEE Computational Intelligence Magazine* 2(1), 57–62 (2007)
7. Karnik, N., Mendel, J.: Centroid of a type-2 fuzzy set. *Information Sciences* 132, 195–220 (2001)
8. Karnik, N.N., Mendel, J.M.: Applications of type-2 fuzzy logic systems to forecasting of time-series. *Information Sciences* 120, 89–111 (1999)
9. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* 220(4598), 671–680 (1983)
10. Klir, G.J., Folger, T.A.: *Fuzzy Sets, Uncertainty and Information*. Prentice-Hall, Englewood Cliffs (1988)
11. Mendel, J.M.: Advances in type-2 fuzzy sets and systems. *Information Sciences* 177(1), 84–110 (2007)
12. Mendel, J.M., John, R.I., Liu, F.: Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems* 14(6), 808–821 (2006)
13. Mendel, J.M., John, R.I.B.: Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems* 10(2), 117–127 (2002)
14. Miller, S., John, R.: An interval type-2 fuzzy multiple echelon supply chain model. *Knowledge-Based Systems* 23(4), 363–368 (2010)
15. Miller, S., Gongora, M., Popova, V.: A comparison of methods for optimising resource plans. In: Proceedings of the 9th Annual Workshop on Computational Intelligence (UKCI 2009), Nottingham, UK, September 7–9, pp. 37–42 (2009),
<http://ima.ac.uk/ukci2009/>
16. Petrovic, D., Xie, Y., Burnham, K., Petrovic, R.: Coordinated control of distribution supply chains in the presence of fuzzy customer demand. *European Journal of Operational Research* 185, 146–158 (2008)
17. Tang, O.: Simulated annealing in lot sizing problems. *International Journal of Production Economics* 88(2), 173–181 (2004)
18. Wagner, C., Hagras, H.: A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots. In: Proceedings of the IEEE International Conference on Fuzzy Systems, London, UK (July 2007)

An Evidential Fusion Architecture for People Surveillance in Wide Open Areas

M. Fornaciari¹, D. Sottara², A. Prati³, P. Mello², and R. Cucchiara¹

¹ DII - Univ. of Modena and Reggio E. - Via Vignolese, 905 - 41125 Modena, Italy

² DEIS - Univ. of Bologna - Via Zamboni, 33 - 40126 Bologna, Italy

³ DISMI - Univ. of Modena and Reggio E. - Via Amendola, 2 - 42122 R.E., Italy

Abstract. A new evidential fusion architecture is proposed to build an hybrid artificial intelligent system for people surveillance in wide open areas. Authorized people and intruders are identified and localized thanks to the joint employment of cameras and RFID tags. Complex Event Processing and Transferable Belief Model are exploited for handling noisy data and uncertainty propagation. Experimental results on complex synthetic scenarios demonstrate the accuracy of the proposed solution.

1 Introduction and Related Works

There are several fields in Information Technologies research where specific strategies are required for *localizing* and *identifying* objects, such as automatic logistics of goods, surveillance, industrial pick-and-place applications, etcetera. These are typically two competing tasks since localization needs to not focus too much on a single object to have a wide coverage of the scene (and look at more objects simultaneously), whereas identification often requires a close-up of the object. This is particularly true in video surveillance and when the objects are people: identification might require zooming on the person's face and localization needs an unzoomed view to find correct position with respect to the scene, even if many people are potentially present.

Instead of cameras, alternative sensors can be used for localization and identification purposes. Among the many, RFID (Radio Frequency IDentification) tags gained much attention thanks to their ease of use, low cost and touchless way-of-reading. The identification with RFID tags is accurate, but there is a severe limitation: true positives are detected, i.e. people wearing a tag, but not true negatives, i.e. people without the tag. Regarding localization, there have been previous attempts [1][2] which use multiple tags or readers to assess people's locations using triangulation and Received Signal Strength Indicator (RSSI) value. These approaches, however, do not guarantee a sufficient accuracy.

The two tasks of localization and identification are certainly made more challenging when sensors (both cameras and RFIDs) are affected by noise, uncertainty, distractors and complex scenarios. One may think to the application of this technology to construction working sites, where wide open areas with no obliged entrances are considered. In this scenario, illumination changes, occlusions and reflexes can make hard the task for computer vision algorithms applied

to cameras, while multiple signal sources and the presence of metallic objects can introduce much noise in RFID signals.

With these premises, this work proposes to jointly use cameras and RFIDs and to take the best from both of them: camera-based systems can localize all the people in the scene (regardless if they are intruders or not), while RFIDs can identify allowed people only. The envisaged application (localization and identification of people in wide open areas) presents two strict requirements: (i) to provide a quick response, the system must work with incrementally acquired observations, so the knowledge must be updated online; (ii) both sensory modalities (cameras and RFIDs) are very heterogeneous and present a high degree of noise and uncertainty in the measurements. With these requirements, we propose a system which includes a reasoning engine with a Complex Event Processing module capable to handle uncertainty and an evidential fusion architecture (based on Transferable Belief Model - TBM [3]) to process imprecise (or missing) data, to combine various sources of information and to manage the conflict between the sources. It is an excellent tool for filtering false alarms by an optimal management of the uncertainty estimation.

TBM has been used in the literature for different applications, such as for the classification of the camera motion [4] and for developing a system for advanced driver assistance [5], but at the best of our knowledge this is the first case of application to people surveillance. The work in [5] is particularly interesting since it considers two heterogeneous sources of information (omnidirectional cameras and a laser scanner), but with similar objective (the localization of vehicles). Here, instead, the two heterogeneous sources also have heterogeneous purposes.

The final proposal is defined as a hybrid artificial intelligence system (HAIS) [6,7], which represents an useful tool since they combine both symbolic and sub-symbolic paradigms for increasing the robustness in problem-solving problems [8]. These systems are becoming more and more popular due to their ability to handle aspects such as imprecision, uncertainty or high dimensionality of data. In recent past, they have been used for computer network security and intrusion detection [8], for home care assistance through a multi-sensory architecture [9], or for minimizing the energy consumption in heating systems [10].

2 The Evidential Fusion Architecture

The surveillance system processes data coming from both cameras and RFID tags in order to generate a correct mapping between people and tags. In this way the system is able to authenticate *authorized people* (which are provided with one – and only one – tag) and to recognize *intruders* (which have no tag) and even *unfair people* (which hold more than one tag). This latter case, which can be potentially treated with the proposed evidential fusion architecture, will not be considered in this paper. The final system will process real-time data coming from real cameras and RFIDs (a first example is reported in [11]). This paper, however, will focus mainly on the reasoning engine and will test our novel proposal on synthetic yet realistic data only.

The target scenario is a wide outdoor area surveilled by a certain number of cameras with given fields of view (FoV) and RFID antennas. In addition, several locations are defined in this area in order to have the same RSSI behavior within a location and different behavior when changing the location. This can be done manually or automatically, for instance following the approach described in [11].

2.1 (Uncertain) Complex Event Processing

Each piece of information coming from the sensors can be considered as an *event*: to handle them properly, we adhere to the principles of Complex Event Processing [12] (CEP), an emerging approach to model and implement event-oriented systems. In our CEP, an *Event* is a significant state change in the observed environment at a given time, encoded using a symbolic representation which, in addition to the pieces of information required to formalize the state change, explicitly keeps a notion of the instant the event took place. Events assume importance due to their temporal, causal and hierarchical correlations with other events. The latter, in particular, allows to define a complex event A by aggregation of other events $E_{i:1..n}$ defined at a lower level of abstraction. Primitive events, then, are generated by interfaces with the environment, such as cameras and RFIDs in our case, while complex ones are raised internally. The event handlers are usually defined in a symbolic way, for example using reactive rules [13]. In CEP, however, events are usually defined and known with precision, while in our system the sensors provide only probability distributions for each person and tag in every location. An uncertain event, then, is a pair $\langle A, \pi \rangle$ associating a traditional event descriptor A to an uncertainty model, such as a probability degree π . When a complex event is generated by aggregation, its uncertainty degree is obtained by analysis and combination of the uncertainties associated to the component events: $\langle A, \pi_A \rangle \doteq \langle f(E_{j:1..n}), g(E_{j:1..n}, \pi_{j:1..n}) \rangle$. For the scope of this paper, f denotes any reactive rule used to aggregate events. The uncertainty models and the combination rules g are described in section 3.

Given a set of persons $\Pi = \{P_{i:1..\pi}\}$, a set of tags $\Theta = \{T_{j:1..\tau}\}$ and a set of locations $\Lambda = \{L_{k:1..\lambda}\}$, the relevant events in the proposed surveillance systems are as follows. All the events are instantaneous and marked with a timestamp t .

Person in Location $personInLoc(P_i, L_k, t)$: P_i was located in L_k . This primitive event is generated for each frame by a *Video Module* analyzing the scene derived from the camera with frame rate ν_C .

Tag in Location $tagInLoc(T_j, L_k, t)$: T_j was located by in L_k . Each tag transmits its identification code with frequency ν_T : the antenna provides also a RSSI value that can be analyzed by a *RFID Module* using a Hidden Markov Model to determine the position of the tag.

Person Movement P_i moved from L_k to L_h :

$$personInLoc(P_i, L_k, t-1) \wedge personInLoc(P_i, L_h, t) \Rightarrow mov(P_i, L_{kh}, t)$$

Tag Movement T_j moved from L_k to L_h :

$$tagInLoc(T_j, L_k, t-1) \wedge tagInLoc(T_j, L_h, t) \Rightarrow mov(T_j, L_{kh}, t)$$

Movement Correlation P_i and T_j performed a similar movement:

$$mov(P_i, L_{kh}, t) \wedge mov(T_j, L_{kh}, t) \Rightarrow corr(P_i, T_j, t)$$

Tag Owner P_i holds T_j : $\text{corr}(P_i, T_j, t) \Rightarrow \text{holds}(P_i, T_j, t)$

Intruder P_i was known not to hold any valid tag:

$$\exists P_i : \forall T : \neg \text{holds}(P_i, T, t) \Rightarrow \text{alarm}(P_i, t)$$

3 Implementation

The probabilistic nature of primitive events $X\text{inLoc}()$ (where “X” stands for either “person” or “tag”) is not the only source of uncertainty, since the chosen aggregation rules are not certain themselves. For example, the fact that a person P_i and a tag T_j perform the same movement at the same time may not be sufficient to decide that P_i holds T_j , possibly because there is more than one person and/or tag which performed the same movement, or the location of P_i (resp T_j) could not be determined with precision. On the other hand, if a person and a tag continue moving in the same fashion over time, the mapping between the two may become stronger. To model such concepts, we chose the TBM [3] because (i) it supports the combination of different pieces of evidence coming from different sources and (ii) it allows the information to be updated in time, thus permitting to propagate the uncertainty associated to the events.

Each probability function provided by Video and RFID modules induces a set of isopignistic belief functions on the frame of discernment Λ [14], of which we keep the q-least committed one. The movement $\text{mov}(X, L_{kh}, t)$ of a person/tag X from location L_k to L_h at time t is *possible* only when it happens between *adjacent* locations, i.e. start and end locations are reachable without passing through any other location. For each person/tag, belief functions over Λ at time $t-1$ and t are combined using the following combination rule:

$$m^M(W) = \sum_{Y, Z \subseteq \Lambda : W = (\{Y \times Z\} \cap M) \subseteq \Lambda \times \Lambda} m_{t-1}^A(Y) m_t^A(Z). \quad (1)$$

In other words, we obtain a new belief function on the power set 2^M over the frame of discernment of possible movements $M = \{L_{kh} \equiv (L_k, L_h) : L_k, L_h \in \Lambda, L_k \text{ is adjacent to } L_h\} \subseteq \Lambda \times \Lambda$.

The *correlation* between P_i and T_j is the support to the fact that P_i holds T_j . It can be derived comparing belief functions on movements of people and tags with a similarity measure which accounts for the similarity between focal elements through the Jaccard index [15], where F_i is the set of focal elements:

$$\psi(F_1, F_2) = \sum_{A \in F_1} \sum_{B \in F_2} m_1(A) \cdot m_2(B) \cdot \frac{|A \cap B|}{|A \cup B|}. \quad (2)$$

Then it can be translated, as a basic mass assignment, into the piece of evidence:

$$\text{corr}(P_i, T_j, t) \Rightarrow evd_{P_i}^\Omega = \begin{cases} m(T_j) = \psi(F_{P_i}, F_{T_j}) \\ m(I \setminus T_j) = 1 - \psi(F_{P_i}, F_{T_j}) \end{cases}, \quad (3)$$

with $\Omega = \Theta \cup \{\mathbb{k}\}$, where \mathbb{k} is the identifier of the dummy tag held by an intruder used to *close the world* on the frame of discernment of identifiers. A common

sense rule is that people *seen* at time t can hold tags *sensed* at time t or \emptyset only, and so, for each person, $\psi(F_{P_i}, F_{T_j}) = 0$ for each tag *not sensed* at time t .

System (which is the agent that entertains the belief on people's tags ownership) keeps a belief function for each person (present in the scene) over the frame of discernment Ω representing System's belief that P_i holds a subset of tags in Ω . Among different conjunctive combination rules [16], we use Dubois-Prade's rule to combine the evidence $evd_{P_i}^{\Omega}$ with the belief function of P_i . The use of a normalized combination rule ($m(\emptyset) = 0$) is consistent with the closed world assumption, and also allows to exclude the Smets' rule. The redistribution of conflicting masses over all focal elements of Dempster-Shafer's rule has poor significance in our application. Yager's rule is more cautious but less committed than Dubois-Prade's rule, in which masses resulting from pairs of conflictual focal elements are transferred to the union of these subsets. Moreover, true but conflicting information are combined as a disjunctive combination rule and thus this is the only rule which accounts for a person holding more than one tag allowing to handle the unfair people case mentioned above.

The pignistic transformation on System's beliefs about tags ownership generates a probabilistic distribution over the betting frame Ω which (picking the maximum value) indicates which tag is held by each person. If the maximum value corresponds to \emptyset then the person is an intruder and an alarm ($alarm(P_i, t)$) must be triggered, otherwise authorized people holding one tag only may be authenticated ($holds(P_i, T_j, t)$). Besides, unfair people may be recognized with a more sophisticated rule which exploits also System's belief over sets of tags.

4 Experimental Results

For evaluating the effectiveness of the proposal we present here a set of synthetic experiments, where the inputs from cameras and RFIDs are realistically simulated. The envisaged layout is composed of locations concentric with the only RFID antenna positioned exactly in the middle of the square area representing the FoV, eventually obtained merging the FoV of multiple cameras (Fig. II(a,b)).

The radius of each location is determined by the behavior of RSSI over distance from antenna obtained experimentally. This behavior is reproduced first defining the RSSI value as a function of the distance from antenna and then perturbing with white noise with standard deviation σ . Trajectories have been traced manually and so exact position, location and distance from antenna are known. Both ν_C and ν_T are 1 Hz.

While in real applications we intend to use state-of-the-art computer vision techniques, in this simulated context $personInLoc(P_i, L_h, t)$ is α if P_i is in location L_h and $(1-\alpha)/(\lambda-1)$ otherwise, with $\alpha=0.8$. A Hidden Markov Model with locations as hidden states and RSSI values as observations is trained simulating RSSI values of a tag following a simulated probing trajectory. For tag T_j , $tagInLoc(T_j, L_h, t)$ is the probability of the hidden state L_h given as observation the simulated RSSI value after the application of a median filter with a window size of 3 to further reduce the noise.

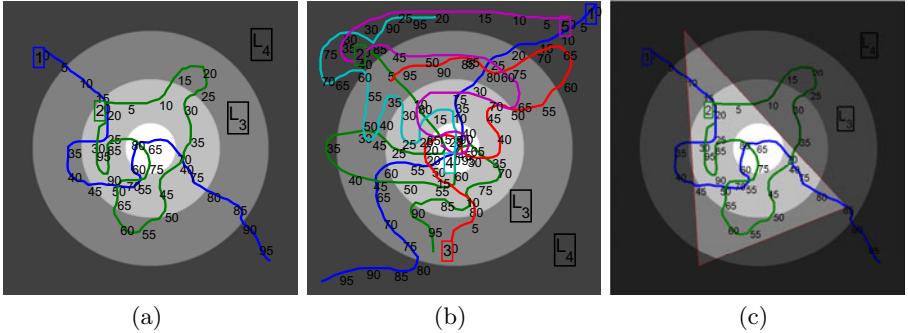


Fig. 1. Three scenarios considered in our tests. Numbers represent time.

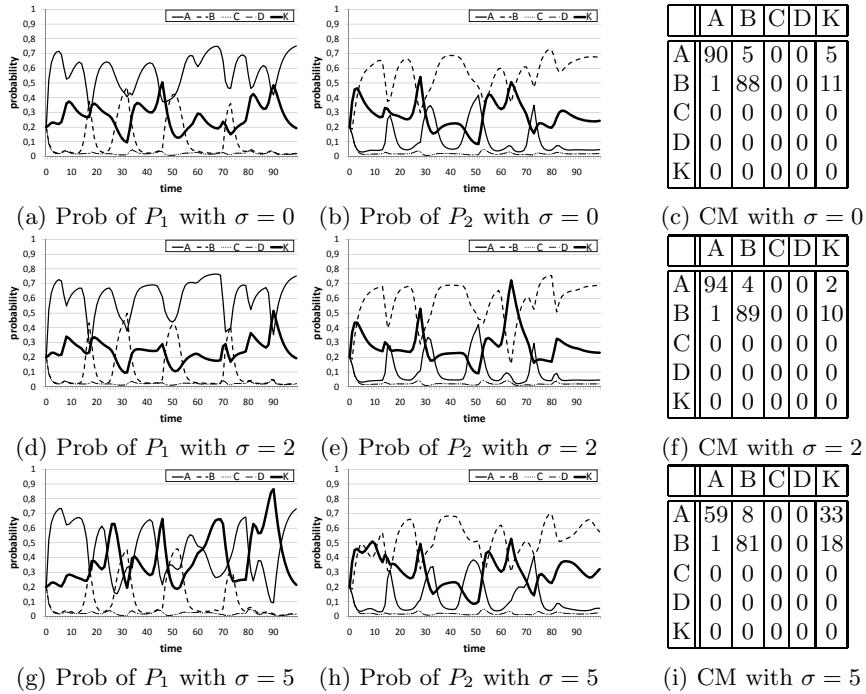


Fig. 2. Scenario with two people P_1 and P_2 holding tag A and B, respectively. Results at different values of σ : $\sigma = 0$ avg prec=96.8%, avg recall=89.0%; $\sigma = 2$ avg prec=97.3%, avg recall=91.5%; $\sigma = 5$ avg prec=94.7%, avg recall=70.0%

Hereafter three scenarios of increasing level of complexity are proposed. For each person present in the scene, the probability that he/she holds a certain tag is depicted in the graphs of Figs. 2[3]. The maximum value at a given time is the System's bet over the mapping between people and tags. The first scenario (Fig. 1(a)) represents two people moving in the scene holding tag A and B

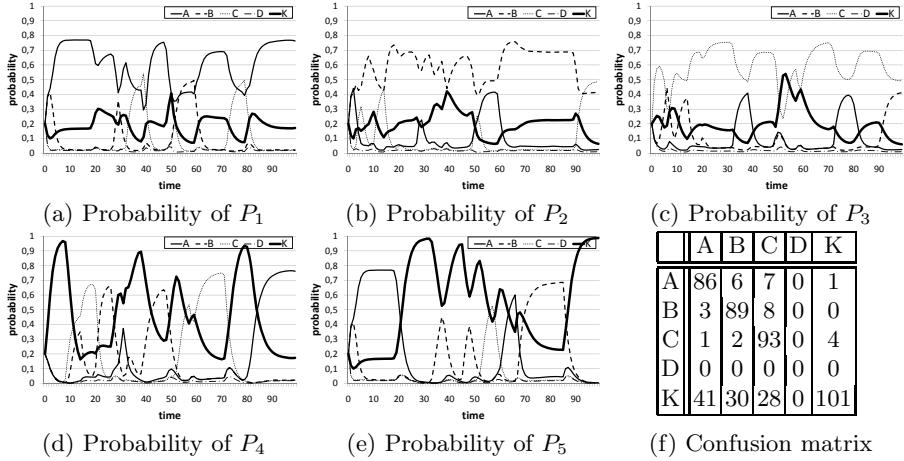


Fig. 3. Complex scenario with five people: P_1 , P_2 and P_3 hold a tag (A,B and C), while P_4 and P_5 are intruders (K) ($\sigma = 2$): avg prec=74.8%, avg recall=79.6%

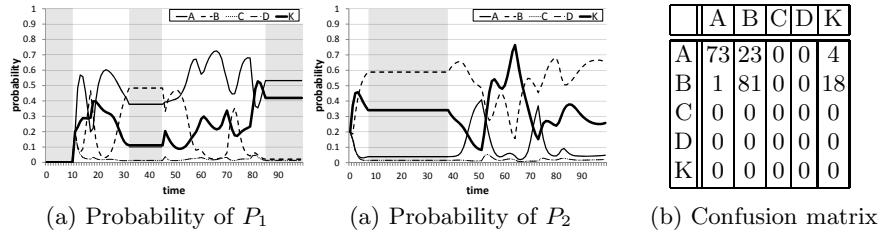


Fig. 4. Same scenario as Fig. 2, but with different FoV

respectively. The probability of the correct mapping is always the highest (Fig. 2(a-c)), except for few samples when persons change location: since tags localization is less responsive than people localization due to the HMM, for a short interval the tag and person holding it are, wrongly, localized in different locations. This scenario is analyzed also considering different amount of noise in the RSSI of the tags (Fig. 2(d-i)). Average precision and recall values (see caption of Fig. 2) are very high for $\sigma = 0$ and 2, while recall decreases to 70% in the case of high noise ($\sigma = 5$) which affects localization accuracy. The second scenario (Fig. 1(b)) is the most complex and the precision and recall drop to around 70-80% (Fig. 3). Finally, the third scenario (Fig. 1(c)) only differs from the first one for the FoV which does not cover the whole layout, thus simulating a more realistic scenario. The results in Fig. 4, where the darkened areas represent the intervals in which people are outside the FoV, with avg. precision = 88.3% and avg. recall = 77.0%, are comparable with the ones in Fig. 2, showing the robustness of the proposed approach also in more realistic cases. In general, most of the errors are due the assignment of a tag to one of the intruders, mainly

because for some time an intruder is in the same location of an authorized person. However authorized people are correctly identified most of the time.

5 Conclusions

The proposed architecture successfully demonstrates to be very accurate on synthetic yet complex data. Though more evidence must be collected on real data, we firmly believe that the proposed methodology, which is conceived to deal with noise and uncertainty typical of real data, will be effective also in that case.

References

1. Sanpechuda, T., Kovavisaruch, L.: A review of RFID localization: Applications and techniques. In: ECTI-CON, pp. 769–772 (2008)
2. Xin, H., Janaswamy, R., Ganz, A.: Scout: Outdoor localization using active RFID technology. In: BROADNETS, pp. 1–10 (2006)
3. Smets, P.: The transferable belief model. *Artif. Intell.* 66, 191–234 (1994)
4. Guironnet, M., Pellerin, D., Rombaut, M.: A fusion architecture based on tbm for camera motion classification. *Image Vision Comput.* 25, 1737–1747 (2007)
5. Clerentin, A., Delahoche, L., Marhic, B., Delafosse, M., Allart, B.: An evidential fusion architecture for advanced driver assistance. In: IEEE/RSJ IROS, pp. 327–332 (2009)
6. Medsker, L.: Hybrid Intelligent Systems. Kluwer Academic Pub., Dordrecht (1995)
7. Corchado, E., Abraham, A., de Carvalho, A.: Editorial: Hybrid intelligent algorithms and applications. *Inf. Sci.* 180, 2633–2634 (2010)
8. Herrero, Á., Corchado, E., Pellicer, M.A., Abraham, A.: Movih-ids: A mobile-visualization hybrid intrusion detection system. *Neurocomp.* 72, 2775–2784 (2009)
9. Tapia, D.I., Fraile, J.A., de Luis, A., Bajo, J.: Healthcare information fusion using context-aware agents. In: Graña Romay, M., Corchado, E., García Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 96–103. Springer, Heidelberg (2010)
10. Villar, J.R., de la Cal, E., Sedano, J.: Hybrid Artificial Intelligence Systems: Minimizing energy consumption in heating systems under uncertainty. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 583–590. Springer, Heidelberg (2008)
11. Cucchiara, R., Fornaciari, M., Prati, A., Santinelli, P.: Mutual calibration of camera motes and rfids for people localization and identification. In: ACM/IEEE ICDSC, pp. 1–8 (2010)
12. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman, Amsterdam (2002)
13. Paschke, A.: A homogenous reaction rule language for complex event processing. In: EDA-PS (2007)
14. Dubois, D., Prade, H., Smets, P.: New semantics for quantitative possibility theory. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 410–421. Springer, Heidelberg (2001)
15. Jousselme, A.L., Grenier, D.: loi Boss: A new distance between two bodies of evidence. *Information Fusion* 2, 91–101 (2001)
16. Smets, P.: Analyzing the combination of conflicting belief functions. *Inf. Fusion* 8, 387–412 (2007)

Artificial Neural Networks Application in Software Testing Selection Method

Kristina Smilgyte and Jovita Nenortaite

Kaunas University of Technology, Department of Information Systems,

Studentu st. 50-313a, LT-51368, Kaunas, Lithuania

kristina.smilgyte@gmail.com, jovita.nenortaite@ktu.lt

Abstract. The importance of software testing is growing as a concurrent part of software development. In order to improve the financial allocation of the software testing, software developers have to make a choice between automatic and manual testing methods. The solution related to the problematic choice of testing methods is presented in this paper. The method used for testing method selection is based on the application of artificial neural networks (ANN). In the paper the main idea of the method and its appliance possibilities are introduced. Experimental investigations on ANN structure selection and method evaluation are also presented in this paper.

Keywords: artificial neural networks, software testing, project manager, experiments.

1 Introduction

Software companies are using various methods of testing in software development process. It would be difficult to find the company, which has never made a choice between automatic and manual testing methods. While choosing manual testing the tester is required in the whole testing process, starting from entering testing data and finishing with results evaluation [1]. The programs and tools used in the automatic method of testing are suitable for checking other programs or module work. In such cases the tester, who is running the tests, can do other tasks and make a test summaries when tests are finished. The company saves time and finance required for testing by successfully using the tests for several times. The opposition would be the manual testing which would determine the increase of these resources. However, both methods have pros and cons [2]. The main problem is that none of these methods can fully cover each other. The problem of the automatic and manual tests is also analyzed by Aspire Systems. As the solution to the problem it proposes the test automation calculator (ROI) [3]. The main concept of the ROI is to reveal the distribution of the time and money expense in a several year period. This product of Aspire Systems does not directly offer the suitable testing method for the particular case. The investment return is not the main argument in order to select the testing method. It is essential to consider the coverage of the testing, efficiency, accuracy and other parameters as well.

The project managers coordinating the process of the project realization have to take the best solutions in different stages of the project. This paper gives the maximum regard to the stage of the information system (IS) testing. In order to coordinate the main testing tasks the project manager frequently has to face with the difficulties to decide whether the automatic tests are necessary for particular components of the system testing or not. Because of the possibility to evaluate the demand of the automatic tests, their benefit and the level of buy-off, the best conditions are given to project managers in order more effectively to distribute the resources and reduce the expenditures.

The method introduced in this paper is dedicated to project managers or testers in order to alleviate and justify the choice of the testing method. The main concept of the method is to present the recommendation whether the automatic or manual testing method is better to use or whether their usage is simply adequate. The choice of the testing method is based on the application of the artificial neural network. ANN can model complex non-linear relationships and approximate any measurement function [4]. They can be used as an effective tool for pattern classification and clustering [5], [6].

The second part of the paper gives the review of the related works. The third part introduces the method of the choice of the software testing type. The accomplished experiments by adjusting neural network in order to solve the problem of the testing method choice are reviewed in the fourth part of this paper. Moreover, the conclusions of the experiments are summarized. The fifth part gives the conclusions of the paper and the predictable works in a future.

2 Related Work

Various testing programs have a big impact on the final quality of the software. In order to present the information system to the customer without system errors, the developers of the IS seek to find them in a process of the development. Furthermore, the timely corrections of system errors require less effort. For information systems errors identification it is essential to select the suitable type of the testing (automatic or manual). It is important to take into account that automatic testing is not always effective [1] as it might require big investments while selecting the testing method. Also it is important to evaluate if the choice the testing method was appropriate.

One of the evaluation methods used by project managers is the return of investment calculation. Such calculations of automatic testing are created by Aspire Systems. Calculator introduced by them presents how much money and time is saved each year by using automatic tests in software development process. Paper [3] gives presentation of the return of investment calculator of the automatic tests. However this paper does not introduce the solution of the testing method selection problem. In most cases system testing requires the composite testing. That means that it needs more than just to know how much money and time could be saved in particular period of time.

In other case, the choice of the automatic testing closely depends on the contemplation of the tester and his/her competence, selected metrics [7], [8]. While deciding

which test method to use, one of the most popular questions is: how often the test will be used. It is possible to evaluate the advantages and disadvantages of the single method by choosing the testing method though it is not always enough. Automatic testing may save time, however it can bring opposite results when it is used just for few times. There is no concrete method or manner which can help to evaluate the need of the method type precisely analyzing the problem of the testing method choice. Nothing but the ROI product of the Aspire Systems can be used as an additional though insufficient aid by choosing the right testing type.

In this paper, considering that there is no alternative way to solve the analyzed problem the choice method of the testing type considers the series of factors, which influence the choice between the automatic and manual testing. Moreover, the experience obtained in the company of the luckiness of the automatic and manual testing is also used. The artificial neural network is trained with the cumulative examples on the use of the testing methods of the company.

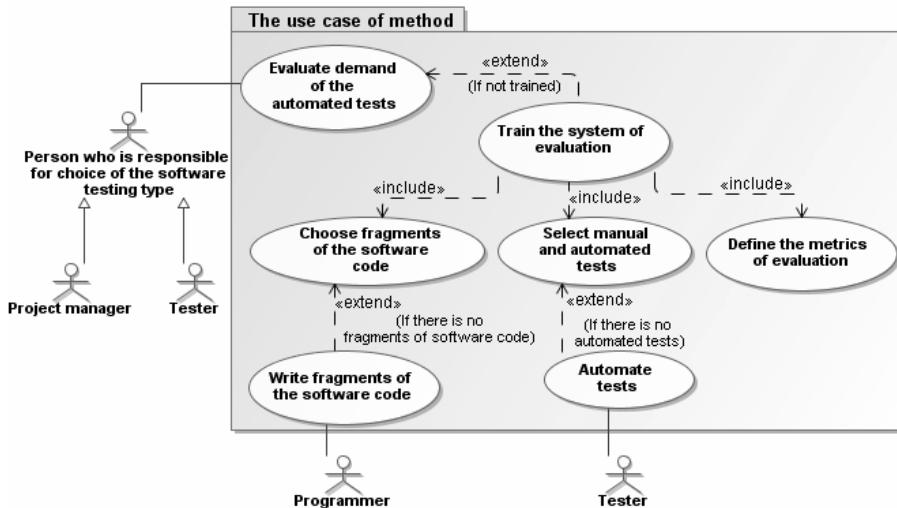
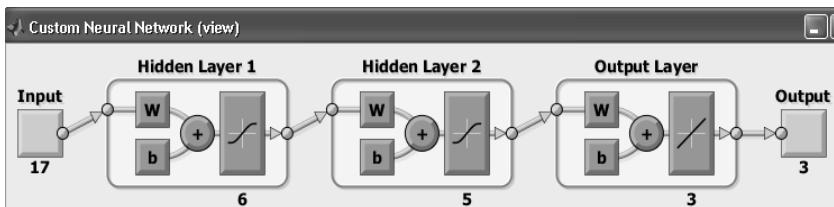
Firstly, ANN capability for this problem is analyzed. Later the study could be done applying ANN + swarm intelligence or any other computational intelligence method. According to [9], [10] papers hybridization of intelligence techniques gives better results in many areas than individual methods. In this paper, individual technique is presented.

3 Software Testing Type Selection Method

The represented method is oriented to the project managers and testers however it can be used by other concerned individuals who are interested in the testing of the information systems. The software testing type selection method base on the application of artificial neural networks can be helpful when there is no clue what testing method (automatic or manual) is better to choose.

The results of the method should be considered as the recommendation since the accuracy depends on the date which was used for training. According to this reason this method is more suitable for those companies or customers which already have historical testing data of the project. Thus the company can formulate the application of the method (artificial neural network training) based on the experience as the company already knows when the automatic or manual usage of the tests was successful or unfortunate. Also the presented method can be used by those companies which have no experience in testing. In this case, the license to use the presented method by the other company or individual is required. The general view of the appliance is presented in Figure 1.

The ANN is used to implement the method where the criterion estimation serve as network inputs and testing type serves as an output. In total, 17 different criteria are used in order to outline the realization of a single testing type (to automate or to use a manual way). The output of ANN can be 3 types: manual testing method; automatic testing method; both methods are adequate. Therefore method of the choice gives one answer out of 3 types mentioned above and ANN consists of 3 neurons at its output. The general view of the ANN use is presented in Figure 2.

**Fig. 1.** The diagram of the method appliance**Fig. 2.** The artificial neural network structure of the method

4 Experiments

In order to determine the ANN structure which would be the most suitable for the solution of the choice problem the data set allocated to network training was prepared for the tests. The inputs data and criteria of the ANN used in the method are: repeated number of test use, number of programming loops in a code, experience of the tester, the number of project members, the amount of the project tasks, the experience of the testing team (tester) in automation, the time of the project development, certain time for the testing, functional points or the number of the code lines, the tariff/valuation of the tester, number of the testing team (manual testing), number of the testing team (automatic testing), the composition of the testing team, finance for the testing, price of the testing tool, time required for the development of the automatic test, the predictable number of the project versions till the project is finished. By determining the network output structure the used ANN transfer function was evaluated. Since the three different meanings (automatic, manual, irrelevant) can be used in the network output the Tan-sigmoid transfer function was chosen. The values of the Tan-sigmoid transfer function can be between -1 and 1. In order to interpret the obtained ANN

values appropriate the single answer of method of the choice is divided into three parts – three neurons of output layer. By referring the method values to numerals the following are received: manual testing [0 0 1], automatic testing [1 0 0], the choice of the testing is irrelevant [0 1 0]. For example, the network in the output gives values [0.8 0.3 -0.06], and the maximum value is changed to 1, and the all others are changed to 0 we obtained the following [1 0 0]. In other words the maximum value of output values indicates the strongest relationship between method decision and output values. In accordance with the previous note this code means that we gain the recommendation to automate the test. Thus, the values of ANN can be accurately distinguished.

The prepared data during every experiment was delivered to the network. Overall, 1000 data of training was divided into the following: 70% training, 15% validation and 15% test data. The maximum number of epochs to train is 100 and the required accuracy of the training is 0,00001 since the precision of the network work is one of the essential criterion of the final network evaluation. The experiments were accomplished in MATLAB application. The perceptron and the relevance of the multilayer neural network in order to solve the problem of the testing choice were analyzed during these experiments. The network of perceptrons was eliminated as an inappropriate because of its considerable errors and small number of trained and new recognizable data. Therefore the experiments were fulfilled in order to determine the structure of the multilayer neural network: number of layers, number of neurons in every layer, the training function of the network and the function of the performance. The following network training functions analyzed during the experiments: trainlm (Levenberg-Marguardt), trainbr (Bayesian Regularization), trainbfg (BFGS Quasi-Newton), trainrp (Resilient Backpropagation), trainscg (Scaled Conjugate Gradient), traincgb (Conjugate Gradient with Powell/Beale Restarts), traincfg (Fletcher-Power Conjugate Gradient), traincgp (Polak-Ribiere Conjugate Gradient), trainoss (One Step Secant), traingdx (Variable Learning Rate Gradient Descent), traingdm (Gradient Descent with Momentum), traingd (Gradient Descent).

Every type of the experiment was performed 30 times in order to estimate the represented results of the ANN after the calculating the average. Implementing the experiments, estimating the general estimation of the presumptions (the correct recognition of the new data), learning accuracy, and number of acquired entries and the graph of the training the network having 6 neurons in a first hidden layer and 5 neurons in a second layer performed the best. The training function trainbr and performance function sum squared error (SSE) were used. Moreover, ANN with a single hidden layer was eliminated because its results were worse. In accordance with the data in Table 1 and by estimating the acquired number of data (overall it could acquire 1000 data), correctly predicted new data and achieved accuracy it is still unambiguously difficult to assume which experiment is the best. Consequently, the data of the Table 1 are analyzed together with the training graphs in Figure 3. M2HL6-5N network has the best results according to its parameters and training graph, which is gradually and without greater fluctuations, improve the accuracy of the training. Conversely, the network M2HL1-5N differs. However, it has the greatest recognition rate of the new data because the training graph firstly increases the accuracy and then contains the sudden angle of refraction. Therefore the achieved accuracy is the worst compare to the other experiments. M2HL4-10N, M2HL8-10N graphs are also

subsides too quickly, so if they have one better parameter the other ones distinguish better. For example, M2HL8-10N achieves one of the best precision and acquired data; however it has difficulty predicting new data. M2HL8-5N has the biggest number of acquired data and the best accuracy, but the worst recognition of the new data because its accuracy of the training graph at the first iterations was the least. To sum up, the M2HL6-5N network is chosen for the further research.

Implementing the further research when the performance function is mean squared error (MSE) the training function traincgp performed the best by surpassing the already used trainbr function (Table 2). However after the function experiments were accomplished using the sum squared error traincgp training function the best results were obtained by the traincgb function. The results can be compared graphically in Figure 4. The experiment data (asterisk) is nearest to x axis, because the used performance function is MSE, while other experiments use SSE. Consequently, the difference is so striking because the errors are not counted. Graphically, the traincgb SSE graph looks best since there are no great training fluctuations and the accuracy increases gradually.

Table 1. The data of the sorted experiments by selecting the number of neurons

Quality	M2HL1-5N	M2HL4-10N	M2HL6-5N	M2HL8-5N	M2HL8-10N
Acquired data	818.5	856.1	855.3	867.7	855.9
New data predict	85.56%	68.33%	68.33%	65.00%	70.56%
Achieved accuracy	41.21	37.34	37.25	35.50	36.51
Correct classification rate	81.96%	85.43%	85.34%	87.13%	86.20%
Time, s	2.4270	3.8811	3.5294	4.9971	6.6019

Table 2. The data of the sorted experiments by selecting the training function

Quality	traincgp mse	trainbr sse	traincgp sse	traincgb sse
Acquired data	833.5	855.3	832.1	833.7
New data predict	78.00%	68.33%	72.22%	78.33%
Achieved accuracy	0.0833	37.25	39.48	39.25
Correct classification rate	84.11%	85.34%	82.21%	83.90%
Time, s	2.8246	3.5294	2.8590	2.6821

The network using M2HL6-5N traincgb SSE predicts the new data best of all. The network during every experiment had to recognize new six data sets: 2 x manual testing, 2 x when the choice of the testing is irrelevant, 2 x automatic testing. Summing up the results of the experiments when predicting the correct answer this network shows the following results: 10 times and 30 by manual testing, 29 and 13 when the testing method is irrelevant, and 29 and 30 by automatic testing. The results show that it is important to find out the reasons why the set of data of the manual testing is predicted just 10 out of 30, while the next time 30 times out of 30. It indicates the need for further research in order to improve the accuracy, to analyze the behavior of the neuron network, when the used new data are different, when the data set for the training differs.

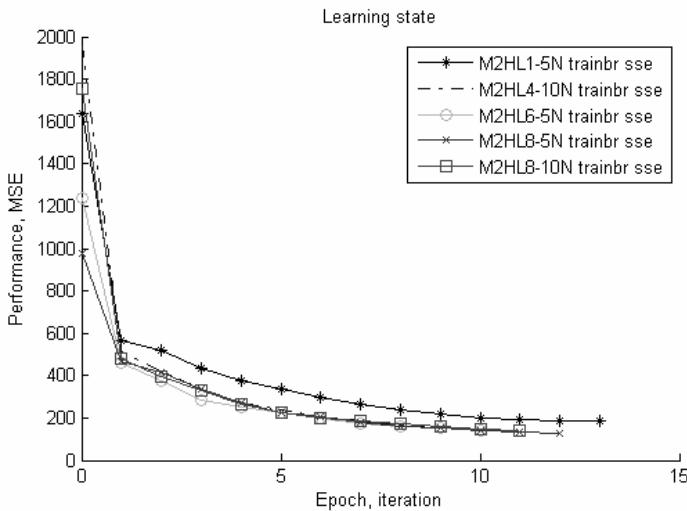


Fig. 3. The data of the sorted experiments by selecting the number of neurons

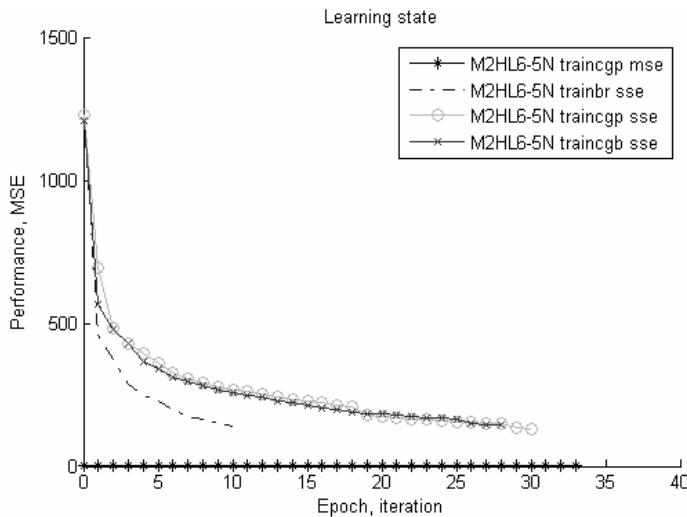


Fig. 4. The data of the sorted experiments by selecting the training function

5 Conclusion and Future Work

The accomplished ANN experiments showed that their application for the problem of the choice of the software testing type is suitable. At present, the analyzed problems are best solved by the multilayer neuron network with 2 hidden layers of which the first has 6 neurons while in a second it has 5 neurons. The sum squared error performance, Tan-sigmoid transfer functions are used and ANN is trained by using an

algorithm Conjugate Gradient with Powell/Beale Restarts. However, the future research is required in order to find out the better ANN structure, supporting the better accuracy of the prognostication. This method of the testing type choice is for the project managers and testers. In order to choose easier and to evaluate more responsible the best testing method in certain case the main purpose is to analyze the composition of the training data set, the network work while giving different training data, and how the network is resistant to select the right solution.

In a future the additional mentioned analyses and investigation between individual and hybrid intelligence methods are planned. If any chance occurs there is a need to test the developed method by using real and practical data.

References

1. Fewster, M., Graham, D.: Software Test Automation: Effective Use of Test Execution Tools, p. 574. ACM Press, New York (1999)
2. Triou, E., Abbas, Z., Kothapalle, S.: Declarative Testing: A Paradigm for Testing Software Applications. In: Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations, pp. 769–773. IEEE, New York (2009)
3. Narayanan, A.: Aspire Systems.: Test Automation ROI Calculator, <http://www.aspiresys.com/testautomationroi/>
4. Elizondo, D.A., Ortiz-De-Lazcano-Lobato, J.M., Birkenhead, R.: Choice Effect of Linear Separability Testing Methods on Constructive Neural Network Algorithms: An Empirical Study. Expert Systems with Applications 38(1), 2330–2346 (2011)
5. Nenortaitė, J., Butleris, R.: Application of Particle Swarm Optimization Algorithm to Decision Making Model Incorporating Cluster Analysis. In: 2008 Conference on Human System Interactions, pp. 88–93. IEEE, New York (2008)
6. Nenortaitė, J., Butleris, R.: Improving Business Rules Management Through the Application of Adaptive Business Intelligence Technique. Information Technology and Control 36(1), 21–28 (2009)
7. Hirayama, M., Mizuno, O., Kikuno, T.: Test Item Prioritizing Metrics for Selective Software Testing. IEICE Transactions on Information and Systems, 2733–3743 (2004)
8. Kan, S., Parrish, J., Manlove, D.: In-process Metrics for Software Testing. IBM Systems Journal 40(1), 220–241 (2001)
9. Corchado, E., Abraham, A., Carvalho, A.: Hybrid Intelligent Algorithms and Applications. Information Sciences, 2633–2634 (2010)
10. Gabrys, B.: Do Smart Adaptive Systems Exist? Hybrid Intelligent Systems Perspective. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 2–3. Springer, Heidelberg (2008)

Combining OWL Ontology and Schema Annotations in Metadata Management

Tadeusz Pankowski

Institute of Control and Information Engineering, Poznań University of Technology
tadeusz.pankowski@put.poznan.pl

Abstract. We discuss a problem of using knowledge acquired in semantic annotation and provided by OWL ontology for deciding about the existence of semantic mappings between XML schemas. The problem is identified as a metadata management problem faced in data exchange and data integration systems. We show how AI, semantic Web and database technologies may be combined to obtain schema mappings and to reason about semantic preservation of these mappings.

1 Introduction

An increasingly growing issue in distributed data exchange and data integration systems is the problem of establishing semantic relationships between independently designed data repositories. The aim is to obtain meaningful interoperation between peer-to-peer connected systems. To this order, one needs semantic mappings between the schemas of data repositories, i.e., a set of expressions that specify how the data in one repository corresponds to the data in the other. To cope with the problem we need hybrid intelligence techniques [5] combining different AI, semantic Web and database methods and technologies. Among these techniques we consider: (1) *Semantic schema annotation* – a process of stating correspondences between concepts in OWL ontology and structures in XML schema. In general, this activity, like *ontology alignment*, has received steady attention in the database and AI communities over the years [13,6,10]. (2) *Theory of schema mappings*, in particular composition and inversion of schema mappings [8,2]. (3) *Reasoning techniques* [9], especially on a class of so called *source-to-target dependencies* [1] used to express mappings.

A well-defined mapping should preserve semantics of data. Unfortunately, very often the semantics behind the schema is unclear and two syntactically similar schemas can be semantically different. For example, the rule `paper → author+` might be interpreted as the property `WrittenBy` in one schema, or as the property `CitedBy` in another. Then, transferring data between repositories with such schemas can be misleading.

In this paper, we focus on some formal aspects for deciding whether two given XML schemas (or their parts defined as tree patterns) can be mapped one into the other preserving data semantics. We assume that schemas are annotated (manually or quasi-automatically [10]) in OWL ontology: labels are annotated

by classes, non-terminal edges – by object properties, and terminal edges – by data properties. The result of the annotation is used for automatic derivation of XML-to-OWL mappings. The mappings, as well as the ontological knowledge, are then used for inferring XML-to-XML mappings between underlying schemas.

The approach is illustrated in Figure 1. I_1 and I_2 are instances of XML schemas S_1 and S_2 , respectively. These instances are related by means of XML-to-OWL mappings μ_1 and μ_2 (automatically derived from annotations) to some instances O_1 and O_2 of OWL ontology \mathcal{O} . Next, reasoning techniques are used to prove that a mapping m_{12} from O_1 to O_2 is satisfied. If yes, then the mapping \mathcal{M}_{12} from I_1 to I_2 is obtained as the result of composition, $\mathcal{M}_{12} = \mu_1 \circ m_{12} \circ \mu_2^{-1}$, where μ_2^{-1} is an inversion of μ_2 .

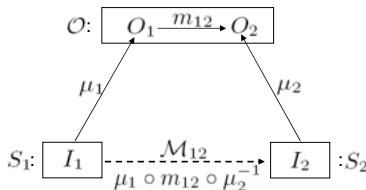


Fig. 1. The XML-to-XML mapping \mathcal{M}_{12} is obtained as the composition of an XML-to-OWL mapping μ_1 , an OWL-to-OWL mapping m_{12} and an OWL-to-XML mapping μ_2^{-1} (an inversion of μ_2)

Further on, we will focus on: (1) automatic derivation of XML-to-OWL mappings (μ_1 and μ_2) based on annotations, and (2) deciding about the existence of an OWL-to-OWL mapping m_{12} . In our solution we take advantages of OWL DL, in particular the OWL DL construct `hasValue` allows us to specify classes based on the existence of particular property values. Thanks to this, we are able to capture specialization abstraction controlled by classification attributes (like `type` data element in schema S_1 , Figure 2).

The rest of the paper is organized as follows: In Section 2, all necessary definitions are provided. In particular, we define a relational representation of OWL ontology, and the notion of annotation of an XML schema into the ontology. In Section 3, semantics for annotation is proposed. A semantic relationship (subsumption) between tree patterns is discussed in Section 4. Section 5 concludes the paper.

2 XML Data, OWL Ontology and Annotation

Let *Lab* be a set of labels and *Sring* a set of text values (literals); let `#PCDATA` be a distinguished label denoting text values.

Definition 1. An XML schema over Lab is a tuple $S = (\text{top}, \rho)$, where
 (a) $\text{top} \in \text{Lab}$ is the root label, (b) ρ is a function assigning regular expressions,
 e , to labels, $e ::= \#PCDATA \mid l \mid l? \mid l+ \mid l* \mid e\ e$, where $l \in \text{Lab} \setminus \{\text{top}\}$.

Definition 2. An XML tree over Lab is a tuple $I = (r, N, Edge, \lambda, Text)$, if:
 (a) N is a finite set of nodes; r is a distinguished root node, $r \in N$;
 (b) $Edge \subseteq N \times N \setminus \{r\}$ is a relation introducing tree structure into N ;
 (c) $\lambda : N \rightarrow Lab$ is a function assigning labels to nodes (#PCDATA is assigned to terminal nodes);
 (d) $Text : N \rightarrow String$ is a partial function assigning text values to nodes labeled with #PCDATA.

Definition 3. A tree pattern, π , over L is an expression conforming to the syntax $\pi ::= \#PCDATA \mid c \mid l \mid l[\pi_1, \dots, \pi_k]$, where: $c \in String$; $l \in Lab$, $l[\pi_1, \dots, \pi_k]$ is a tree rooted in l . Instead of $l[\#PCDATA]$ we will write $l[]$.

An XML tree I satisfying a schema S will be called an *instance* of S , denoted $I \models S$. A tree pattern π will be said to be a tree pattern over schema S , if its structure conforms to the structure of S .

There are three *elementary* tree patterns: (a) l – a single label, (b) $l[]$ – a terminal edge (#PCDATA occurs in $\rho(l)$); (c) $l_1[l_2]$ – a non-terminal edge (l_2 occurs in $\rho(l_1)$). Elementary tree patterns will be subjects of annotation in an OWL ontology [11]. In this paper, we assume a relational representation of OWL ontology.

Definition 4. A tuple $\mathcal{O} = (CA, OPA, DPA, \Delta)$ is an OWL ontology, if: (a) CA (ClassAssertion) is a binary relational name, $CA(C, a)$ states that the individual a is an instance of the class C ; (b) OPA (ObjectPropertyAssertion) is a ternary relation name, $OPA(R, a_1, a_2)$ states that the individual a_1 is connected by the object property R to the individual a_2 ; (c) DPA (DataProperty Assertion) is a ternary relation name, $DPA(D, a, c)$ states that the individual a is connected by the data property D to the text value c ; (d) Δ is a set of axioms of the form (called also OWL-to-OWL mappings)

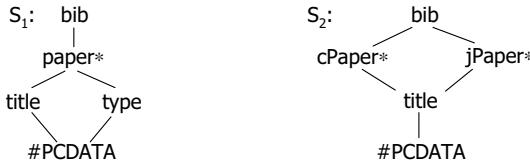
$$\forall \mathbf{y}_1, \mathbf{v} (\Psi_1(\mathbf{y}_1, \mathbf{v}) \Rightarrow \exists \mathbf{y}_2 \Psi_2(\mathbf{y}_1, \mathbf{y}_2, \mathbf{v})), \quad (1)$$

where (a) \mathbf{y}_1 and \mathbf{y}_2 are vectors of variables valued with ontology's individuals and \mathbf{v} is a vector of variables valued with text values (literals); (b) Ψ_1 and Ψ_2 are conjunctions of atomic formulas built from relational names CA , OPA , and DPA constants and variables.

Let us denote: *Class* – a set of classes, *ObjectProperty* – a set of object properties, *DataProperty* – a set of data properties, *Individual* – a set of individuals.

Definition 5. A tuple O of relations, $O = (CA^O, OPA^O, DPA^O)$, is an instance of OWL ontology \mathcal{O} , denoted $O \models \mathcal{O}$, if: (a) $CA^O \subseteq Class \times Individual$;
 (b) $OPA^O \subseteq ObjectProperty \times Individual \times Individual$;
 (c) $DPA^O \subseteq DataProperty \times Individual \times String$;
 (d) in O all axioms from Δ are satisfied.

Definition 6. An annotation of an XML schema S into OWL ontology \mathcal{O} is a function α assigning classes to labels, object properties to non-terminal edges, and data properties to terminal labels. An annotation will be denoted: $l \mapsto C$, $C \in Class$; $l_1[l_2] \mapsto R$, $R \in ObjectProperty$; $l[] \mapsto D$, $D \in DataProperty$.

**Fig. 2.** Graphical representations of two XML schemas

Sample XML schemas (in graphical representation) are given in Figure 2. They describe bibliography (**bib**). In S_1 each paper has title and type – among types of papers we distinguish *conference papers* denoted by **type** = "c", and *journal papers* denoted by **type** = "j". In S_2 papers are classified as conference papers, **cPaper**, and journal papers, **jPaper**, respectively. Each paper has the title. The schemas will be annotated in OWL ontology, where:

$$\begin{aligned} \text{Class} &= \{\text{Bibliography}, \text{Paper}, \text{Title}, \text{Type}, \text{ConfPaper}, \text{JournalPaper}, \dots\}, \\ \text{ObjectProperty} &= \{\text{Contains}, \text{hasTitle}, \text{hasType}, \dots\}, \\ \text{DataProperty} &= \{\text{valueOfTitle}, \text{valueOfType}, \dots\}. \end{aligned}$$

The set Δ of axioms in the ontology includes, among others, the following axioms (variables occurring on the left-hand site are quantified universally):

- (A1) $CA(\text{ConfPaper}, y) \Rightarrow CA(\text{Paper}, y)$
- (A2) $DPA(\text{valueOfTitle}, y, v) \Rightarrow CA(\text{Title}, y)$
- (A3) $DPA(\text{valueOfType}, y, v) \Rightarrow CA(\text{Type}, y)$
- (A4) $OPA(\text{hasTitle}, y_1, y_2) \Rightarrow CA(\text{Paper}, y_1) \wedge CA(\text{Title}, y_2)$
- (A5) $OPA(\text{hasType}, y_1, y_2) \Rightarrow CA(\text{Paper}, y_1) \wedge CA(\text{Type}, y_2)$
- (A6) $CA(\text{Paper}, y_1) \wedge OPA(\text{hasType}, y_1, y_2) \wedge DPA(\text{valueOfType}, y_2, "c") \Rightarrow CA(\text{ConfPaper}, y_1)$
- (A7) $CA(\text{ConfPaper}, y_1) \Rightarrow \exists y_2 (OPA(\text{hasType}, y_1, y_2) \wedge DPA(\text{valueOfType}, y_2, "c"))$

1. Rule (A1) says that **ConfPaper** is a subclass of **Paper**.
2. Rules (A2) and (A3) defines domain classes for individuals occurring in data properties **valueOfTitle** and **valueOfType**.
3. Rules (A4) and (A5) defines domain and range classes for individuals occurring in object properties **hasTitle** and **hasType**.
4. Rule (A6) says that a paper with special properties is a conference paper.
5. Rule (A7) asserts that conference paper is a paper with a particular object property and data property.

Annotation of S_1 into \mathcal{O} is the following:

$$\begin{aligned} \text{bib} &\mapsto \text{Bibliography}, \text{paper} \mapsto \text{Paper}, \text{title} \mapsto \text{Title}, \text{type} \mapsto \text{Type}; \\ \text{bib[paper]} &\mapsto \text{Contains}, \text{paper[title]} \mapsto \text{hasTitle}, \text{paper[type]} \mapsto \text{hasType}; \\ \text{title[]} &\mapsto \text{valueOfTitle}, \text{type[]} \mapsto \text{valueOfType}. \end{aligned}$$

3 Semantics of Annotations: XML-to-OWL Mappings

Before defining semantics for annotations, we establish a relationships between XML tree and OWL ontology. The diagram in Figure 3 represents conceptual model of database of OWL ontology, conceptual model of XML tree and a functional relationship h relating (interpreting) nodes with individuals. One individual can correspond to many nodes – the tree structure of XML enforces that one individual could be represented by different (duplicating each other) nodes. Thus, we assume that every node in XML tree has semantic interpretation, i.e. a node x is interpreted by $h(x)$. Note that *String* is shared by the conceptual model of XML tree and the conceptual model of OWL ontology.

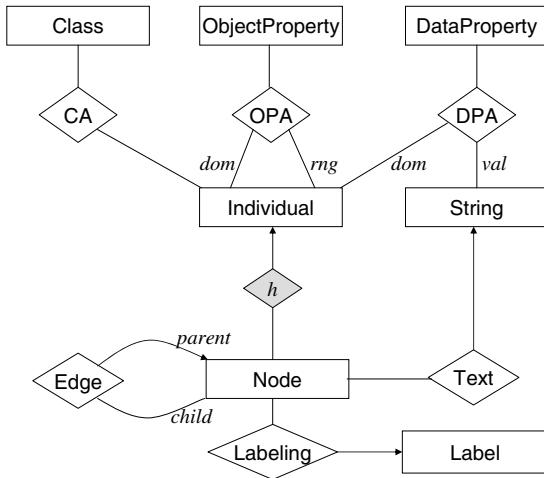


Fig. 3. Entity-Relationship diagram modeling XML tree, OWL ontology and correspondences between nodes and individuals. Arrowed lines denote functional relationships.

In conformance with the diagram in Figure 3 and the above discussion, we will use the following notations (we always assume that x, x_1, x_2 are evaluated with nodes, y is evaluated with individuals, and v with strings):

1. $l(x)$ – iff node x has label l , i.e. $\lambda(x, l)$ is true.
2. $Edge(x_1, x_2)$ – iff x_1 is the parent of x_2 ; the following parent-to-child constraint is required, saying that different parents have disjoint sets of children

$$Edge(x_1, x_2) \wedge Edge(x'_1, x'_2) \wedge x_1 \neq x'_1 \Rightarrow x_2 \neq x'_2.$$

3. $Text(x, v)$ – iff node x has the text value v .
4. h is a function from the set of nodes into the set of individuals; $h(x)$ is an individual y represented by x ; by h^{-1} is denoted the *preimage* of h , i.e.

$$h^{-1}(y) = \{x \mid h(x) = y\}.$$

If two nodes, x_1 and x_2 , represents the same individual y , then $h^{-1}(y) = \{x_1, x_2\}$. Later on, we will use $h(x_1, \dots, x_k)$ as the abbreviation for $(h(x_1), \dots, h(x_k))$.

The semantics of an annotation is define by means of XML-to-OWL mappings, understood as a class of *full source-to-target dependencies* [18], defined as follows.

Definition 7. An XML-to-OWL mapping is a formula of the form

$$\forall \mathbf{x}, \mathbf{v} (\Phi(\mathbf{x}, \mathbf{v}) \Rightarrow \Psi(h(\mathbf{x}), \mathbf{v})), \quad (2)$$

where (a) \mathbf{x} , and \mathbf{v} are vectors of variables valuated with nodes, and text values, respectively; (b) Φ – (the source formula) is a conjunction of atomic formulas of the form $l(x)$, $Edge(x_1, x_2)$, or $Text(x, v)$; (c) Ψ – (the target formula) is a conjunction of atomic formulas over CA, OPA, and DPA.

An algorithm creating XML-to-OWL mapping, μ , to a tree patterns based on the annotation of the underlying XML schema, is proposed in [12].

Example 1. For tree pattern $\pi_1 = \text{paper}[\text{title}[], \text{type}["c"]]$ over S_1 , and for appropriate annotation, we can create the following XML-to-OWL mapping:

$$\mu_1 = \forall x_1, x_2, x_3, v_1 (\Phi_1(x_1, x_2, x_3, v_1) \Rightarrow \Psi_1(h(x_1, x_2, x_3), v_1)),$$

where

$$\begin{aligned} \Phi_1 &= \text{paper}(x_1) \wedge \text{title}(x_2) \wedge \text{type}(x_3) \wedge Edge(x_1, x_2) \\ &\quad \wedge Edge(x_1, x_3) \wedge Text(x_2, v) \wedge Text(x_3, "c") \\ \Psi_1 &= CA(\text{Paper}, h(x_1)) \wedge CA(\text{Title}, h(x_2)) \wedge CA(\text{Type}, h(x_3)) \\ &\quad \wedge OPA(\text{hasTitle}, h(x_1), h(x_2)) \wedge OPA(\text{hasType}, h(x_1), h(x_3)) \\ &\quad \wedge DPA(\text{valueOfTitle}, h(x_2), v) \wedge DPA(\text{valueOfType}, h(x_3), "c"), \end{aligned}$$

and for $\pi_2 = \text{cPaper}[\text{title}[]]$ over S_2 :

$$\mu_2 = \forall x_1, x_2, v (\Phi_2(x_1, x_2, v) \Rightarrow \Psi_2(h(x_1, x_2), v)),$$

where

$$\begin{aligned} \Phi_2 &= \text{cPaper}(x_1) \wedge \text{title}(x_2) \wedge Edge(x_1, x_2) \wedge Text(x_2, v) \\ \Psi_2 &= CA(\text{ConfPaper}, h(x_1)) \wedge CA(\text{Title}, h(x_2)) \\ &\quad \wedge OPA(\text{hasTitle}, h(x_1), h(x_2)) \wedge DPA(\text{valueOfTitle}, h(x_2), v) \end{aligned}$$

4 Semantic Subsumptions between Tree Patterns

A mapping between two XML schemas specifies how data structured under a source schema is to be transformed into data structured under a given target schema. In order to be sure that these transformation preserves the semantics of data we should be convinced that the source schema is semantically subsumed by the target schema.

Definition 8. Let μ_1 and μ_2 be XML-to-OWL mappings from, respectively, π_1 and π_2 , to OWL ontology \mathcal{O}

$$\begin{aligned}\mu_1 &= \forall \mathbf{x}_1, \mathbf{v}_1 (\Phi_1(\mathbf{x}_1, \mathbf{v}_1) \Rightarrow \Psi_1(h(\mathbf{x}_1), \mathbf{v}_1)), \\ \mu_2 &= \forall \mathbf{x}_2, \mathbf{v}_2 (\Phi_2(\mathbf{x}_2, \mathbf{v}_2) \Rightarrow \Psi_2(h(\mathbf{x}_2), \mathbf{v}_2)).\end{aligned}$$

We say the π_1 is (semantically) subsumed by π_2 , denoted $\pi_1 \sqsubseteq \pi_2$, if the following deduction rule holds, by consistent substitution of variables for terms (unification)

$$\frac{\Delta}{\Psi_1(\mathbf{y}, \mathbf{v}) \Rightarrow \exists \mathbf{y}' \Psi_2(\mathbf{y}, \mathbf{y}', \mathbf{v})} \quad (3)$$

where Δ is the set of axioms of \mathcal{O} .

In order to create the deduction rule (3), we have to consistently substitute terms with variables in Ψ_1 , Ψ_2 and in every formula δ in the set Δ of axioms (for details, see [12]).

The following example illustrates how the semantic subsumption between tree patterns can be proven.

Example 2. We can prove that

$$\text{cPaper}[\text{title}[]] \sqsubseteq \text{paper}[\text{title}[], \text{type}["c"]].$$

Indeed, let $\Psi_1(h(x_1), h(x_2), h(x_3), v_1)$ and $\Psi_2(h(x_1), h(x_2), v)$ be defined as in Example 1, and let t/y' denote that a term t is replaced with variable y . In the deduction we will also need axioms (A1), (A5) and (A7). After consistent substitution of variables, we obtain the following deduction rule:

$$\frac{A1(p), A5(p, t), A7(p, n, y_2)}{\psi_2(p, n, v) \Rightarrow \exists y_2 \psi_1(p, n, y_2, v)}$$

i.e.

$$\frac{\begin{array}{c} CA(\text{ConfPaper}, p) \Rightarrow CA(\text{Paper}, p), \\ OPA(\text{hasType}, p, y_2) \Rightarrow CA(\text{Paper}, p) \wedge CA(\text{Type}, y_2), \\ CA(\text{ConfPaper}, p) \Rightarrow \exists y_2 (OPA(\text{hasType}, p, y_2) \\ \wedge DPA(\text{valueOfTitle}, y_2, "c")) \end{array}}{\begin{array}{c} CA(\text{ConfPaper}, p) \wedge CA(\text{Title}, n) \wedge OPA(\text{hasTitle}, p, n) \\ \wedge OPA(\text{valueOfTitle}, n, v) \Rightarrow \\ \exists y_2 (CA(\text{Paper}, p) \wedge CA(\text{Title}, n) \wedge CA(\text{Type}, y_2) \\ \wedge OPA(\text{hasTitle}, p, n) \wedge OPA(\text{hasType}, p, y_2) \\ \wedge OPA(\text{valueOfTitle}, n, v) \wedge OPA(\text{valueOfType}, y_2, "c")) \end{array}}$$

It is easy to show that the conclusion of this deduction rule can be reduced to TRUE. Similarly, we can show fulfillment of the opposite subsumption

$$\text{paper}[\text{title}[], \text{type}["c"]] \sqsubseteq \text{cPaper}[\text{title}[]].$$

Thus, the tree patterns are semantically equivalent in the considered environment (i.e. by the given ontology \mathcal{O} and the annotation α)

$$\text{paper}[\text{title}[], \text{type}["c"]] \equiv \text{cPaper}[\text{title}[]].$$

5 Conclusions

In this paper we propose a method for combining both, the knowledge acquired in a process of semantic annotation, and the knowledge provided by an OWL ontology, to verify the existence of semantic subsumption between XML schemas (or parts of them defined as tree patterns). The existence of the subsumption is the necessary condition for semantics-preserving mappings between those (sub)schemas. In general, the semantic annotation is a complex activity involving different methods and techniques developed in AI and IR. Results of annotation can be used to generate complex XML-to-OWL mappings. To analyze mappings, we have used results of theory of mappings, developed intensively in [8, 3, 7, 2], as well as conventional reasoning in first-order logic. The results of the paper form a theoretical basis for dealing with schema mappings in a system for semantic integration of XML data in P2P environment [4].

Acknowledgement. The work was supported in part by the Polish Ministry of Science and Higher Education under Grant 3695/B/T02/2009/36.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Arenas, M., Barceló, P., Libkin, L., Murlak, F.: Relational and XML Data Exchange. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, San Francisco (2010)
3. Arenas, M., Pérez, J., Reutter, J.L., Riveros, C.: Foundations of schema mapping management. In: Paredaens, J., Gucht, D.V. (eds.) PODS, pp. 227–238. ACM, New York (2010)
4. Brzykcy, G., Bartoszek, J., Pankowski, T.: Schema Mappings and Agents' Actions in P2P Data Integration System. Journal of Universal Computer Science 14(7), 1048–1060 (2008)
5. Corchado, E., Abraham, A., de Carvalho, A.C.P.L.F.: Hybrid intelligent algorithms and applications. Inf. Sci. 180(14), 2633–2634 (2010)
6. Doan, A., Halevy, A.Y.: Semantic integration research in the database community: A brief survey. AI Magazine 26(1), 83–94 (2005)
7. Fagin, R.: Inverting schema mappings. ACM Trans. Database Syst. 32(4) (2007)
8. Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.C.: Composing schema mappings: Second-order dependencies to the rescue. ACM Trans. Database Syst. 30(4), 994–1055 (2005)
9. Lloyd, J.W.: Foundations of logic programming. Springer, Berlin (1987)
10. Madhavan, J., Bernstein, P.A., Doan, A., Halevy, A.Y.: Corpus-based schema matching. In: ICDE 2005, pp. 57–68. IEEE Computer Society, Los Alamitos (2005)
11. OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (2009), <http://www.w3.org/TR/owl-syntax>
12. Pankowski, T.: Detecting semantics-preserving XML schema mappings based on annotations to OWL ontology. In: EDBT/ICDT Workshop on Logic in Databases (LID 2011). ACM Digital Library, New York (2011)
13. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal 10(4), 334–350 (2001)

Benchmarking IBHM Method Using NN3 Competition Dataset

Pawel Zawistowski and Jarosław Arabas

Warsaw University of Technology, Institute of Electronic Systems
`{P.Zawistowski.2,jarabas}@elka.pw.edu.pl`

Abstract. We apply a novel black box approximation algorithm, called IBHM, to learn both structure and parameters of a nonlinear regression model. IBHM incrementally creates a model as a weighted sum of activation functions which are nonlinear functions of the input vector. In each iteration the error between the current model and the approximated function is analyzed and a function is selected with the highest possible correlation with the observed error. This function is then added to the set of the model's activation functions and the process repeats. In effect IBHM determines both the model structure and parameter values. In this paper we briefly outline the method and present the results on the NN3 benchmark set. We compare results with other state-of-the-art methods that share a similar model structure: Multilayer Perceptron with a single hidden layer and Support Vector Regression.

Keywords: neural networks, svr, time series, forecasting, approximation, non-linear models, black box modeling, hybrid models.

1 Introduction

Time series forecasting is a problem, arising in various fields of research and industry, which involves creating a model capable of predicting future values of a given time series. A popular approach is to build a regression model that makes the forecast as a function of past values from a certain time window [4]. Diversity of nonlinear black box regression techniques such as Multilayer Perceptrons (MLPs), Radial Basis Functions (RBFs) [5] and Support Vector Regression machines (SVR) [11] can be applied to perform such regression. Quality of regression results is highly dependent on a proper choice of the model structure which translates to the number of nonlinear functions of the input vector that are aggregated to produce the model's output.

For SVR and RBF, the model structure creation is built into the learning process. For MLP no generic method to select the structure exists. Variety of approaches have been proposed to overcome this drawback, e.g., to grow networks using cascade correlation [6] or to prune existing networks [12].

In this paper we present results of yet another algorithm to build the approximator structure in parallel to learning its parameters. The method, called IBHM (Incrementally Built Heterogeneous Model), has been introduced in [2] and [3].

The method builds a hybrid model which allows to use in the hidden layer neurons of various kind, including radial basis and sigmoid/dot product functions. Hidden neurons are added in an incremental fashion and the method to select a neuron to be added is based on correlation between the approximating error and the neuron's output. The correlation is maximized using Covariance Matrix Adaptation Evolution Strategy — CMA-ES [7].

Combination of Evolutionary Computation (EC) with Neural Networks (NNs) is a popular choice to both designing and tuning NNs — see [14] for a comprehensive overview. Typically, networks of one canonical form (e.g. RBF or MLP) undergo evolutionary optimization. Less popular seem approaches that combine neurons of various kind in one network — see [8] for an example. In this paper we investigate in more details efficiency of the IBHM training process. For that reason we put aside the heterogeneity of the hidden layer and present a specialized IBHM version which uses only one type of hidden neurons. Doing that we are able to perform comparison with other well established and recognized training algorithms of MLP and SVR models.

An overview of the IBHM method is presented in section 2. Section 3 outlines the methodology of testing on the NN3 benchmark data. We also briefly overview two other methods which share similar model of nonlinearity which are MLP and SVR with hyperbolic tangent kernel. Test results and their discussion are provided in section 4. Section 5 concludes the paper.

2 Learning Structure and Weights of MLP Using IBHM

Let $f : R^n \rightarrow R$ be the approximated function, $g : R \rightarrow R$ be an activation function and $h : R^n \rightarrow R$ be a scalarization function. We consider approximators $\hat{f}_k : R^n \rightarrow R$ defined as linear combinations of a form

$$\hat{f}_k(\mathbf{x}) = \sum_{i=1}^k w_i \cdot g(a_i h(\mathbf{x}, \mathbf{d}_i) + b_i) + w_0 , \quad (1)$$

where \mathbf{d}_i are parameter vectors and a_i, b_i are scalar parameters. By ε_k we denote the error function $\varepsilon_k(\mathbf{x}) = \hat{f}_k(\mathbf{x}) - f(\mathbf{x})$ and by MSE (Mean Squared Error) we denote the value $MSE(X) = \frac{1}{|X|} \sum_{\mathbf{x} \in X} (\varepsilon_k(\mathbf{x}))^2$ where X is a set of samples.

IBHM, which is outlined in Algorithm 1, incrementally builds the model f_k by iteratively searching for the combination of parameters obtaining the best correlation of the term $g(a_i h(\mathbf{x}, \mathbf{d}_i) + b_i)$ with the error function $\varepsilon_{k-1}(\mathbf{x})$.

We take advantage of the observation that the perfect model \hat{f}_k would have a form $\hat{f}_k(\mathbf{x}) = \hat{f}_{k-1}(\mathbf{x}) + \varepsilon_{k-1}(\mathbf{x})$. In other words, in a perfect model we should have $|r(\varepsilon_{k-1}(\mathbf{x}), g(a_k h(\mathbf{x}, \mathbf{d}_k) + b_k))| = 1$ where r stands for the linear (Pearson) correlation coefficient. Observe that if the function g is monotonous then vector \mathbf{d}_k can be found by maximizing the rank (Spearman) correlation coefficient $\rho(\varepsilon_{k-1}(\mathbf{x}), h(\mathbf{x}, \mathbf{d}))$ with respect to \mathbf{d} . In a perfect case this correlation coefficient should equal one. Then the proper choice of parameters a_k, b_k can be obtained by maximizing the linear correlation $|r(\varepsilon_{k-1}(\mathbf{x}), g(a_k h(\mathbf{x}, \mathbf{d}_k) + b_k))|$.

Algorithm 1. IBHM

Input: $X = \{\mathbf{x}_1, \dots, \mathbf{x}_m : \mathbf{x}_i \in R^n\}$ - training sample set
Result: \hat{f} - approximation function

```

1  $\varepsilon_0(\mathbf{x}) \leftarrow f(\mathbf{x})$ ,  $k \leftarrow 0$ 
2 while there is no increase in  $AIC(X)$  do
3    $k \leftarrow k + 1$ 
   /* Part 1 - finding scalarization vector */  

4    $\mathbf{d}_k \leftarrow \arg \max_{\mathbf{d}} |\rho_\omega(h(X, \mathbf{d}), \varepsilon_{k-1}(X))|$ 
   /* Part 2 - finding activation function parameters */  

5    $(a_k, b_k) \leftarrow \arg \max_{(a, b)} |r_\omega(h_j(a \cdot h(X) + b), \varepsilon_{k-1}(X))|$ 
6   assume  $\hat{f}_k(\mathbf{x}) = \sum_{i=1, \dots, k} w_i \cdot g(a_i \cdot h(\mathbf{x}, \mathbf{d}_i) + b_i)$ 
7   assume  $\hat{\varepsilon}_k(\mathbf{x}) = f_k(\mathbf{x}) - f(\mathbf{x})$ 
   /* Part 3 - extending the approximator */  

8    $[w_0, \dots, w_k] \leftarrow \arg \min_{[w_0, \dots, w_k]} \sum_{\mathbf{x} \in X} (\varepsilon_k(\mathbf{x}))^2$ 
9 end
10  $\hat{f}(\mathbf{x}) = \hat{f}_k(\mathbf{x})$ 
```

In each iteration of IBHM we find a vector \mathbf{d} that maximizes the rank correlation between the scalarization function and the error function. The next step is a proper choice of the nonlinear transformation of the scalarized argument vector. In both correlation maximization steps, parameters of nonlinear models are optimized. Therefore the resulting optimization problem may have multiple local maxima and a global optimization method needs to be applied. In the final part of the iteration the approximator \hat{f}_k is enlarged by one functional term and weights w_0, \dots, w_k are adjusted to minimize the mean squared approximation error. As the model (1) is linear with respect to w_0, \dots, w_k this problem has a unique minimum which can be found using convex optimization methods. The main loop stops when an increase in Akaike Information Criterion (AIC) would be observed, where AIC is defined as

$$AIC(X) = 2 \cdot p + |X| \cdot \ln \left(\sum_{\mathbf{x} \in X} (\varepsilon_k(\mathbf{x}))^2 \right), \quad (2)$$

where p is the number of parameters estimated for the model.

When analyzing IBHM results reported in [23] we observed that the method analyzes global change patterns of the error function. In this contribution we introduce a modification which allows to capture local features for better approximation accuracy. The modification consists in introducing weighted coefficients of linear correlation r_ω and rank correlation ρ_ω defined as follows

$$r_\omega(X, Y) = \frac{E_\omega(XY) - E_\omega(X) \cdot E_\omega(Y)}{\sqrt{(E_\omega(X^2) - E_\omega^2(X)) \cdot (E_\omega(Y^2) - E_\omega^2(Y))}}, \quad (3)$$

$$\rho_\omega(X, Y) = r_\omega(rank(X), rank(Y)), \quad (4)$$

where E_ω is the weighted mean which uses the weighing function $\omega : R \rightarrow R$:

$$E_\omega(X) = \frac{\sum_{x \in X} \omega(x) \cdot x}{\sum_{x \in X} \omega(x)}, \quad \omega(x) = \frac{1}{\sqrt{2\pi}v} \cdot e^{\frac{-x^2}{2v^2}}. \quad (5)$$

Value of the user-defined parameter v controls the degree of “locality” of the weighted correlation function. Basing on preliminary tuning we suggest to set its value to $v = (u - l)/3$ where l and u are the lower and upper limits of values which can be taken by elements of the vector X .

It should be noted that in its original formulation IBHM allows to assign a different scalarization function h and a different activation function g for each element from the sum (1). In this paper we decided to constrain on the dot product and hyperbolic tangent to allow for comparison with MLP. For the same reason we assumed that MLP contains only one hidden layer and SVR uses only the hyperbolic tangent kernel.

3 Experiments and Results

NN3 benchmark has been introduced to allow for comparison of prediction abilities of various models during the time series forecasting competition held in 2007 [10]. This benchmark consists of 111 sets containing time series from the business domain. For each time series its history is to be used to prepare predictions for the next 18 time steps. Originally, all competing algorithms were evaluated using a squared mean average percentage error defined as $SMAPE = (200/n) \cdot \sum_{t=1}^n |\hat{x}(t) - x(t)| / (\hat{x}(t) + x(t))$ where $x(t)$ is the original time series and $\hat{x}(t)$ is its prediction. SMAPE may however give misleading results in some situations, e.g. when some denominator values would be negative and some other positive. During the testing phase we observed models which exploited this drawback and yielded SMAPE values below 1%, while the best NN3 competitors achieved SMAPE values around 14 – 15%. Therefore we decided to adopt the NN3 benchmark data and to use the MSE as a comparison criterion.

3.1 Transformation of Time Series Prediction into Regression

All training time series from the benchmark set were linearly scaled to fit into the range $[-0.5, 0.5]$. We chose that range rather than $[-1, 1]$ to address potential problems of saturation effects which may appear when the test data exceed the range of the training data. The test series were put aside and used only during the final evaluation stage.

To predict future values of the time series we used a first-order differenced, nonlinear autocorrelation model which defines the forecast as

$$\hat{x}(t) = x(t - \tau_1) + \hat{f}(x(t - \tau_1), \dots, x(t - \tau_n)) \quad (6)$$

where τ_1, \dots, τ_n are lags and \hat{f} is a nonlinear function to be approximated.

Quality of results is greatly influenced by a careful choice of the number of lags and their values. We assumed that the minimum lag value is 18 which equals the prediction horizon. The maximum lag value we assumed somewhat arbitrarily to equal 35 since we wanted to avoid situations when the training set is too small. We defined lags independently for each time series according to the following rules. The first lag τ_1 was chosen as the lag for which the maximum autocorrelation function value of the time series is observed. To determine the remaining lags we considered the differentiated time series $y(t) = x(t) - x(t - \tau_1)$. Values of τ_2, \dots, τ_n were chosen as lags where absolute values of the partial autocorrelation function (PACF) exceeded 0 at the level of significance greater than 0.05.

We are aware that there might exist more efficient data preparation techniques but our goal was to compare the relative efficiency of IBHM, SVR and MLP rather than to obtain the best possible forecasting result.

Table 1. Summary of models created by the compared algorithms

Algorithm	model structure
	$\hat{f}(\mathbf{x}) = \sum_{i=1}^k w_i \cdot \tanh(\mathbf{d}_i^T \cdot \mathbf{x} + b_i) + b$
	where:
MLP	\mathbf{d}_i weight vector of connections from inputs to the i -th neuron b_i bias of the i -th neuron w_i i -th weight of the output neuron b bias of the output neuron
	the user has to set: k number of neurons in the hidden layer
	$\hat{f}(\mathbf{x}) = \sum_{i=1}^k w_i \cdot \tanh(\gamma \cdot \mathbf{x}_i^T \mathbf{x} + \beta) + b$
SVR	where: k number of support vectors w_i weight of the i -th support vector, \mathbf{x}_i i -th support vector — an element from the training set, b bias
	the user has to set: γ, β kernel parameters ϵ insensitivity parameter (used for learning) C regularization constant (used for learning)
IBHM	$\hat{f}(\mathbf{x}) = \sum_{i=1}^k w_i \cdot \tanh(a_i \cdot \mathbf{d}_i^T \mathbf{x} + b_i) + b$ where the parameters are commented in the text nearby equation (II) the user has to set: v “locality” parameter, currently set to 0.3 times the range of the input data

3.2 Setup of Algorithms under Comparison

IBHM was implemented under Matlab. Global optimization (steps 4 and 5) was performed by the Matlab implementation of CMA-ES provided by Hansen in [1].

Since CMA-ES is a very effective method in both local and global optimization, we do not need to use a hybrid approach that comprises Evolutionary Algoirthm with some local tuning as suggested in [13]. Optimization of weights (step 7) was performed using the Matlab implementation of the Nelder-Mead simplex — *fminsearch*.

MLP model contained a single layer of hidden neurons with the hyperbolic tangent function. We used implementation provided in the Matlab's Neural Network Toolbox. SVR model with the hyperbolic tangent kernel function was considered. We used the Matlab implementation under LibSVM [5]. Table II summarizes functional form and parameters of models under comparison.

3.3 Test Procedure

The procedure used to prepare and evaluate forecasts consisted of three steps:

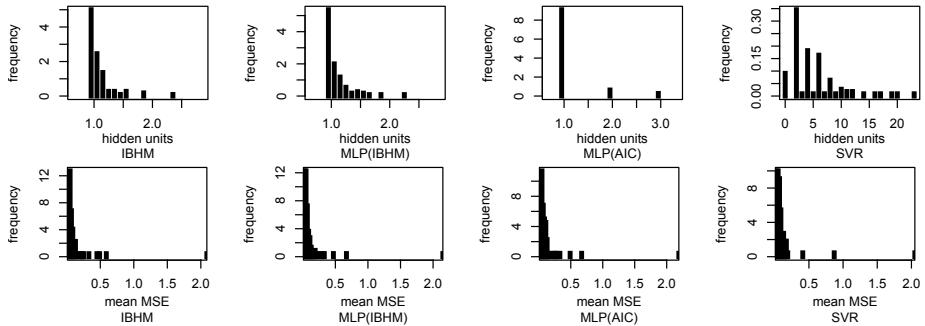
1. Preparation of each time series following the procedure in section 3.1
2. Selection of the best settings of SVR and MLP for each series separately by minimizing the Akaike Information Criterion (2) for the training set. The minimization was performed using a grid search for the following sets of parameter values:
 - for SVR: $\gamma \in \{2^{-10}, 2^{-9}, \dots, 2^1\}$, $\epsilon \in \{2^{-8}, 2^{-7}, \dots, 2^1\}$, $C \in \{2^{-10}, 2^{-8}, \dots, 2^2\}$, $\beta \in \{2^{-8}, 2^{-6}, \dots, 2^2\}$
 - for MLP: $k \in \{1, \dots, 50\}$
3. Training of four models: SVR and MLP (selected by the AIC minimization), IBHM and MLP whose number of hidden neurons was determined by IBHM. For each time series and each model we performed 25 independent training and testing cycles reporting the MSE value for the test set each time. Thus each model was characterized with a population of 25 results for each time series. We used these results to perform pairwise comparison of models. In the comparison we assumed that model A is better than B only when the median MSE for A is smaller than for B and the difference is statistically significant according to the Wilcoxon test at the significance level 0.05. Rank of a model for a time series equals one plus the number of models whose results are better for that time series.

3.4 Results

Table 2 presents aggregated results for the whole NN3 benchmark. We report mean and standard deviation of the mean value of the MSE for the test set obtained by each method for each time series. In addition we report mean value of the number of hidden neurons (MLP), support vectors (SVR) and nonlinear terms (IBHM), and the mean value of the rank of each model. Presented results have been averaged over all time series from the benchmark set. The table is accompanied with histograms of the number of hidden units and of the MSE error values for all time series — see Figure II.

Table 2. Summary of results aggregated over the whole NN3 benchmark

model	MSE mean	MSE std. dev.	no. of hidden units	rank
IBHM	0.0912	0.2118	1.09	1.76
MLP(IBHM)	0.0909	0.2167	1.10	1.83
MLP(AIC)	0.0939	0.2191	1.11	2.07
SVR	0.0932	0.2095	4.80	2.46

**Fig. 1.** Histograms of the number of hidden units and of the mean MSE in created models

From the results we conclude that models created by all algorithms under comparison were characterized by a comparable level of the average MSE error. Detailed analysis revealed that in 74% of cases the best MLP structure according to AIC was also selected by IBHM. Analysis of ranks shows also that in some cases IBHM yielded models of a better structure than models resulting from applying the AIC for MLP. This explains better results of the MLP structure suggested by IBHM than resulting from the AIC minimization.

From the histograms of the number of hidden units it follows that a vast majority of generated models contained only a single hidden neuron. This observation indicates that most of the time series from the NN3 benchmark seem quite easy to perform prediction. Closer look at the distribution of mean MSE reveals three time series with substantially larger error. To inspect these extreme cases, in Table 3 we provide results of comparison in a form of quality measures aggregated for these three time series only. In that subset of data, superiority of IBHM over other competitors is clearly visible in terms of ranks. IBHM also

Table 3. Summary of results aggregated over 3 most difficult time series

model	MSE mean	MSE std. dev.	no. of hidden units	rank
IBHM	1.0576	0.9096	1.16	1.33
MLP(IBHM)	1.1134	0.9451	1.05	2.00
SVR	1.1085	0.8453	6.00	2.00
MLP(AIC)	1.1123	0.9366	1.66	2.66

yields models with the smallest mean MSE, although this criterion does not allow to observe big differences between the compared algorithms.

4 Summary

We used the NN3 benchmark data to present the ability of the IBHM method to build good quality models which can compete with models generated by well established algorithms such as MLP and SVR. The presented algorithm can be successfully used both as a standalone method and as a procedure to suggest the optimum network structure for single layer MLP networks. The algorithm requires only a single tunable parameter which makes it easy to use as an "off-the-shelf" tool.

Future work on IBHM will include comparison with other approaches to grow nonlinear approximators, such as [6], and recent hybrid models as [8]. We also plan to make use of the full version of IBHM which allows to combine in a single model many types of scalarization functions h and activation function g , and to perform more exhaustive benchmarking.

References

1. http://www.lri.fr/~hansen/cmaes_inmatlab.html
2. Arabas, J., Dydyński, A.: An Algorithm of Incremental Construction of Nonlinear Parametric Approximators. In: Arabas, J. (ed.) Evolutionary Computation and Global Optimization 2006, pp. 31–38. Warsaw Univ. of Techn. Press, Poland (2006)
3. Arabas, J., Dydyński, A.: Nonlinear Time-Series Modeling and Prediction using Correlation Analysis. Proc. in Applied Mathematics and Mechanics 7, 2030013–2030014 (2007)
4. Box, G., Jenkins, G., Reinsel, G.: Time Series Analysis, Forecasting and Control. Prentice-Hall, Englewood Cliffs (1994)
5. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
6. Fahlman, S., Lebiere, C.: The cascade-correlation learning architecture. Advances in Neural Information Processing Systems 2, 524–532 (1990)
7. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. IEEE Congr. on Evol. Comp., 1769–1776 (2005)
8. Martínez-Estudillo, F., et al.: Evolutionary product-unit neural networks classifiers. Neurocomputing 72(1), 548–561 (2009)
9. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1999)
10. Artificial Neural Network and Computational Intelligence Forecasting Competition, <http://www.neural-forecasting-competition.com/NN3/index.htm>
11. Schölkopf, B., Smola, A.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge (2002)
12. Reed, R.: Pruning algorithms — a survey. IEEE Trans. on Neural Networks 4(3), 740–747 (1993)
13. Wedge, D., et al.: On global-local artificial neural networks for function approximation. IEEE Trans. on Neural Networks 17(4), 942–952 (2006)
14. Yao, X.: Evolving Artificial Neural Networks. Proc. of the IEEE 87(9), 1423–1447 (1999)

Face Tracking Using Adaptive Appearance Models and Convolutional Neural Network

Bogusław Rymut and Bogdan Kwolek

Rzeszów University of Technology

W. Pola 2, 35-959 Rzeszów, Poland

<http://www.prz.edu.pl>

Abstract. One inherent problem of online learning based trackers is drift consisting in a gradual accommodation of the tracker to non-targets. This paper proposes an algorithm that does not suffer from the template drift inherent in a naive implementation of the online appearance models. The tracking is done via particle swarm optimization algorithm built on adaptive appearance models. The convolutional neural network based face detections are employed to support the re-diversification of the swarm in the course of the tracking. Such candidate solutions vote simultaneously towards true location of the face through correcting the fitness function. In particular, the hybrid algorithm has better recovery capability in case of tracking failure.

Keywords: Swarm intelligence, Hybrid intelligent algorithms.

1 Introduction

Face tracking is an important problem for various applications, like video surveillance, user friendly interfaces, emotion recognition, biometrics and face recognition. It is an active area of the research because of the lack of a satisfactory tracking system that can deal with intrinsic and extrinsic distortions. The face can be tracked as single entity or as several individual facial features that were selected in advance [1]. Tao et al. [2] present a system, which tracks the location of the face and facial features using a probability network encoding the spatial relationships between facial features. A method proposed in [3] uses shape constrained search technique in combination with a set of feature templates, which are updated using a nearest neighbor approach. Some methods combine several techniques in a single tracking system. An example of such an approach is in [4], where fast motion-based face tracking and neural network based face detection are combined to achieve better tracking. In [5], Haar cascade based face detection is used to correct the proposal distribution for the particle filter.

Online appearance models were successfully used for facial features tracking [6] on account of their strong capability to adapt to variations of the appearance. However, the discussed algorithm like other template-based algorithms is not drift free. Template drift is a common phenomenon in which the target gradually shifts away from the template and the template is occupied progressively step by

step by background objects due to template update by background pixels. Thus, if something happens wrong in some number of consecutive frames, for example due to complete occlusion of the target or very fast motion, the tracking can be lost and the algorithm should be re-initialized in order to continue the tracking of the target. On the other hand, because the observations are noisy, the error grows gradually over time and this in turn leads to accumulation of the drift, which can lead to difficulties in any long term tracking.

Several remedies were proposed to ameliorate the above mentioned difficulties. An algorithm proposed in [7] corrects template drift by the use of robust weights that are based on evidence, accumulated over many frames. Although such an algorithm is able to cope with template drift, it can still fail due to fast motion of the face or sudden appearance changes. In order to cope with drift accumulation as well as fast motion of the face we propose an algorithm that combines face tracking and face detection. The face finder permits face re-detection as well as reduces the accumulation of the drift over time. The tracker and detector are combined elegantly using swarm intelligence. Moreover, face detections are used to support the re-diversification of the particle swarm in the course of the tracking. Such candidate solutions vote simultaneously towards real location of the face through correcting the fitness function. Thanks to synergistic combination of different techniques [8], the hybrid algorithm is better. In particular, it has better recovery capability in case of tracking failure.

Convolutional neural network [9][10] integrates feature extraction and classification into single structure. It extracts two-dimensional features at increasing scales, and by comparison to relevant techniques it is relatively tolerant to local geometric distortions in the image. Thanks to such generalization capability it is used in our face tracking algorithm. The face tracker is based on adaptive appearance models and particle swarm optimization [11].

2 Visual Appearance Modeling Using Adaptive Models

Our intensity-based appearance model consists of three components, namely, the W -component expressing the two-frame variations, the S -component characterizing the stable structure within all previous observations and F -component representing a fixed initial template. The model $A_t = \{W_t, S_t, F_t\}$ represents the appearances existing in all observations up to time $t - 1$. It is a mixture of Gaussians [6] with centers $\{\mu_t^{(l)}\}_{l=w,s,f}$, their corresponding variances $\{(\sigma_t^{(l)})^2\}_{l=w,s,f}$ and mixing probabilities $\{m_t^{(l)}\}_{l=w,s,f}$.

Let $I(z, t)$ denote the brightness value at the position $z = (x, y)$ in an image \mathcal{I} that was acquired at time t . Let \mathcal{R} be a set of J locations $\{z(j)\}_{j=1}^J$ defining a template. $Y_t(\mathcal{R})$ is a vector of the brightness values at locations $z(j)$ in the template. The fitness score has the following form:

$$f(x_t) = \prod_{j=1}^J \sum_{l=w,s,f} \frac{m_t^{(l)}(j)}{\sqrt{2\pi(\sigma_t^{(l)}(j))^2}} \exp \left[-\frac{1}{2} \left(\frac{Y_t(j) - \mu_t^{(l)}(j)}{\sigma_t^{(l)}(j)} \right)^2 \right]. \quad (1)$$

In the objective function we utilize a recursively updated appearance model, which depicts stable structures seen so far, two-frame variations as well as initial object appearance. The update of the current appearance model A_t to A_{t+1} is done using the Expectation Maximization (EM) algorithm [12][11].

3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based algorithm that exploits a set of particles representing potential solutions of the optimization task [13]. The particles fly through the n -dimensional problem space with a velocity subject to both stochastic and deterministic update rules. The algorithm seeks for the global best solution through adjusting at each time step the location of each individual according to personal best and the global best positions of particles in the entire swarm. Each particle i keeps the position $p^{(i)}$ in the problem space, which is associated with the best fitness it has achieved personally so far. Additionally, when a particle considers all the population as its topological neighbors, each particle employs g location, which has been obtained so far by any particle in the swarm. The new positions are subsequently scored by a fitness function f . The velocity of each particle i is updated in accordance with the following equation:

$$v^{(i)} \leftarrow wv^{(i)} + c_1 r_1(p^{(i)} - x^{(i)}) + c_2 r_2(g - x^{(i)}) . \quad (2)$$

where $v^{(i)}$ is the velocity of the i th particle, c_1, c_2 denote the acceleration coefficients, r_1 and r_2 are uniquely generated random numbers in the interval [0.0, 1.0]. The new position of a particle is calculated in the following manner:

$$x^{(i)} \leftarrow x^{(i)} + v^{(i)} . \quad (3)$$

The local best position of each particle is updated as follows:

$$p^{(i)} \leftarrow \begin{cases} x^{(i)}, & \text{if } f(x^{(i)}) < f(p^{(i)}) \\ p^{(i)}, & \text{otherwise} \end{cases} . \quad (4)$$

and the global best position g is defined as:

$$g \leftarrow \arg \min_{p^{(i)}} \{f(p^{(i)})\} . \quad (5)$$

The value of velocity $v^{(i)}$ should be restricted to the range $[-v_{max}, v_{max}]$ to prevent particles from moving out of the search range.

At the beginning of the optimization the PSO initializes randomly the locations as well as the velocities of the particles. Then the algorithm selects $p^{(i)}$ and g values. Afterwards, equations (2)-(5) are called until maximum iterations or minimum error criteria is attained. After that, given $\hat{x}_t = g$ the algorithm calculates \hat{Y}_t . Then the algorithm uses it to update the model.

In the simplest solution the object tracking can be realized as deterministic searching of window location, whose content best matches a reference window

content. PSO allows us to avoid such time consuming exhaustive searching for the best match. It provides optimal or sub-optimal match without the complete knowledge of the searching space. In PSO based tracking, at the beginning of each frame, an initial position is assigned to each particle:

$$x_t^{(i)} \leftarrow \mathcal{N}(g_{t-1}, \Sigma) . \quad (6)$$

given the location g_{t-1} that has been estimated in the previous frame $t - 1$.

4 Convolutional Neural Networks

A convolutional neural network [9] is a special kind of a feedforward neural network. It incorporates prior knowledge about the input signal and its distortions into its architecture. Convolutional neural networks (CNNs) are specifically designed to cope with the variability of 2D shapes to be recognized. They combine local feature fields and shared weights as well as utilize spatial subsampling to ensure some level of shift, scale and deformation invariance. Using the local receptive fields the neurons can extract simple visual features such as corners, end-points. These elementary features are then linked by the succeeding layers to detect more complicated features.

A typical CNN contains a set of layers each of which consists of one or more planes. Each unit in the plane is connected to a local neighborhood in the previous layer. The unit can be seen as a local feature detector whose activation characteristic is determined in the learning stage. The outputs of such a set of units constitute a feature map. The units in a feature map are constrained to perform the same operation on different parts of the input image or previous feature maps, extracting different features from the same image. A feature map can be obtained in a sequential manner through scanning the input image by a single unit with weights forming a local receptive field and storing the outputs of this unit in corresponding locations in the feature map. This operation is equivalent to a convolution with a small kernel. The feature map can be treated as a plane of units that share weights. The subsampling layers, see Fig. 1, which usually follow layers with local, convolutional feature maps introduce a certain level of invariance to distortions and translations. Features of decreasing spatial resolution and of increasing complexity as well as globality are detected by the units in the successive layers.

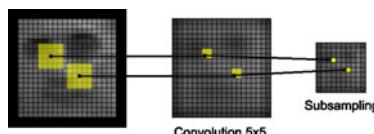


Fig. 1. Convolution and subsampling process in convolutional neural networks

The convolutional neural network we use consists of 7 layers, see Fig. 2. Layer C1 performs a convolution on gray images. The weights in the convolution mask are shared by all the neurons of the same feature map. The receptive fields of neighboring units overlap. The size of the scanning windows was chosen to be 24x24 pixels. The size of the mask is 5x5 and the size of the feature map of this layer is 20×20 . The layer has 156 trainable parameters. Layer S1 is the averaging/subsampling layer. It consists of 6 planes of size 20 by 20. Each unit in one of these planes receives four inputs from the corresponding plane in C1. Receptive fields do not overlap and all the weights are equal within a single unit. Therefore, this layer performs a local averaging and 2 to 1 subsampling. The number of trainable parameters utilized in this layer is 12. Once a feature has been extracted through the first two layers its accurate location in the image is less substantial and spatial relations with other features are more relevant. Therefore layers S1 and C2 are partially connected, and the task of such a configuration is to discover the relationships between different features.

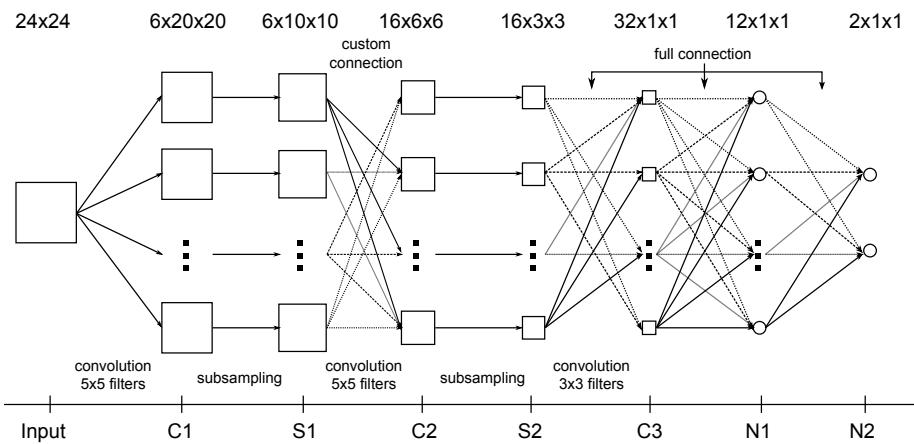


Fig. 2. Convolutional neural network for face detection

Layer C2 is composed of 16 feature maps. Each unit contains one or two receptive fields of size 5×5 , which operate at identical positions within each S1 maps. The first eight feature maps use single receptive fields. They form two independent groups of units responsible for distinguishing between face and non-face patterns. The remaining eight feature maps take inputs from every contiguous subsets of two feature maps in S1. This layer has 416 free parameters. Layer S2 plays the same role as the layer S1. It is constructed of 16 feature maps and has 32 free parameters. The next layer C3 is the convolution layer, which consists of 32 neurons and has 320 free parameters. The next fully connected layer consists of 12 neurons and has 396 trainable parameters. Finally, the output layer has two nodes that are fully connected to all the nodes from the previous layer. The network contains many connections but relatively few free trained

parameters. Weight sharing allows us to reduce considerably the number of free parameters and improves the generalization capability.

The face detector was trained on 3000 non-face patches collected from about 1500 images and 1500 frontal faces covering out-of-plane rotation in the range $-20^\circ, \dots, 20^\circ$. All faces were manually aligned by eyes position. For each face example the synthesized faces were randomly generated by in-plane rotation in the range $-10^\circ, \dots, 10^\circ$, random scaling about $\pm 10\%$, random shifting up to ± 1 pixel and mirroring. All faces were then cropped and re-scaled to windows of size 20×20 pixels while preserving their aspect ratio. The training collection contains also images acquired from our video cameras. To provide more false examples we utilized a training with bootstrapping [14]. By using bootstrapping we iteratively gathered examples, which were close to the boundaries of face and non-face clusters in the early stages of training. The activation function in the network was a hyperbolic tangent.

5 The Algorithm

In the particle swarm optimization algorithm each particle in the population represents a candidate solution to the optimization problem. Much of the success of PSO algorithms arises from the tendency of the individual particles to deviate from the best known position in any given iteration, enabling them to neglect local optima, while the swarm as a whole gravitates towards the global extremum. In a dynamic optimization problem the aim is not only to seek the extrema, but also to follow their progression through the space as closely as possible. Since the face tracking is a dynamic optimization problem, the tracking can be accomplished by means of incorporating the temporal continuity information into the conventional PSO algorithm. In consequence, the tracking can be achieved by a sequence of static PSO-based optimizations to seek the best object location, followed by re-diversification of the particles using (6) to cover the possible object state in the next time step.

Our tracking algorithm estimates the location of the face and the size of the template. In the second image in upper row of Fig. 3 we can see the locations of the particles after re-diversification. As we can see in this illustrative example, despite covering by the swarm almost the whole face, the tracker lost the target, see last image in the upper row of Fig. 3.

In order to reduce the computation overhead of the face detection algorithm we extract skin like patches, see 1st image in the bottom row of Fig. 3. Through the histogram back-projection we calculate the skin probability images, which are then thresholded to extract the skin like patches. The 2D histogram of the skin color is constructed in rg color space and consists of 8×8 bins. The skin patches are then refined using morphological closeing. Afterwards, the connected components labeling takes place. Using the extracted connected components we calculate areas and then remove small components. The neural network is then executed only on such refined patches, which likely contain faces. Encountered detections are passed to a non-maxima suppression algorithm, which removes possible multiple detections of the same face.

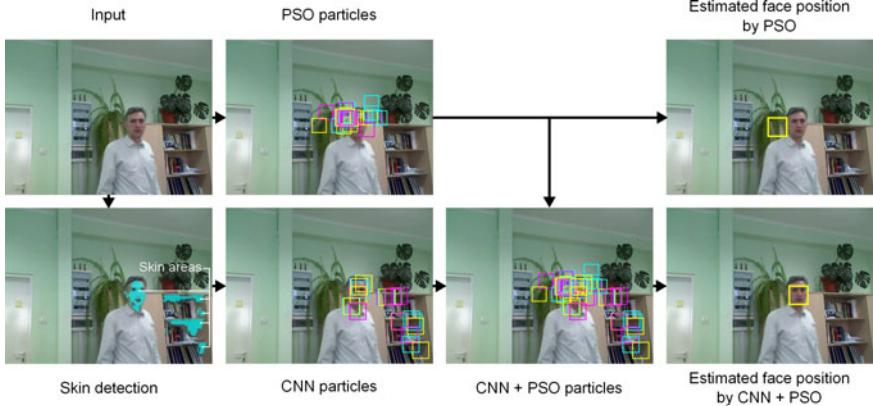


Fig. 3. The main stages of image processing in the hybrid algorithm

At the beginning of each frame, using the probabilities generated by softmax function $h(\tilde{x})$, where \tilde{x} denotes the location determined by the face detector, we select 32 best face candidates. Given the estimate of the face location in the previous frame we perform re-diversification using locations generated by the face detector, see 2nd line in the pseudo-code as well as 3rd image in bottom row of Fig. 3, and locations obtained by (6), see 3rd line in the pseudo-code. In line #10, the algorithm selects the nearest face candidate for the currently considered particle i as well as a face candidate that is closest to g_t . The term $[1 - \mathcal{N}(x_t^{(i,k+1)} - \tilde{x}_t^{(j*)}, \Sigma_d)h(\tilde{x}_t^{(j*)})]$, see code in line #11, allows the algorithm to prefer the particles, which are situated near the detected faces. In turn, the term $[1 - \mathcal{N}(x_t^{(i,k+1)} - \tilde{x}_t^{(l*)}, \Sigma_d)h(\tilde{x}_t^{(l*)})]$ allows the algorithm to prefer such a global particle, which has in the proximity a detected face. The w_1 and w_2 are

-
1. initialize $v_t^{(i,0)}$, detect faces, determine $\{\tilde{x}_t^{(i)}\}_{i=1}^{32}$
 2. $\{x_t^{(i,0)} = \tilde{x}_t^{(i)}, \hat{f}_t^{(i)} = (1 - h(\tilde{x}_t^{(i)}))\}_{i=1}^{32}$
 3. $\{x_t^{(i,0)} \sim \mathcal{N}(g_{t-1}, \Sigma_0), \hat{f}_t^{(i)} = f(x_t^{(i,0)})\}_{i=33}^N$
 4. $\{p_t^{(i)} = x_t^{(i,0)}\}_{i=1}^N$
 5. $i^* = \arg \min_i \hat{f}_t^{(i)}, \quad g_t = p_t^{(i^*)}, \quad \hat{f}_t^{(g)} = \hat{f}_t^{(i^*)}$
 6. For $k = 0, 1, \dots, K$
 7. For each particle i
 8. $v_t^{(i,k+1)} = w v_t^{(i,k)} + c_1 r_1 (p_t^{(i)} - x_t^{(i,k)}) + c_2 r_2 (g_t - x_t^{(i,k)})$
 9. $x_t^{(i,k+1)} = x_t^{(i,k)} + v_t^{(i,k+1)}$
 10. $j^* = \arg \min_j d(x_t^{(i,k+1)}, \tilde{x}_t^{(j)}), \quad l^* = \arg \min_l d(g_t, \tilde{x}_t^{(l)})$
 11. $f_c^{(i,k)} = w_1 f(x_t^{(i,k+1)}) + w_2^{(k)} \left[1 - \mathcal{N}(x_t^{(i,k+1)} - \tilde{x}_t^{(j*)}, \Sigma_d)h(\tilde{x}_t^{(j*)}) \right] + (1 - w_1 - w_2^{(k)}) \left[1 - \mathcal{N}(x_t^{(i,k+1)} - \tilde{x}_t^{(l*)}, \Sigma_d)h(\tilde{x}_t^{(l*)}) \right]$
 12. If $f_c^{(i,k)} < \hat{f}_t^{(i)}$ then $p_t^{(i)} = x_t^{(i,k+1)}, \quad \hat{f}_t^{(i)} = f_c^{(i,k)}$
 13. If $\hat{f}_t^{(i)} < \hat{f}_t^{(g)}$ then $g_t = p_t^{(i)}, \quad \hat{f}_t^{(g)} = \hat{f}_t^{(i)}$
-

weighting factors. The weighting factor w_2 depends on k and is used to balance the influence of the discussed terms. This way, the tracker and the detector are combined within the particle swarm optimization algorithm.

6 Experimental Results

The algorithm has been tested on real images that were acquired by a laptop built in camera. In Fig. 4 we can see some selected images that were utilized in the experiments. In the upper row we can see the results that were generated by adaptive appearance based tracker. The tracking was done on images acquired at rate 24 fps. In the images from middle row of Fig. 4 we can see the results, which were generated by the same algorithm, but using every third image of the input sequence. As we can observe, due to faster motion of the person in such an image sequence, the algorithm fails in frame #240. The tracking recovered in frame #360, see Fig. 4 and the algorithm continued successfully the tracking of the person. In the images from bottom row of the Fig. 4 we can see the experimental results, which were obtained by our hybrid algorithm. As we can observe, the tracker is capable of tracking the person despite fast motion as well as wooden bookshelf in the background.

The demonstrated above results were obtained by PSO consisting of $N = 128$ particles in $K = 5$ iterations. The weighting coefficient w_1 has been set to 0.6, and $w_2^{(k)}$ assumed the values 0.4, 0.3, ..., 0.0. The average error of the template location on the sequence depicted in Fig. 4 (on every third frame, until the tracking is lost) is equal to 11 pixels. Such error arose due to fast motion of the person as well as noisy images acquired by a laptop camera. The average error of the hybrid algorithm is about 5 pixels. Future work will concentrate on



Fig. 4. Face tracking by adaptive appearance models (images in upper row), by adaptive appearance models on every third frame of the input sequence (middle row), and by hybrid algorithm on every 3rd frame (bottom row)

implementation of the complete algorithm on GPU. We intend to use our GPU implementation of the adaptive appearance based tracker [II].

7 Conclusions

This paper proposes a hybrid algorithm consisting of face tracking and face detection modules. In the first module we employ particle swarm optimization and the adaptive appearance models. In order to increase the resistance of the tracking module to the drift we employ in the second module the convolutional neural network. The face detections support the re-diversification of the swarm in the course of the tracking. Such candidate solutions vote also simultaneously towards true location of the face through correcting the fitness function. In particular, the hybrid algorithm has better recovery capability in case of unsuccessful tracking.

References

1. Karlsson, S., Taj, M., Cavallaro, A.: Detection and tracking of humans and faces. *J. Image Video Process.* 2008, 1–9 (2008)
2. Tao, H., Lopez, R., Huang, T.: Tracking facial features using probabilistic network. In: *IEEE Int. Conf. on Aut. Face and Gesture Recognition*, pp. 166–170 (1999)
3. Cristinacce, D., Cootes, T.F.: Facial feature detection and tracking with automatic template selection. In: *Proc. of the 7th Int. Conf. on Automatic Face and Gesture Recognition*, pp. 429–434 (2006)
4. McKenna, S., Gong, S.: Tracking faces. In: *Int. Conf. on Automated Face and Gesture Recognition*, pp. 271–276 (1996)
5. Kwolek, B.: Face tracking using color, elliptical shape features and a detection cascade of boosted classifiers in particle filter. In: *Int. Conf. on Computer Vision and Graphics. Computational Vision and Graphics* 32, pp. 287–292. Springer, Heidelberg (2004)
6. Jepson, A.D., Fleet, D.J., El-Maraghi, T.: Robust on-line appearance models for visual tracking. *IEEE Trans. on PAMI* 25, 1296–1311 (2003)
7. Schreiber, D.: Robust template tracking with drift correction. *Pattern Recognition Letters* 28, 1483–1491 (2007)
8. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. *Information Sciences* 180, 2633–2634 (2010)
9. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time-series. In: Arbib, M. (ed.) *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge (1995)
10. Garcia, C., Delakis, M.: Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Tran. on PAMI* 26, 1408–1423 (2004)
11. Rymut, B., Kwolek, B.: GPU-supported object tracking using adaptive appearance models and particle swarm optimization. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) *ICCVG 2010. LNCS*, vol. 6375, pp. 227–234. Springer, Heidelberg (2010)
12. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society. Ser. B* 39, 1–38 (1977)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proc. of IEEE Int. Conf. on Neural Networks*, pp. 1942–1948. IEEE Press, Piscataway (1995)
14. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. *IEEE Tran. on Pattern Analysis and Machine Intelligence* 20, 23–38 (1998)

Genetic Selection of Subgraphs for Automatic Reasoning in Design Systems

Barbara Strug

Department of Physics, Astronomy and Applied Computer Science,
Jagiellonian University, Reymonta 4, Krakow, Poland
`barbara.strug@uj.edu.pl`

Abstract. In this paper a hybrid artificial intelligence approach consisting of a genetic algorithm combined with frequent subgraphs mining and its use in a design system is presented. The design system uses hypergraphs as the internal representation. The frequent graphs mining approach is then used to find common elements in designs while genetic search is performed to reduce the size of the found set of frequent patterns. The application of this method to automatic evaluation of designs with some experimental results are also presented.

1 Introduction

Designing is an important process. The human design process, whether computer aided or traditional, is usually an iterative one consisting of several steps. Firstly a preliminary or conceptual design is created. Then it is analysed or tested in order to find out which of its components must be redesigned or refined. The process of evaluation and optimisation is repeated until an acceptable solution is found. The longer the process the more expensive it is. Hence any method able to speed up this process, and thus lower its costs, is highly desirable.

A number of methods for speeding the generation of designs has been developed, such as generative approach, in which graph grammars were used to generate new designs [16, 8, 3, 9, 15], and grammar systems [5, 17]. Other methods used to generate graphs representing designs include evolutionary computations used in different domains of design [3, 10, 15].

Graphs are used because of their usefulness as the way of representing complex objects in different domains of computer science [16]. In this paper hypergraphs [11], in which multi-argument relations between elements can be expressed, are used. While in ordinary graphs edges connect only two nodes, hypergraphs are composed of a set of nodes and a collection of hyperedges with different numbers of ordered *tentacles*, each of them linked to a node. Hyperedges of hypergraphs used in this paper represent both design components and multi-argument relations among them.

All generation methods result in producing a number of graphs representing designs. The main problem lies in the complexity and size (understood as the number of graphs) of such database. It is difficult to automatically evaluate the

quality of the graphs (the quality of graph is here understood as the quality of the design a graph represents as a solutions to a given design problem). Thus the process of evaluation usually requires a human designer who can choose the best solution or give some numerical values to each design. The problem is that for a human "evaluator" all graphs have to be rendered to designs. This process in turn requires visualizing designed objects. Unless the design problem is very simple the rendering of designs may be very complex and time-consuming.

A possible method of eliminating the visualization step, and thus speeding the evaluation step of designing, could be one using a number of graphs representing designs, for which a human "evaluator" has defined a quality value, as a basis for evaluating other designs in the same design problem. As it can be noticed that the designs getting higher quality values usually have some common elements, finding frequent substructures in graphs seems a useful approach. As the number of frequent subgraphs tend to be large a method of reducing its size is also proposed. Such a combination of pattern mining and genetic optimization can be considered a hybrid artificial intelligence approach [14, 17, 21].

The use of methods based on frequent graph mining techniques combined with evolutionary optimization of the size of the set of frequent patterns proposed in this paper is tested on the examples from the domain of floor layout design [18]. This task consists in generating a set of floor layouts within a given floor size but with different room arrangements.

2 Hypergraphs-Based Design Representation

Different types of graphs have been researched and used in this domain. Here an extension of ordinary graphs - known as hypergraphs is used.

Hypergraphs (HG) consist of nodes and edges, but they differ from standard graphs because the edges can connect more than two nodes, thus allowing for more complex relations to be represented. A hyperedge in a hypergraph may represent a geometrical object or a relation. A hyperedge representing an



Fig. 1. Hypergraph representing a new design

object is called object hyperedge, while the one representing a relation is called a relational hyperedge.

Nodes and hyperedges in hypergraphs can be labelled and attributed. Labels are assigned to nodes and hyperedges by means of node and hyperedge labelling functions, respectively, and attributes - by node and hyperedge attributing functions. Attributes represent properties (for example size, position, colour or material) of a component or relation represented by a given hyperedge. In case of floor layout design example the object hyperedges are labelled with the names of rooms (like bedroom, kitchen) and relational ones by relations like adjacency or accessibility. An example of such a hypergraph is depicted in fig. 2. In all figures object hyperedges are depicted as rectangles and relational ones - as ovals.

3 Frequent Graph Mining in Design System

Frequent pattern mining was first proposed by Agrawal et al. [2]. Since then various kinds of extensions and applications, from scalable data mining methodologies, to handling a wide diversity of data types and a huge number of applications were developed [23][22][12]. In graph mining a subgraph is considered frequent if its support, i.e. the number of graphs that contain this subgraph, is larger than some predefined threshold. The support is usually expressed as a percentage of graphs containing a given subgraph. In this paper two algorithms, namely FFSM and gSpan, are considered. Both algorithms can work on undirected graphs with labelled nodes and edges [23][22].

The automatic evaluation of designs based on frequent subgraph mining starts with a database of hypergraphs coded in the GraphML format and imported to the GraphSearcher application [20]. Then the application generates the set of frequent patterns and these patterns are used to evaluate new designs. A number of experiments was carried for the floor layout design task with the support parameter set to 100%, 90%, and 60%, respectively. In table 1 the number of frequent patterns found for these support parameters is presented.

The frequent patterns found with the support set to 100% were used as the reasoning rules and tested against a number of new designs. Some of the rules are depicted in fig. 2a and b. One of the hypergraphs representing new designs is presented in fig. 2. Then the designs were evaluated on the basis of the number of frequent patterns present in their hypergraph representations in such a way that



Fig. 2. Examples of frequent patterns representing design requirements

Table 1. The number of frequent patterns for different values of support for FFSM and gSpan algorithms

algorithm	FFSM	FFSM	FFSM	gSpan	gSpan	gSpan
support	100%	90%	60%	100%	90%	60%
num. of freq.	910	1775	19075	1021	1808	19104

those which contain the highest number of frequent substructures are considered to be the best.

The results obtained with the use of the FFSM method were presented in [18], and the results obtained with the use of the gSpan algorithms and a comparison of both results were presented in [19].

However the number of frequent subgraphs is very high. As the evaluation of a hypergraph representing a new design consists in checking how many of these frequent subgraphs are also subgraphs of the new hypergraph the evaluation requires a huge number of subgraph isomorphism checking operations.

Thus combining the frequent patterns mining with genetic algorithm, used as a way of selecting a reduced set of patterns, is proposed.

4 Genetic Algorithms

Genetic algorithms are well established as a search and optimization method. They are based on natural evolution. Instead of one solution at a time a larger subset of the search space, known as population, is considered. Then genetic search consists in evaluating and refining possible solutions. For the purpose of the evaluation a domain specific fitness function is used. After the evaluation the population is transformed by genetic operators, crossover and mutation. The better individuals, i.e. having higher value of the fitness function, are more likely to contribute to the next population.

In evolutionary methods transformation is often performed not on actual solutions (in this paper - sets of frequent patterns) but on their coded equivalents. In genetic algorithms a binary coding is used most often in which each solution is transformed to a binary string, modified by genetic operators, then decoded and evaluated. [6][10][13][14].

4.1 Genetic Selection of Subgraphs

Let $P = \{P_1, \dots, P_{N_P}\}$ - be a set of patterns found by the frequent pattern mining algorithm used to calculate the scores for new designs and N_P be the number of these patterns (subgraphs). Let G be the family of hypergraphs evaluated by the set P with N_G being the number of these hypergraphs. Let $EV(G_i)$ be the quality value for the i -th hypergraph from G , such that $0 < EV(G_i) < 1$, i.e. real value instead of percentage is used. Let GP be a matching matrix, such that:

$$GP_{ij} (i = 1, \dots, N_G, j = 1, \dots, N_P) = \begin{cases} 0 : P_j \text{ not a subgraph of } G_i, \\ 1 : P_j \text{ subgraph of } G_i. \end{cases} \quad (1)$$

Thus for each hypergraph its quality score equals to the number of frequent patterns it contains divided by the total number of frequent patterns. Using the values from the matching matrix the score can be expressed by the following equation:

$$EV(G_i) = \frac{\sum_{j=1}^{N_P} GP_{ij}}{N_P}. \quad (2)$$

Let N be the size of the population. The population consists of binary strings, called individuals, of length N_P . Thus the i -th element of the population represents a reduced set of subgraphs denoted P_{B_i} . The j -th position of a the i -th individual contains 1 if the j -th frequent pattern belongs to the reduced set represented by this string. The i -th individual has thus the form:

$$B_i = (b_1^i, \dots, b_{N_P}^i), \text{ where } b_j^i = \begin{cases} 0 : P_j \notin P_{B_i}, \\ 1 : P_j \in P_{B_i}. \end{cases} \quad (3)$$

The size of the reduced set P_{B_i} will be denoted by $|P_{B_i}|$ ($|P_{B_i}| = \sum_{j=1}^{N_P} b_j^i$).

The aim of the genetic search is to find the smallest set P_{B_i} preserving the quality of evaluation of designs.

4.2 Fitness Function

The score of the i -th hypergraph obtained from the set induced by the k -th individual, denoted $Sc_i B_k$, is calculated as the number of frequent patterns belonging the the i -th hypergraph divided by the size of the induced set. The number of frequent patterns belonging to the i -th hypergraph is computed by using the logical *AND* operation on an individual and the i -th row of the matching matrix GP . This score can be expresses by the following equation:

$$Sc_i(B_k) = \frac{\sum_{j=1}^{N_P} (b_j AND GP_{ij})}{|P_{B_k}|}. \quad (4)$$

The fitness function is defined as the error of the evaluation. It is calculated as the difference between the actual score (obtained by the unreduced set) and the score obtained by the set induced by the individual. The error is firstly computed for one hypergraph by the following equation:

$$F^i(B_k) = EV(G_i) - Sc_i(B_k). \quad (5)$$

Then to calculate the fitness of the individual B_k the error is squared and averaged for all hypergraphs. Thus

$$F(B_k) = \frac{\sum_{i=1}^{N_G} (F^i(B_k))^2}{N_G}. \quad (6)$$

and after substituting the equation for the error on a single hypergraph the fitness function has the following form:

$$F(B_k) = \frac{\sum_{i=1}^{N_G} (EV(G_i) - Sc_i(B_k))^2}{N_G}. \quad (7)$$

4.3 Results

The genetic search process was used on the set of frequent subgraphs found for the floor layout design problem. The set found by the gSpan with support set to 100% (containing 1021 frequent graphs) was used as the starting point (as presented in table ①). Firstly, all hypergraphs were evaluated by the full set of subgraphs and these scores were used for comparison during fitness testing of each individual.

The first population was generated randomly with uniform distribution. The population consisted of 100 individuals. Standard genetic operators, i.e. one point crossover and mutation, were used to generate the next population. The probabilities for crossover and mutation were set, after several experiments, to 0.63 and 0.03, respectively. Elitism was also used to preserve the best individual found during the evolution process. The process was run for 1000 generations. Table ② presents results obtained after 20 runs of the genetic optimization. This table shows the average fitness of the 100th, 500th and 1000th generation (averaged over all 20 experiments), respectively, the average size of the set induced by the individuals in these generations (rounded). The best fitness in the given generation and the size of the set induced by the best individual are also presented. It can be observed that the average size of the sets induced gets smaller during the genetic search what suggests that some subgraphs are not essential for correct evaluation. An interesting parameter to observe during the optimization process is the fitness of the individual inducing the smallest set of graphs. At the beginning of the process a very small set was found but its fitness was unacceptable and it was soon eliminated during the optimization process. But the smallest set found in the last population in one of the experiments (inducing a set of 155 graphs) has a fitness worse by 0.05 from the best individual while having inducing 92 graphs less. This observation suggests that including the size of the set induced by a given individual in the fitness function could possibly lead to finding a smaller set with fitness similar (i.e. not much worse) to the best one found in current experiments.

Fig. ③ depicts a graph of the changes of the fitness of the whole population and fitness of the best individual in the experiment which produced the best solution. It can be noticed that the use of elitism strategy actually prevented losing the best individual. The best individual found by the genetic algorithm induced a subset of frequent subgraphs containing 247 subgraphs. Thus it shows

Table 2. The averages and best individuals over 20 experiments

	100	500	1000
average fitness	0.72	0.45	0.22
average size	538	479	380
best fitness	0.63	0.31	0.18
size of best individual	378	410	247
smallest set size	115	351	155
fitness of the smallest sized individual	0.81	0.40	0.23

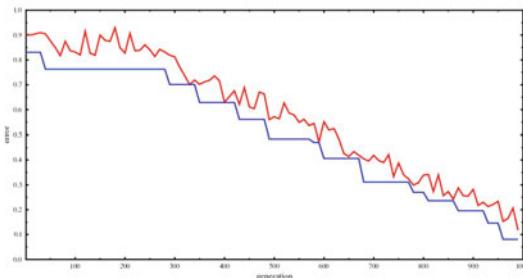


Fig. 3. Graph showing the change of the fitness (error) of the population and of the best individual

a considerable reduction against the starting set. At the same time the error is relatively small.

5 Concluding Remarks and Future Work

In this paper a method of reducing the sets of frequent subgraphs in hypergraphs used to represent designs is presented. This method was tested on set of hypergraphs representing floor layouts. The frequent patterns found were then used to evaluate the quality of new designs. The genetic optimization allowed for reducing the set to a much more manageable one. It makes the evaluation of new design much faster - as a smaller number of subgraph isomorphism tests must be carried out while still preserving the quality of the evaluation.

As the aim of this method is to significantly lower the number of subgraphs used to evaluate a design one possible place for improvement is changing the fitness function in such a way that it would prefer individuals inducing smaller sets even if their error is slightly higher. But more experiments are planned to test how exactly the size should contribute to the fitness function.

The possibility of using other than one-point crossover operator is also planned. And as the probabilities of mutation and crossover were constant during the evolutionary process a meta evolution is another direction for future research.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72(13-15), 2729–2730 (2009)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proc. 1993 ACM-SIGMOD, Washington, DC, pp. 207–216 (1993)
3. Borkowski, A., Grabska, E., Nikodem, P., Strug, B.: Searching for Innovative Structural Layouts by Means of Graph Grammars and Evolutionary Optimization. In: Proc. 2nd ISEC Conf., Rome, pp. 475–480 (2003)
4. Corchado, E., Abraham, A., de Carvalho, A.C.P.L.F.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)

5. Csuha-J-Varjú, E., Dassow, J., Kelemen, J.: Grammar systems. A grammatical approach to distribution and cooperation. *Topics in Computer Mathematics* 8 (1994)
6. De Jong, K., Arciszewski, T., Vyas, H.: An Overview of Evolutionary Computation and its Applications. In: AIE 1999, Warsaw, pp. 9–22 (1999)
7. Derrac, J., García, S., Herrera, F.: A first study on the use of coevolutionary algorithms for instance and feature selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
8. Grabska, E.: Graphs and designing. In: Ehrig, H., Schneider, H.-J. (eds.) Dagstuhl Seminar 1993. LNCS, vol. 776, Springer, Heidelberg (1994)
9. Grabska, E., Nikodem, P., Strug, B.: Evolutionary Methods and Graph Grammars in Design and Optimization of Skeletal Structures Weimar. In: 11th ICE 2004, Weimar (2004)
10. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
11. Habel, A., Kreowski, H.J.: Some structural aspects of hypergraph languages generated by hyperedge replacement. In: Brandenburg, F.J., Wirsing, M., Vidal-Naquet, G. (eds.) STACS 1987. LNCS, vol. 247, pp. 207–219. Springer, Heidelberg (1987)
12. Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation: A Frequent-pattern Tree Approach. *Data Mining and Knowledge Discovery* 8(1), 53–87 (2004)
13. Holland, J.H.: Adaptation in Natural and Artificial Systems, Ann. Arbor. Univ. of Michigan Press (1975)
14. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolutionary Programs. Springer, New York (1996)
15. Nikodem, P., Strug, B.: Graph Transformations in Evolutionary Design. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 456–461. Springer, Heidelberg (2004)
16. Rozenberg, G.: Handbook of Graph Grammars and Computing by Graph. Transformations 99, 1–3 (1997)
17. Simeoni, M., Staniszkis, M.: Cooperating graph grammar systems. In: Grammatical Models of Multi-Agent Systems, pp. 193–217. Gordon and Breach, Amsterdam (1999)
18. Strug, B., Ślusarczyk, G.: Reasoning about designs through frequent patterns mining. *Advanced Engineering Informatics* 23, 361–369 (2009)
19. Strug, B., Ślusarczyk, G.: Frequent Pattern Mining in a Design Supporting System, Key Engineering Materials. Trans. Tech. Pub. 450, 1–4 (2011)
20. Tomanek, M.: Searching for graph patterns and applications, MSc thesis, Jagiellonian University, (2009) in polish
21. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants – Evolutionary and Neural Methods of Training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 590–597. Springer, Heidelberg (2010)
22. Yan, X., Yu, P.S., Han, J.: Substructure Similarity Search in Graph Databases. In: SIGMOD 2005, pp. 766–777 (2005)
23. Yan, X., Yu, P.S., Han, J.: Graph Indexing: A Frequent Structure-based Approach. In: SIGMOD 2004, pp. 335–346 (2004)

Controlling the Prediction Accuracy by Adjusting the Abstraction Levels

Tomasz Łukaszewski, Joanna Józefowska, Agnieszka Lawrynowicz,
Łukasz Józefowski, and Andrzej Lisiecki

Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60-965 Poznań, Poland

{tomasz.lukaszewski, joanna.jozefowska, agnieszka.lawrynowicz,
lukasz.jozefowski, andrzej.lisiecki}@cs.put.poznan.pl

Abstract. The predictive accuracy of classifiers is determined among others by the quality of data. This important property of data is strongly affected by such factors as the number of erroneous or missing attributes present in the dataset. In this paper we show how those factors can be handled by introducing the levels of abstraction in data definition. Our approach is especially valuable in cases where giving the precise value of an attribute is impossible for a number of reasons as for example lack of time or knowledge. Furthermore, we show that increasing the level of precision for an attribute significantly increase predictive accuracy, especially when it is done for the attribute with high information gain.

Keywords: naïve Bayes, prediction accuracy, attribute value ontology.

1 Introduction

Data Mining is an analytic process designed to explore large amounts of data in search of patterns and/or relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The concept of Data Mining is becoming increasingly popular as large amounts of data become available in data repositories. Recently, increased interest in developing new data mining techniques has been observed, specifically combining components of several general approaches [48]. Hybrid intelligent techniques coming from different computational intelligence areas become popular because in many cases such combinations perform better than the individual techniques [13]. The proposed approach attempts to build a hybrid framework in which both inductive and deductive reasoning methods are involved in order to improve prediction accuracy of data mining with noisy data.

The *prediction accuracy* of classifiers is determined by two factors: the inductive bias of the learning algorithm and the quality of training data. Given a specific learning algorithm it is obvious that its prediction accuracy depends on the quality of training data. This quality is determined by two factors: the selection of attributes and errors introduced into a dataset [10]. Two types of errors occurring in a dataset are distinguished: erroneous or missing attribute values

and misclassified objects. Non-systematic errors of this kind are called *noise* [7]. According to [6] there are three major sources of noise: (i) insufficiency of the description of training data, (ii) corruption of attribute values in training data, (iii) erroneous classification of training data. Let us call the noise following from this insufficiency *description noise*.

Following [2] the main source for description noise may be the language used to represent the attribute values, which is not expressive enough to model different *levels of abstraction*. In such a case, erroneous or missing attribute values may be introduced by users that are required to provide very specific values, but the level of their knowledge of the domain is too general to precisely describe the observation by the appropriate value of an attribute. Even if the person is an expert in the domain, erroneous or missing attribute values can be introduced as a consequence of lack of time or other resources to precisely describe the observation by the appropriate value of an attribute.

Observe that if the language enabled modeling different levels of abstraction (*precise descriptions* and *imprecise descriptions*), we would be able to reduce the number of erroneous or missing attribute values. However, decreasing the precision of the description may decrease the prediction accuracy as well. We can improve the precision of the description performing additional diagnostic tests. Let us notice, that usually there is a shortage of time or other resources to precisely describe the observation for *each* attribute.

The goal of the paper is to present a naïve Bayesian classifier generalized to the levels of abstractions. In the experiment we show for some datasets, that increasing the precision of the description of observations give the most effective increase in terms of the prediction accuracy for attributes with the highest information gain.

2 Description Noise Handling by AVO

The precise and imprecise descriptions of examples should reflect the domain knowledge. Let us notice, that in some domains, hierarchical relationships between *concepts* may be observed and this knowledge could be explored. Such knowledge is often available in the form of *ontologies*. Thus, the precise and imprecise descriptions for a given attribute A , are represented by a hierarchical structure, called the *attribute value ontology* (AVO). We assume that precise descriptions (specific values) are represented by *primitive* concepts while imprecise descriptions are represented by *abstract* concepts.

Example 1. Let us consider the following medical problem. In order to determine the correct treatment, an agent that caused the infection needs to be specified. Specific values of this attribute are the following *Streptococcus*, *E.Coli*, *Salmonella*, *Fungi*, *Virus*. An AVO describing the domain of infectious agents is presented in Fig. 11. Description noise is handled by a hierarchy of primitive and abstract concepts. Primitive concepts are the following: *Streptococcus*, *E.Coli*, *Salmonella*, *Fungi*, *Virus*. Abstract concepts are the following: *Infectious Agent*, *Bacteria*, *Gram-positive Bacteria*, *Gram-negative Bacteria*.

Let us observe that Streptococcus is not the only Gram-positive Bacteria in the real world and our hierarchy, for some reasons, does not contain concepts of the other Gram-positive Bacteria. Therefore, the concept Gram-positive Bacteria should be correctly interpreted as: Streptococcus or other Gram-positive Bacteria. Similarly, the concept Gram-negative Bacteria should be interpreted as: E.Coli or Salmonella or other Gram-negative Bacteria. It is easy to observe that we can not represent any Gram-positive Bacteria other than Streptococcus using the specific values only. This example illustrates the fact that a language containing the specific values only suffers from the description noise.

Given is an attribute A and set $V = \{v_1, v_2, \dots, v_n\}$, $n > 1$, of specific values of this attribute. Let us assume that given is an ontology, which represents domain knowledge. In particular, it expresses a multilevel subsumption ("is-a") hierarchy of concepts representing the precise and imprecise descriptions. We define an attribute value ontology (AVO) \mathcal{A} as follows:

Definition 1. An attribute value ontology (AVO) \mathcal{A} is a directed acyclic graph $\langle C, R \rangle$, where: C is a set of concepts (primitive and abstract ones), R is a subsumption relation over C , subset $C^P \subseteq C$ of nodes without predecessors is a finite set of primitive concepts of \mathcal{A} .

Further in this paper we use an AVO with the following properties: \mathcal{A} is an *in-tree*, each concept $c_i \in C$ represents a *non-empty subset* of V , the root represents set V and each primitive concept $c_i \in C^P$ represents a value $v_i \in V$. For each pair of concepts $(c_i, c_j) \in R$ where c_i represents $V_i \subseteq V$ and c_j represents $V_j \subseteq V$ we have $V_i \subseteq V_j$.

Assume that given is a set of n attribute value ontologies $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$. An (training or testing) example E is represented by a vector (c_1, c_2, \dots, c_n) , where c_i is a primitive or an abstract concept. We associate the example E to concept c_i for each \mathcal{A}_i . Let us observe that the association of (training and testing) examples to concepts of \mathcal{A}_i has the following property:

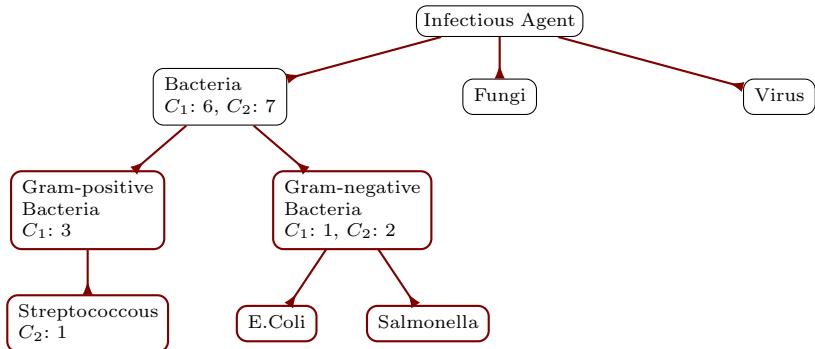
Property 1. For a given testing example represented by a concept c_i of \mathcal{A}_i each training example described *equally or more precisely* by concepts of \mathcal{A}_i , is associated to the concept c_i or its subconcept.

Let us consider the medical problem presented in Example 1. In order to determine the correct treatment (C_1 or C_2), an agent that caused the infection needs to be specified. The training data with primitive and abstract concepts is given in Tab. II. For the simplicity we consider the classification problem with only one AVO: Infectious Agent. Thus, each training example is described using only a concept from this AVO.

The association of training examples to AVO is presented in Fig. II. For each concept we present the number of associated training examples. For example, the concept *Bacteria* is used by 6 training examples with class label C_1 and 7 training examples with class label C_2 .

Table 1. A medical diagnosis training data

Number of examples	Infectious Agent	Treatment
6	Bacteria	C_1
3	Gram-positive Bacteria	C_1
1	Gram-negative Bacteria	C_1
7	Bacteria	C_2
1	Streptococcus	C_2
2	Gram-negative Bacteria	C_2

**Fig. 1.** An example of AVO with associated training examples

3 Extending Naïve Bayesian Classifier to AVO

Assume that given is a set of n attributes A_1, A_2, \dots, A_n . An (training or testing) example is represented by a vector (v_1, v_2, \dots, v_n) , where v_i is the specific value of A_i . Let C represent the class variable and C_j represent the value it takes (a class label). The naïve Bayesian classifies a testing example E by selecting the class C_j with the largest posterior probability $P(C_j|E)$, as indicated below:

$$P(C_j|v_1, v_2, \dots, v_n) \propto P(C_j) \prod_{1 \leq i \leq n} P(v_i|C_j) . \quad (1)$$

where $P(v_i|C_j)$ is the probability of an example of class C_j having the observed attribute A_i value v_i . The probabilities in the above formula must be estimated from training examples, e.g. using relative frequency:

$$P(C_j) = \frac{n_j}{n} \quad P(v_i|C_j) = \frac{n_{ij}}{n_j} . \quad (2)$$

where n is the number of training examples, n_j is the number of training examples with class label C_j , n_{ij} is the number of training examples with the value of the attribute $A_i = v_i$ and class label C_j .

In the approach with AVO, the naïve Bayesian classifier needs to be generalized to estimate the value $P(c_i|C_j)$, where c_i is a primitive or abstract concept of \mathcal{A}_i . In order to estimate this probability, e.g. by relative frequency, we use Property 1 and we have:

$$P(c_i|C_j) = \frac{\sum_{c_k \in \{c_i\} \cup desc(c_i)} n_{kj}}{n_j}. \quad (3)$$

where n_j is the number of training examples with class label C_j , n_{kj} is the number of training examples with the value of the attribute $\mathcal{A}_i = c_k$ and class label C_j , $desc(c_i)$ is the set of concepts that are descendants of the concept c_i in \mathcal{A}_i . Similar approach (AVT-NBL) was presented in [9]. However, AVT-NBL is based on a taxonomy of attribute values and makes a so called *cut* through this taxonomy in order to get a more compact classifier. In our approach, we use ontologies, that are managed without any compression and the primary goal is to improve the predictive accuracy.

Let us consider our medical problem presented in Fig. II. The number of all examples is equal to 20. Each class is described exactly by the same number of examples, therefore the prior probability that one will observe class C_j is equal to 0.5 for C_1 and C_2 . Let us consider the following scenario: there is a patient and the diagnosis is *Bacteria*. We estimate the posterior probability $P(C_j|Bacteria)$. Therefore we concentrate on these examples, that are associated to the concept *Bacteria* or his descendants. From (3), (2) and (1) we have: $P(Bacteria|C_1) = \frac{6+3+1}{10} = 1$, $P(Bacteria|C_2) = \frac{7+2+1}{10} = 1$, $P(C_1|Bacteria) \propto 0.5 * 1 = 0.5$, and $P(C_2|Bacteria) \propto 0.5 * 1 = 0.5$. As we can see, both classes are still equally probable. Therefore, we need to conduct a medical diagnosis test to know what kind of *Bacteria* is the *Infectious Agent*: *Gram-positive* or *Gram-negative*. The result of the test indicated that *Gram-positive Bacteria* (shortly *GpBacteria*) is the *Infectious Agent*. Taking into account this information we again estimate the posterior probabilities: $P(GpBacteria|C_1) = \frac{3}{10} = 0.3$, $P(GpBacteria|C_2) = \frac{1}{10} = 0.1$, $P(C_1|GpBacteria) \propto 0.5 * 0.3 = 0.15$, and $P(C_2|GpBacteria) \propto 0.5 * 0.1 = 0.05$. As we can see, class C_1 is three times more probable than class C_2 and the treatment represented by class C_1 is recommended.

4 Experiments and Results

The goal of the performed computational experiment was to estimate, which AVO would give the most effective increase in terms of the prediction accuracy by reducing the level of abstraction of the description of examples. We analyzed each AVO independently. Moreover, two paradigms were considered: reducing the level of abstraction in testing examples and reducing the level of abstraction in training examples. For the simplicity of the experiment, we transformed each original attribute into an AVO that consisted of necessary primitive concepts and one abstract concept only: the root.

Assume that given is a set of n attribute value ontologies $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$. A *precisely described* example E is represented by a vector (c_1, c_2, \dots, c_n) , where every c_i is a primitive concept. For an analyzed \mathcal{A}_i we increased the level of abstraction in description of examples as follows: we modified the primitive concept c_i in \mathcal{A}_i into the root of \mathcal{A}_i . In this way we modified some number of instances (either testing or training instances) in order to estimate the predictive accuracy. We conducted an empirical study on some data sets from UCI Machine Learning Repository [5]. Basic information on these tests is given in Tab. 2. All these data sets have only categorical attributes.

A naïve Bayesian classifier was implemented for these experiments. We treat missing values ‘?’ as a separate specific value. The average prediction accuracy and 95% confidence levels using the t-test were obtained using 10 runs of 10-fold cross-validation. Taking into account future goals of our research we decided to implement these statistical functions as a part of our setting. In order to avoid the zero-frequency problem, we implemented the Laplace estimation and m -estimation. The average prediction accuracy and 95% confidence levels obtained for original datasets are presented in Tab. 2. We have chosen the m -estimation approach for the experiment because this estimation performed better.

The estimated prediction accuracy and 95% confidence levels obtained in the experiment are presented in Tab. 3. The column headed Attr. gives the numbers of the first four attributes with the highest information gain. The column headed I.G. gives the value of information gain for the attribute. The next columns give the average predictive accuracy and 95% confidence levels achieved for different percentage of imprecise described examples for the attribute. For the rest of attributes the average predictive accuracy was comparable or higher than the predictive accuracy of the forth presented attribute. Some exceptions were found, e.g for the audiology dataset: attributes with the information gain equal to 0.570 (7. place) and 0.214 (12. place) have the average predictive accuracy lower than the average predictive accuracy of the forth presented attribute.

It follows from the results presented in Tab. 3 that imprecise description reduces the prediction accuracy. The prediction accuracy decreases with the growth of the number of modified examples and is more significant for attributes with higher information gain. Some differences may be observed regarding the modification paradigm. In case where the test instances were modified, the prediction accuracy decreases slowly with the growth of the number of modified examples.

Table 2. Benchmark datasets used in the experiments

Dataset	Instances	Attributes	Classes	Laplace	m -estimate
Audiology	226	69	24	63.05 ± 6.44	77.47 ± 6.54
Breast Cancer	286	9	2	71.87 ± 5.81	70.80 ± 4.64
Car Evaluation	1728	6	4	85.06 ± 2.34	85.93 ± 2.50
Mushroom	8124	22	2	95.43 ± 0.54	99.10 ± 0.20
Nursery	12960	8	5	90.22 ± 0.75	90.25 ± 0.78

Table 3. Results obtained for imprecisely described examples

Dataset	Attr.	I.G.	Percentage of imprecise testing examples				
			20%	40%	60%	80%	100%
Audiology	1	0.902	73.57±6.13	69.48±6.37	66.95±5.30	64.77±5.06	62.14±4.96
	66	0.867	77.01±7.20	76.56±7.62	76.10±7.50	75.19±8.05	74.29±8.19
	59	0.647	77.92±6.81	77.92±6.81	77.11±6.04	77.11±6.04	77.56±5.93
	6	0.633	76.66±6.13	76.20±5.98	75.94±4.89	75.03±4.67	75.49±4.66
Breast	6	0.077	71.51±4.30	71.16±4.74	70.44±5.90	69.37±5.79	69.31±6.40
	4	0.069	70.38±5.12	68.95±6.51	69.60±7.04	70.67±7.10	70.97±7.72
	3	0.057	71.51±3.91	71.87±3.79	71.51±4.83	72.23±5.42	72.16±5.81
	5	0.053	71.09±4.86	69.73±5.94	70.74±5.82	70.74±5.82	71.45±5.70
Car	6	0.262	82.16±2.44	78.74±2.99	75.79±3.30	71.40±3.04	68.90±2.92
	4	0.219	83.19±2.71	80.18±3.62	77.75±3.42	73.64±3.35	71.56±3.01
	1	0.096	84.94±2.25	84.13±2.29	83.67±2.26	82.51±2.02	82.28±1.69
	2	0.074	84.89±2.60	84.19±2.88	83.78±2.62	82.85±1.77	82.27±1.80
Mushroom	5	0.906	97.10±0.44	95.22±0.35	93.62±0.47	91.96±0.59	90.40±0.82
	20	0.481	99.13±0.20	99.08±0.15	99.02±0.16	99.00±0.21	98.97±0.22
	9	0.417	98.97±0.21	98.87±0.16	98.78±0.18	98.60±0.23	98.49±0.28
	19	0.318	99.13±0.22	99.10±0.19	99.09±0.19	99.13±0.21	99.18±0.19
Nursery	8	0.958	82.46±0.85	75.05±0.79	67.83±0.87	60.32±1.01	53.18±0.94
	2	0.196	87.67±0.67	85.40±0.61	82.75±0.80	80.57±1.05	77.99±1.09
	1	0.073	89.00±0.78	87.79±0.73	86.74±0.75	85.46±0.92	84.34±0.84
	7	0.022	90.07±0.70	89.90±0.69	89.69±0.76	89.44±0.75	89.22±0.79
			Percentage of imprecise training examples				
			20%	40%	60%	80%	100%
Audiology	1	0.902	76.56±7.09	76.10±7.67	77.01±7.38	76.75±5.63	63.51±6.14
	66	0.867	76.56±7.09	77.01±7.38	77.01±7.55	75.39±8.22	74.84±7.24
	59	0.647	77.01±7.20	76.10±7.67	77.47±7.64	77.56±6.56	78.02±6.43
	6	0.633	76.66±6.74	76.20±8.03	76.20±7.87	76.75±5.40	76.10±6.37
Breast	6	0.077	71.51±4.83	73.24±4.98	72.46±6.17	72.82±5.92	65.21±6.84
	4	0.069	70.74±5.23	71.16±4.57	70.80±5.73	70.44±6.17	64.56±7.72
	3	0.057	71.39±5.75	71.74±5.95	71.09±5.18	68.95±6.63	70.38±5.99
	5	0.053	71.74±5.68	72.16±4.91	72.16±5.38	71.74±5.95	63.97±6.77
Car	6	0.262	85.05±2.35	85.11±2.53	85.28±2.57	84.00±2.92	60.20±3.24
	4	0.219	85.46±2.15	85.52±3.11	85.87±2.38	84.64±2.29	65.86±1.74
	1	0.096	85.11±2.26	85.81±2.65	85.91±2.66	84.87±2.23	70.95±1.93
	2	0.074	84.82±2.56	84.70±2.77	85.11±2.67	85.04±2.37	73.32±2.66
Mushroom	5	0.906	99.11±0.19	99.13±0.23	98.99±0.19	98.62±0.20	90.46±0.78
	20	0.481	99.10±0.20	99.09±0.19	99.05±0.21	99.05±0.21	98.95±0.23
	9	0.417	99.10±0.20	99.10±0.20	99.10±0.20	99.08±0.19	98.50±0.27
	19	0.318	99.10±0.20	99.10±0.20	99.10±0.20	99.10±0.20	99.15±0.22
Nursery	8	0.958	90.24±0.76	90.23±0.75	90.15±0.76	90.35±0.85	49.07±0.78
	2	0.196	90.39±0.86	90.36±0.86	90.23±0.84	90.27±0.75	74.07±1.12
	1	0.073	90.21±0.79	90.25±0.76	90.05±0.75	90.06±0.81	80.79±1.17
	7	0.022	90.24±0.75	90.18±0.77	90.11±0.76	90.18±0.75	85.96±1.07

In case of modification of the training instances, any significant decline of the prediction accuracy may be observed only for very large modifications.

5 Conclusions

Introducing AVO we allow users to provide more general descriptions (abstract concepts) at different levels of abstraction, instead of requiring to provide very specific values (primitive concepts). Firstly, it allows to avoid introducing erroneous or missing attribute values. Secondly, it allows to classify an example for the current *state of knowledge* using these abstract concepts. However, using abstract concepts instead of primitive ones should have an influence on the prediction accuracy. In our experiment we have shown that the choice of the attribute where we increase the precision of the description (e.g., by performing additional tests) is significant from the point of view of the prediction accuracy. The results show that attributes with high information gain should be described precisely. In further research we plan to examine the properties of AVO increasing the prediction accuracy such as semantics or structure of the ontology.

Acknowledgment. This research was supported by the Polish Ministry of Science and Higher Education Research Grant No. N N516 186437.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72, 2729–2730 (2009)
2. Clark, P., Niblett, T.: Induction in Noisy Domains. In: 2nd European Working Session on Learning, pp. 11–30. Sigma Press, Wilmslow (1987)
3. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. *Information Science* 180, 2633–2634 (2010)
4. Derrac, J., Garca, S., Herrera, F.: A First Study on the Use of Coevolutionary Algorithms for Instance and Feature Selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
5. Frank, A., Asuncion, A.: UCI Machine Learning Repository,
<http://archive.ics.uci.edu/ml>
6. Hickey, R.J.: Noise Modelling and Evaluating Learning from Examples. *Artif. Intell.* 81, 157–179 (1996)
7. Quinlan, J.R.: Induction of Decision Trees. *Machine Learning* 1, 81–106 (1986)
8. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants – Evolutionary and Neural Methods of Training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 590–597. Springer, Heidelberg (2010)
9. Zhang, J., Kang, D.K., Silvescu, A., Honavar, V.: Learning accurate and concise naïve Bayes classifiers from attribute value taxonomies and data. *Knowl. Inf. Syst.* 9, 157–179 (2006)
10. Zhu, X., Wu, X.: Class Noise vs. Attribute Noise: A Quantitative Study. *Artif. Intell. Rev.* 22, 177–210 (2004)

Delta Analysis: A Hybrid Quantitative Approach for Measuring Discrepancies between Business Process Models

Eren Esgin¹ and Pinar Senkul²

¹ Middle East Technical University, Informatics Institute,

² Middle East Technical University, Computer Engineering Department,
065531 Ankara, Turkey

eesgin@ii.metu.edu.tr, senkul@ceng.metu.edu.tr

Abstract. Business process management (BPM) continues to play a significant role in today's highly globalized world. In order to detect and prevent the gap between reference process model and the actual operation, process mining techniques discover operational model on the basis of the process logs. An important issue at BPM is to measure the similarity between the reference process model and discovered process model so that it can be possible to pinpoint *where* process participants deviate from the intended process description. In this paper, a hybrid quantitative approach is presented to measure the similarity between the process models. The proposed similarity metric is based on a hybrid process mining technique that makes use of genetic algorithms. The proposed approach itself is also a hybrid model that considers process activity dependencies and process structure.

Keywords: Business Process Management (BPM), Delta Analysis, Process Mining, Process Modeling, Similarity Measurement.

1 Introduction

While contemporary information systems are intensively utilized in the enterprises, their leverage effect in automating business processes is limited by the difficulties faced in the process design phase [1]. Basic problems are resulting from the discrepancies between process design and process enactment [2,3]. Process design is influenced by personal perceptions, e.g. reference process models are often normative in the sense that reflect what *should* be done rather than describing the actual case [4]. As a result, proposed reference models tend to be rather incomplete, subjective and at a too coarse-grained level.

Process mining is proposed as a remedy to handle these discrepancies by distilling significant patterns from the event logs and discovering the business process model automatically [1,5,6,7,8,9]. Unlike to traditional *design-centric* approach, process mining is not biased and restrictive by normative perceptions [10]. Once the operational process model is discovered, important information to improve the process design is *where* process participants deviate from the intended process definition.

In BPM, this activity is known as *delta analysis*, i.e. comparing the actual process, represented by a process model constructed through process mining, with *prescriptive* reference process model [11]. This work proposes a method for measuring the similarity between process models. The proposed technique is based on a hybrid process discovery technique that makes use of genetic algorithms [12]. The proposed technique is also a hybrid approach that focuses on process activity dependencies and process structure to identify distance measure between business processes.

This paper is organized as follows: Section 2 includes related work. Section 3 highlights the hybrid process discovery approach on which the proposed technique is built. Section 4 introduces the design of the proposed quantitative approach for delta analysis. Section 5 gives the experimental results based on human similarity judgment and comparison with prior similarity metric. Section 6 presents the concluding remarks.

2 Related Work

One of the basic works on delta analysis is given in [13]. In this work, the equivalence of process models is verified on the basis of equivalence of event logs. In [13], author does not discover a graphical process model but use such logs to check for deviation from a prescribed business process model.

Cook and Wolf [7] present an approach for delta analysis in the context of software processes. They use AI algorithms to compare process models and compute the level of correspondence.

Behavioral semantics can lead to performance problems due to large sets of traces and state explosion. In [14], an approximation on behavioral semantics is given by the concept of causal footprint. In this study, process models are represented as points in Euclidian distance space. Hence the underlying problem is reduced to the nearest-neighbor problem.

Another way of defining behavioral semantic is monitoring the states in which the process model can be. This idea is realized in [15] by taking the state-space of two process models and checking if they can simulate one another. By counting such states, we can measure how dissimilar two process models are.

Another perspective in delta analysis is using the *graph theory*. In [16], Bunke has shown that with generic graphs, the maximum common sub-graph (MCS) is equivalent to computing the graph edit-distance emphasized in [17]. This MCS is the baseline to measure the common activities and transitions of workflow processes in [18].

3 Underlying Process Discovery Approach

The proposed process similarity measurement approach is based on the hybrid process discovery technique given in [12]. This technique takes its roots from the process discovery based on using *from-to chart*, which is proposed in [19]. In this approach, from-to chart, which is fundamentally used in tracing material handling routes between operations, machines, departments or work centers on the production floor [20], is used for analyzing the event logs.

This approach is further improved in [12] by several enhancements. In [19], the rearrangement operation is performed by a permutative approach, which leads to an exponential increase in processing time. Hence in [12], the runtime complexity of the approach is improved by adopting *Genetic Algorithms* (GA).

4 Proposed Approach

Comparing the behavior of processes using traditional methods is not sufficient due to various reasons as given in [14]. These methods aim to measure (dis)similarity on the basis of evaluating the stages (subsections) of the underlying business process instead of considering the business process in a holistic manner. In this work, a hybrid quantitative approach in measuring the similarities between business process models is proposed such that; *dependency/frequency graph*, which is finite-state machine like block-model representation, is converted to *vector models*. The major difference of the proposed vector model from the standard vector model is the term weight assignment such that; the terms that reflect structural feature are atomic and the terms that reflect behavioral feature are represented as set. As an example, two dependency/frequency graphs are given in Figure 1 for the following steps.

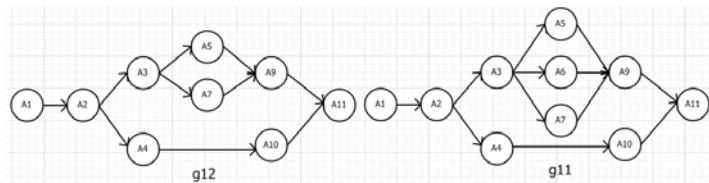


Fig. 1. Sample Dependency/frequency Graphs, g_{12} and g_{11}

4.1 Construction of Vector Models

The proposed similarity measurement for two business process models is built on *vector models* and *process triple*. It is considered that process triples of two processes $PT_1 = \langle N_1, S_{1,succ}, S_{1,pred} \rangle$ and $PT_2 = \langle N_2, S_{2,succ}, S_{2,pred} \rangle$ are the major input for calculation.

Definition 1 (Process triple, PT). Process triple $PT = \langle N_x, S_{x,succ}, S_{x,pred} \rangle$ is a bag of parameters, where:

- i. N_x is a finite set of nodes (activities) take place at dependency/frequency graph g ,
- ii. $S_{x,pred} \in (N_x, P(N_x))$ is a global set representing predecessor list of each node at N_x ,
- iii. $S_{x,succ} \in (N_x, P(N_x))$ is a global set representing successive list of each node at N_x .

The *document collection* consists of two process triples, which are the basic subjects for comparison. The first and second process triples are represented as $PT_1 = \langle N_1, S_{1,succ}, S_{1,pred} \rangle$ and $PT_2 = \langle N_2, S_{2,succ}, S_{2,pred} \rangle$.

The *set of terms* is the *union* over nodes, successive and predecessor lists of selected process triples. It is defined $\beta = N_1 \cup S_{1,succ} \cup S_{1,pred} \cup N_2 \cup S_{2,succ} \cup S_{2,pred}$ as the set of terms and there exists an indexing function $f: \beta \rightarrow \{1, 2, \dots, |\beta|\}$ that assigns an index to each term at the vector.

The proposed similarity measurement reflects both *structural* (e.g. the size of the graph) and *behavioral* (e.g. interaction between the activities) perspective. Consequently the *weight* of each term is determined as follows:

- a. Nodes reflect whether indexed activity exists at the underlying process model. Hence the weight for this type terms is *atomic* ($p_{i,f(\partial)} = 1$ or 0).
- b. Successive or predecessor list holds the neighboring activities of indexed activity at the underlying process model. The weight of this term is dependent to context of the term at candidate process model. Hence the weight is represented as a set.

Definition 2 (Process vector, pv). Let $PT_1 = \langle N_1, S_{1,succ}, S_{1,pred} \rangle$ and $PT_2 = \langle N_2, S_{2,succ}, S_{2,pred} \rangle$ be two process triples, with β the set of terms and $f: \beta \rightarrow \{1, 2, \dots, |\beta|\}$ an indexing function. It is defined two process vectors $\overrightarrow{pv}_1 = (p_{1,1}, p_{1,2}, \dots, p_{1,|\beta|})$ and $\overrightarrow{pv}_2 = (p_{2,1}, p_{2,2}, \dots, p_{2,|\beta|})$ for each element $\partial \in \beta$ and for each process triple $i, j \in \{1, 2\}$ holds that:

$$p_{i,f(\partial)} = \begin{cases} 1 & \text{if } f(\partial) \leq |N_i \cup N_j| \text{ and } \partial \in N_i \\ 0 & \text{if } f(\partial) \leq |N_i \cup N_j| \text{ and } \partial \in N_j \\ \{n_1, n_2, \dots, n_n\} & \text{if } |N_i \cup N_j| < f(\partial) \leq |S_{i,succ} \cup S_{i,pred} \cup S_{j,succ} \cup S_{j,pred}|. \end{cases} \quad (1)$$

4.2 Calculation of Process Similarity

First of all, the difference between weights of the same indexed term, $\omega(f(\partial))$, at process vectors $\overrightarrow{pv}_1 = (p_{1,1}, p_{1,2}, \dots, p_{1,|\beta|})$ and $\overrightarrow{pv}_2 = (p_{2,1}, p_{2,2}, \dots, p_{2,|\beta|})$ is calculated as follows:

$$\omega(f(\partial)) = \begin{cases} |p_{1,f(\partial)} - p_{2,f(\partial)}| & \text{if } f(\partial) \leq |N_1 \cup N_2| \\ 1 - \frac{|p_{1,f(\partial)} \cap p_{2,f(\partial)}|}{|p_{1,f(\partial)} \cup p_{2,f(\partial)}|} & \text{if } |N_1 \cup N_2| < f(\partial) \leq |S_{1,succ} \cup S_{1,pred} \cup S_{2,succ} \cup S_{2,pred}|. \end{cases} \quad (2)$$

The second weight term stated above is based on *maximal common sub-graph* theorem stated in [16]. Finally, the distance between PT_1 and PT_2 , denoted by $dis(PT_1, PT_2)$, is calculated such that:

$$dis(PT_1, PT_2) = \sqrt{\frac{\sum_{\partial=1}^{|\beta|} \omega(f(\partial))^2}{|\beta|}} \in [0, 1]. \quad (3)$$

The value of $dis(PT_1, PT_2)$ ranges between 0 (equivalence) and 1 (no similarity). The complexity of the proposed approach is approximately $O(\beta^2)$, due to the search performed at $S_{x,succ}$ and $S_{x,pred}$ sets.

5 Experimental Results

5.1 Comparison with Human Judgments

The validation of the underlying similarity measurement approach is performed by comparing the similarity measurements of the proposed approach with the judgments of 30 process engineers working as SAP consultants at various modules. Basically, the human judgments are collected by a questionnaire, which consists of synthetic reference and candidate process models given in Figure 2. For each model comparison, the process engineers grade the similarity of the underlying comparison on a 0 to 1 Likert scale.

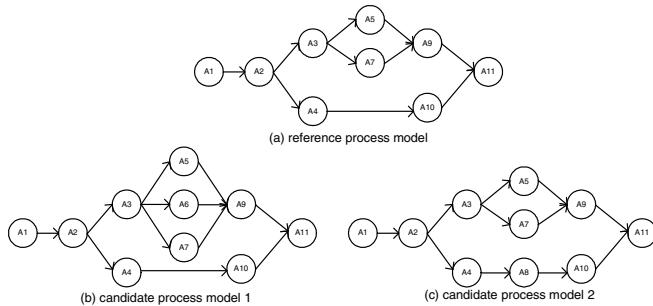
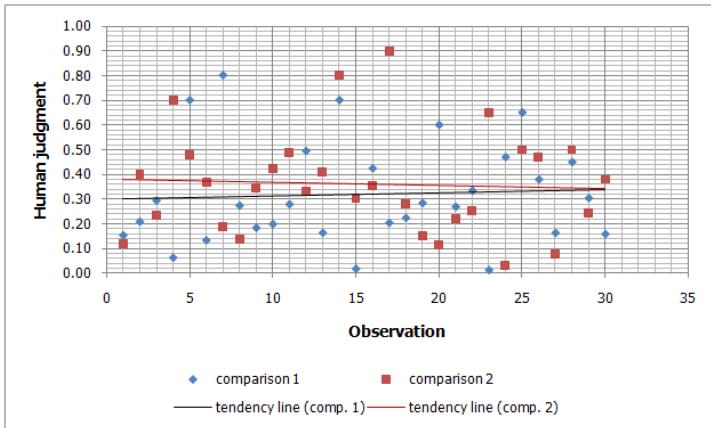


Fig. 2. Synthetic Process Models Used

Afterwards, the comparison between similarity measurement of the proposed approach and similarity judgments obtained from process engineers is performed by dependent t-tests. According to the t-values (0.186 and 1.201 versus $t_{0.05,29}$), the null hypothesis, H_0 , which states that there is no clear distinction between similarity measurement of the proposed approach and similarity judgments obtained from process engineers, is accepted. The results of t-tests are given in Table 1. Figure 3 shows the human similarity judgments as obtained from the questionnaire.

Table 1. t-Tests for Similarity Measurement Comparison

t-Test: Comparison one (reference PM versus candidate PM1)		t-Test: Comparison one (reference PM versus candidate PM1)			
	Proposed Approach	Human Judgment	Proposed Approach	Human Judgment	
Mean	0.328	0.321	Mean	0.408	0.362
Variance	0.000	0.043	Variance	0.000	0.044
Observations	30	30	Observations	30	30
Hypothesized Mean Difference	0		Hypothesized Mean Difference	0	
df	29		df	29	
t Stat	0.186		t Stat	1.201	
P(T<=t) one-tail	0.427		P(T<=t) one-tail	0.120	
T Critical one-tail	1.699		T Critical one-tail	1.699	
P(T<=t) two-tail	0.853		P(T<=t) two-tail	0.240	
T Critical two-tail	2.045		T Critical two-tail	2.045	

**Fig. 3.** Human Similarity Judgments

5.2 Comparison with Prior Similarity Metric

Another comparison is performed with the *dependency difference metric*, d , given in [21]. Dependency difference metric is defined by the trace of the delta matrix of normalized matrices NM_1 and NM_2 as follows:

$$\Delta NM = (NM_1 - NM_2). \quad (4)$$

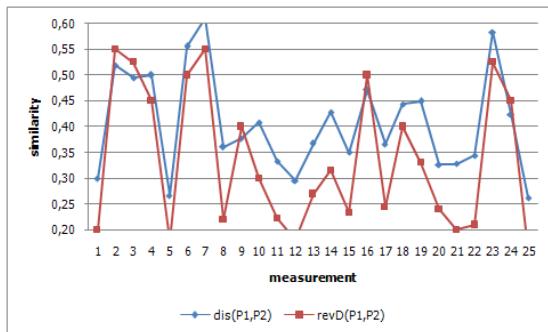
$$d(P_1, P_2) = \text{tr}[\Delta NM \times \Delta NM^T]. \quad (5)$$

where $\text{tr}[\cdot]$ denotes the *trace of matrix*, i.e. the sum of the diagonal values. Hence, multiplication of delta matrix with its transpose yields positioning the squared values along the diagonal and trace of matrix collects these squared values. Actually, this distance function counts the number of edge discrepancies between the process models (i.e. behavioral perspective).

The major shortcoming of the dependency difference metric is the width of range, since the maximal value is $|\Delta NM|^2$. Hence $d(P_1, P_2)$ can be divided by $|\Delta NM|^2$ in order to compare it with $\text{dis}(P_1, P_2)$. On the other hand, this revision at dependency difference metric may result in very small dissimilarity value for conservative cases (i.e. having little difference at behavioral dimension). As a result, dependency difference metric is revised as follows:

$$\text{rev}D(P_1, P_2) = \frac{d(P_1, P_2)}{|\Delta NM|}. \quad (6)$$

In the experiments, twenty five process models are evaluated by both $\text{rev}D(P_1, P_2)$ and $\text{dis}(P_1, P_2)$. Figure 4 shows the dissimilarity measurements obtained from process model evaluation.

**Fig. 4.** Process Model Evaluation by $revD(P_1, P_2)$ and $dis(P_1, P_2)$

Afterwards, the comparison of these two dissimilarity metrics is performed by dependent t-test. According to the t-value (2,166 versus $t_{0.05,43}$), the null hypothesis, H_0 , which states that there is no clear distinction between similarity measurement of $dis(P_1, P_2)$ and $revD(P_1, P_2)$, is rejected. This outcome can be interpreted such that; while $revD(P_1, P_2)$ (and also $d(P_1, P_2)$) tends to emphasize the behavioral perspective, proposed metric, $dis(P_1, P_2)$, reflects both structural and behavioral perspectives. The result of the underlying t-test is given in Table 2.

Table 2. t-Test for Dissimilarity Metric Comparison

t-Test: Dissimilarity Metric Comparison ($dis(P_1, P_2)$ versus $revD(P_1, P_2)$)		
	$dis(P_1, P_2)$	$revD(P_1, P_2)$
Mean	0.407	0.334
Variance	0.010	0.019
Observations	25	25
Hypothesized Mean Difference	0	
df	43	
t Stat	2.166	
$P(T \leq t)$ one-tail	0.018	
T Critical one-tail	1.681	
$P(T \leq t)$ two-tail	0.036	
T Critical two-tail	2.017	

6 Conclusion

In this work, a delta analysis method for process models is proposed with the aim of analyzing the gap between the reference and operational process models. The operational process model is discovered by using the hybrid process mining technique presented in [12]. Then discovered process model is compared with the reference process model by using the proposed distance metric, $dis(PT_1, PT_2)$, which is calculated on the basis of the vector model of the process and an abstraction of process behavior in the form of process triple. Hence, the proposed metric takes into account both the structural and behavioral perspectives.

According to experimental results, proposed dissimilarity metric successfully simulates the human assessment model and outperforms the dependency difference metric which relies only on the behavioral perspective. As the future work, the

tendency of process engineers towards evaluating structural and behavioral features of the process models can be interpreted and term weights can be distinctly graded by mimicking human assessment model.

References

- [1] van den Aalst, W.M.P., Gunther, C., Recker, J., Reichert, M.: Using Process Mining to Analyze and Improve Process Flexibility. In: 7th Workshop on BPMDS 2006, CAiSE 2006 Workshop (2006)
- [2] Mărușter, L., Weijters, A.J.M.M.T., van der Aalst, W.M.P., van den Bosch, A.: Process Mining: Discovering Direct Successors in Process Logs. In: Lange, S., Satoh, K., Smith, C.H. (eds.) DS 2002. LNCS, vol. 2534, pp. 364–373. Springer, Heidelberg (2002)
- [3] Weijters, A., van den Aalst, W.M.P.: Process Mining Discovering Workflow Models from Event-Based Data. In: Proc. of the 13th Belgium-Netherlands Conference on Artificial Intelligence, pp. 283–290 (2001)
- [4] van den Aalst, W.M.P., Dongen, B.F., Herbst, J.L.M., Schimm, G., Weijters, T.A.J.M.M.: Workflow Mining: A Survey of Issues and Approaches. *Data & Knowledge Engineering* 47(2), 237–267 (2003)
- [5] Gunther, C.W., van den Aalst, W.M.P.: Process Mining in Case Handling Systems. In: Proc. PRIMIUM Subconference at the Multikonferenz Wirtschaftsinformatik (2006)
- [6] Agrawal, R., Gunopulos, D., Leymann, F.: Mining Process Models from Workflow Logs. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, pp. 469–483. Springer, Heidelberg (1998)
- [7] Cook, J.E., Wolf, A.L.: Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology* 7(3), 215–249 (1998)
- [8] Weijters, A.J.M.M., van den Aalst, W.M.P.: Rediscovering Workflow Models from Event-Based Data Using Little Thumb. *Integrated Computer-Aided Engineering* 10(2), 151–162 (2003)
- [9] van den Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. *Transaction on Knowledge and Data Engineering* 16(9), 1128–1142 (2004)
- [10] van den Aalst, W.M.P., Dongen, B.F., Herbst, J.L.M., Schimm, G., Weijters, T.A.J.M.M.: Workflow Mining: A Survey of Issues and Approaches. *Data & Knowledge Engineering* 47(2), 237–267 (2003)
- [11] van den Aalst, W.M.P.: Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing. *Requirements Engineering* 10(3), 198–211 (2005)
- [12] Esgin, E., Senkul, P., Cimenbicer, C.: A Hybrid Approach for Process Mining: Using From-to Chart Arranged by Genetic Algorithms. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 178–186. Springer, Heidelberg (2010)
- [13] Kleiner, N.: Delta Analysis with Workflow Logs: Aligning Business Process Prescriptions and Their Reality. *Requirements Engineering* 10(3), 212–222 (2005)
- [14] van Dongen, B.F., Dijkman, R., Mendling, J.: Measuring Similarity between Business Process Models. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer, Heidelberg (2008)

- [15] Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S., Zave, P.: Matching and merging of statecharts specifications. In: Proc. of 29th ICSE, pp. 54–63. IEEE Computer Society, Los Alamitos (2007)
- [16] Bunke, H., Shearer, K.: A Graph Distance Metric Based on the Maximal Common Subgraph. *Pattern Recognition Letters* 19(3), 255–259 (1998)
- [17] Zhang, K., Shasha, D.: Simple Fast Algorithms for the Editing Distance between Trees and Related Problems. *SIAM Journal of Computing* 18(6), 1245–1262 (1989)
- [18] Huang, K., Zhou, Z., Han, Y., Li, G., Wang, J.: An Algorithm for Calculating Process Similarity to Cluster Open-Source Process Designs. In: Proc. of 4th Grid and Cooperative Computing, vol. 3252, pp. 107–114 (2004)
- [19] Esgin, E., Senkul, P.: Hybrid Approach to Process Mining: Finding Immediate Successors of a Process by Using From-to Chart. In: Int. Conf. on Machine Learning and Applications, pp. 664–668. IEEE Computer Society, Los Alamitos (2009)
- [20] Francis, R.L., McGinnis, L.F., White, J.A.: Facility Layout and Location: An Analytical Approach. Prentice Hall, New Jersey (1992)
- [21] Bae, J., Liu, L., Caverlee, J., Zhang, L., Bae, Z.: Process Mining and Integration using Distance Measures. *International Journal of Web Services Research* 1(4), 14–32 (2006)

Analysis of Face Gestures for Human-Computer Interaction

Jacek Rondio and Aleksandra Królak

Technical University of Lodz, Institute of Electronics, ul. Wólczańska 211/215,
PL 90-924 LODZ
jacek.rondio@gmail.com, aleksandra.krolak@p.lodz.pl

Abstract. Vision-based systems become more and more popular due to increasing computation speed of the computer and quality of cheap webcams available for home users. Such systems can be implemented as human-computer interfaces supporting people with alternative, hands free communication with computers. The paper presents a vision-based face gesture recognition system that can be used as face features analysis module or a human-computer interface. It enables analysis of the following face features: eyes, eyebrows, mouth, tongue and face speed. The overall accuracy of the system is equal to 98%.

Keywords: face detection, face features analysis, Human-Computer Interaction.

1 Introduction

Digital image processing is the branch of computer sciences since 1960, however reserved in the beginnings for the military applications. Nowadays, with increasing speed of computers and lowering cost of complete computer systems, vision-based applications can be alternative solutions for the human-computer interfacing employed by civil uses. Such systems can provide inestimable help for people with physical disabilities, especially ones who have no control over their limbs. The system recognizing face gestures such as blinks, mouth moves and other allows disabled people to control the computer and socialize them by connecting to the world.

Socializing handicapped people is one of the aims of today's information societies. Present society is the most computerized society in the history, while the presence of global network – The Internet changes the world in one global village. People can lead their life online without borders. Disabled people with help of modern technologies can lead an almost normal life, working without going out from their houses, meeting new people on social networking services and staying in touch with friends and family. Vision-based human-computer interaction systems can be used for a variety of purposes, such as entertainment, communication, education or rehabilitation.

The aim of the presented research was to design and implement an algorithm for advanced face features analysis that can be used in human-computer interaction. The developed solution should recognize as many face gestures as possible, providing high accuracy and low error rate, as well as compromise these features with the

computation speed. Designed system can be employed as face features analysis module in other program. It can assure communication between the program and the user in hands free manner.

The paper is organized as follows: the next section gives the overview of face gesture recognition solutions, the detailed description of the proposed system is given in Section 3. Experimental results are presented in Section 4 and Section 5 concludes the paper.

2 Previous Work

There exist many vision based system focusing on light independent face expression recognition. The main application of such system can be non intrusive biometric identification of people. Face expression recognition can be used also in the human-computer interfacing. Facial expressions are part of interpersonal communication, used voluntarily, but sometimes involuntarily to show people's emotions.

The problem of face expression recognition is a nontrivial task of image processing. Several systems were designed for automatic recognition of persons internal emotional state. The most interesting and promising attempt for this task is employing FACS (Facial Action Coding System) developed by Paul Ekman and Wallace Friesen in 1976 [1]. FACS is used by actors and animators to learn emotion expression, however it can be used in opposite way – to determine emotional state on the basis of face expression. Described Action Units can be detected using vision based recognition system and interpreted, due to FACS description, as a given emotion face expression [2].

In the existing vision-based facial features recognition system many different approaches are employed, such as optical flow [3] or SVM (Support Vector Machine) and PCA (Principal Component Analysis) [4]. One of the interesting vision-based systems is an Interface Framework to Drive an Intelligent Wheelchair Using Facial Expressions on University of Porto, Portugal [5]. The main idea of the presented system operation is counting occurrences of selected face features like eyebrows and mouth in determined face zones. Extracted information is an input to neural network that has to be trained for each user separately, what takes about 5 minutes.

Vision based systems can be used to create music effects, while playing the standard musical instrument. Such a system was created by Michael J. Lyons in Kyoto, Japan [6]. It uses miniature head mounted camera facing the mouth of the user and laptop computer to ensure mobility. Mouth shape changes are detected, recognized and later converted to MIDI commands controlling a synthesizer or musical effects device. To extract mouth width and height, statistical analysis is applied.

B-link is the open source program created by two Polish telecommunication companies Telekomunikacja Polska S.A. and Orange on the basis of [7]. It is able to distinguish short natural blinks, that keep eye wet, from intentional blinks that bring information to computer that given symbol is chosen by the user. For eye-blink detection Haar-like face detection and template matching methods are employed.

3 Vision-Based Face Gesture Recognition System

Designed system (called iViso) is based on personal computer and simple web camera. It recognizes and measures face, eyebrows, eyes, mouth and tongue and return calculated values for further processing. iViso does not have graphical user interface, it only supports the user with the webcam capture preview. The program also displays the debugging screen showing output of each image processing stage and values of calculated feature measurements.

The system was developed in Visual C++ development kit with usage of OpenCV 2.0 image processing library under the open source BSD license. For image capturing purpose system uses the web camera operating with resolution 640x480 and frame rate strongly dependent on computation speed and class of the processor.

The designed system contains several independent modules, responsible for different image processing stages. Image is passed through the stages, while the output is computed (Fig.1).

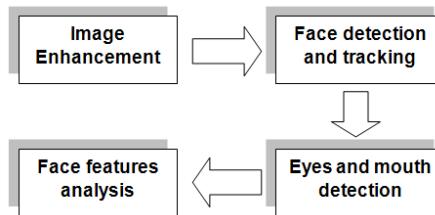


Fig. 1. Scheme of the vision-based face gesture recognition system

3.1 Image Enhancement

The image is resized to the half of the size in order to speed up the calculations.

In the designed system cheap web camera is used for the purpose of image capturing, what causes that quality of the picture obtained can be very low. There exist many artifacts on the image, caused inter alia by noise from the camera sensor. Also the environment conditions can differ for even different parts of one image. The main problem in image processing is proper lighting of the object that is seen by the camera. To get rid of such problems, designed system uses quality improvement stage of image processing that contains median filtering [8] – for noises discarding and logarithm filtering [9] that expands values of dark pixels.

3.2 Face Features Detection

Output of filters is processed by face detection function and if the face is present in the picture and it meets the minimal conditions (size) for required face image, the face bounding box is stored for further processing. Face recognition uses algorithm created by Viola and Jones [10] based on template matching – the most effective one, taking into account computation time and results of detection. Each Haar-like feature mask that is convolved with the image is scaled and moved through the image during computation. Value of each feature is an input of a simple decision tree classifier. In order

to maximize positive classification rate boosting process is employed. In proposed solution, default cascades for face detection, delivered with OpenCV library, were used.

On this stage speed of the found face is calculated, and next program decides, whether learning process should be continued or normal operation of the program can start. Due to computation costs of performing eyes and mouth detection using Haar classifiers, program estimate the placement of face features with respect to face bounding box using simple statistical analysis (mean value of features parameters) for only 10 frames, assuming that the user remains the same. In further frames, the tracking function is employed to minimize the computation time.

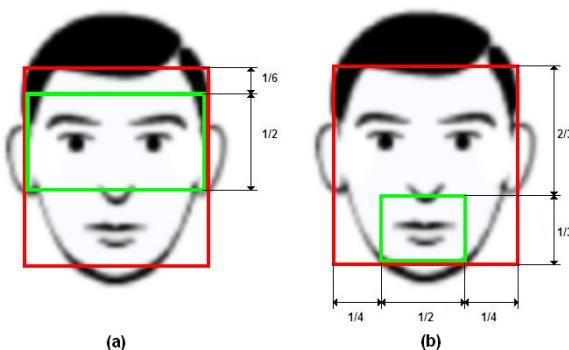


Fig. 2. Regions of interest for eyes and mouth detection

The algorithm for eyes detection is based, as the face detection module, on Haar-like features classifier however it uses different cascades to detect eyes. To increase the computation speed the system determines the region of interest on the face, where eyes are expected to be found (Fig. 2a). The mouth detection is also based on such an algorithm however region of interest is changed (Fig. 2b).

After the estimation of current face features position and sizes, functions responsible for the features analysis can be employed. Analysis functions are independent of each other, however tongue analysis function requires information about exact mouth size and position to the correct operation. Designed analysis procedure requires extraction of skin color from the face image. There exist many skin color extraction algorithms. However, due to using the logarithm transform for the light intensity enhancement all of these algorithms were difficult to implement in the designed system. After analysis of different solutions, skin color extraction based on histogram analysis was employed.

Histogram and cumulative histogram for two halves of the image is calculated, while it is quite often that each half of the face has different brightness and contrast. Having boundary value for 20% (found empirically) of darkest pixels, the system sets all brighter ones to the value of 255 (white), while almost all face features used in given system have darker color than the face skin (Fig 3. Histogram).

The Haar classifier used in the system returns only approximate bounding boxes of detected features, however this data is insufficient for further processing. Having

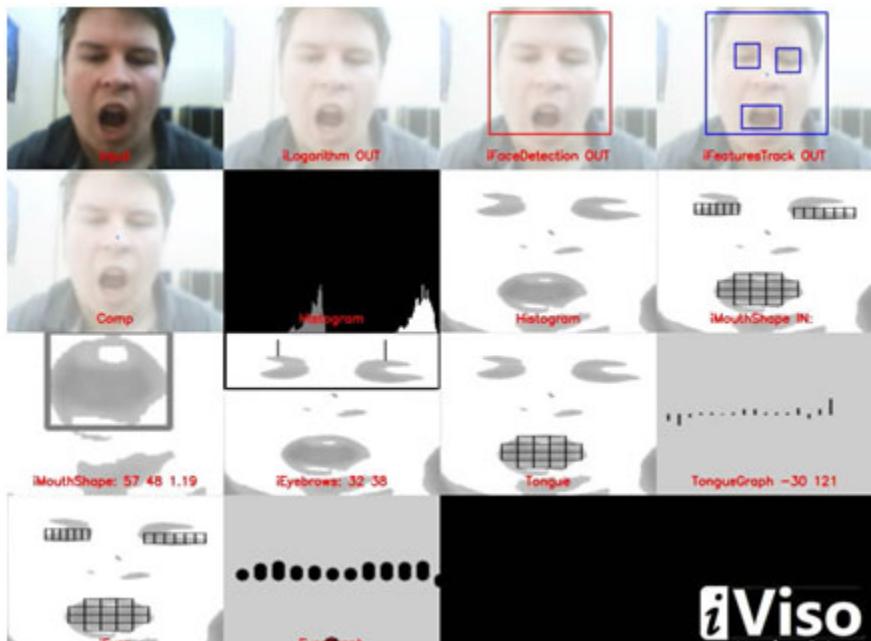


Fig. 3. Program output window with analysis results

rough region of mouth occurrence the program has to extract the exact dimensions of the mouth bounding box. The first stage employed morphological shape processing (erosion and dilation) to discard single pixels that can mislead the algorithm. Afterwards, the algorithm counts the number of dark pixels in columns located in the middle of approximated bounding box of mouth. In calculated array the first continuous series of dark pixels determines upper and lower line of the actual bounding box of the mouth. For horizontal boundaries the system uses the same algorithm (summing of pixels in rows, limited by calculated earlier boundaries). Obtained values are the exact bounding box dimensions, from which the aspect ratio can be calculated (Fig. 3. iMouthshape).

If mouth dimensions are bigger than the set threshold, tongue analysis is employed. In each mesh element the number of dark pixels is calculated (Fig. 3. Tongue). Tongue should appear in the image as white region, affecting number of dark pixels in element corresponding to tongue position. Detection of the position is realized by the calculation of difference of dark pixels quantities in two consecutive frames. The system assumes that tongue has been found if a specific pattern of maximum changes is obtained, to minimize influence of accidental minimum occurrence. Otherwise the state of tongue is set to hidden.

Next, the position of eyebrows with respect to reference line placed on the forehead is computed. The algorithm looks for the position of the first dark pixel below the reference point. The distance is the position of eyebrows (Fig. 3. iEyebrows).

Eyes analysis is quite similar to the analysis of mouth shape. First, the system tries to find the exact bounding box of each eye, as the bounding box of mouth, however in

this case last series of black pixels in sum array is taken into account. Next, the eye is divided into 6 element mesh and total number of dark pixels is calculated in each array element (Fig. 3. iEyes). Assuming that the iris with the pupil is placed in the middle of the eye, most of dark pixels are located in two middle elements of the array. Every movement of the eye results in changing quantity of dark pixels in the array. To determine the direction of movement of the iris, the program calculates pixel sum difference in two consecutive frames and then the difference in elements that are symmetrically placed around the central axis of the eye.

The last stage is displaying the outputs on the debugging image, stored by object responsible for monitoring and debugging the operation of the system (Fig. 3). The output from the proposed system can be used as the input for the neural network to recognize the following face gestures: mouth tightening, mouth opening, smiling, tongue on the right side, tongue on the left side, tongue up, tongue down, eyes squinting, eyebrows up, looking right, looking left, turning head left, turning head right, moving head up, moving head down.

4 Results

In order to evaluate the operation of the designed system several test were prepared and performed for five people (age 22 – 32) having different face characteristics (different skin tone, hair color, having beard) in different lighting conditions. For each person two sequences were recorded.

Lighting conditions play very important role in the vision-based system testing. In designed system deteriorating lighting conditions cause longer computation time of the frame (about 50% higher than bright frame).

All recorded sequences were tested to determine occurrence of recognized face gestures and analyze system response. Results are presented in Table 1, where accuracy, precision, sensitivity and specificity are calculated from formulas (1-4).

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (1)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (2)$$

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (3)$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \quad (4)$$

where TP – true positives, TN – true negatives, FP – false positives, FN – false negatives.

The overall accuracy of the system is equal to 98%, while precision 85%. The results are lowered, while the tongue analysis is not precise. This can be caused by untrained users operating the system. Precision can be also increased by the changes in the algorithm.

Table 1. Results of detection of analysed features

No.	Analyzed feature	Accuracy	Precision	Sensitivity	Specificity
1	Face detection	99,74%	100,00%	99,74%	-
2	Eyes detection	98,75%	100,00%	98,75%	-
3	Mouth detection	95,00%	100,00%	95,00%	-
4	Eyes turned left	99,68%	94,12%	100,00%	99,66%
5	Eyes turned right	98,05%	90,00%	81,82%	99,30%
6	Raised eyebrows	98,70%	100,00%	63,64%	100,00%
7	Smiling	98,05%	90,00%	81,82%	99,30%
8	Mouth opened	97,40%	86,96%	80,00%	98,94%
9	Head turned left	100,00%	100,00%	100,00%	100,00%
10	Head turned right	100,00%	100,00%	100,00%	100,00%
11	Moving head up	100,00%	100,00%	100,00%	100,00%
12	Moving head down	100,00%	100,00%	100,00%	100,00%
13	Tongue on the right side	97,40%	50,00%	50,00%	98,67%
14	Tongue on the left side	97,40%	50,00%	37,50%	99,00%
15	Tongue up	97,08%	28,57%	33,33%	98,34%
16	Tongue down	96,75%	75,00%	25,00%	99,66%

During the tests influence of individual face characteristics on the operation of the system was discovered. The main problem for the program was to detect eyebrows on faces of the people with blonde hair color. Lower sensitivity of detection of raised eyebrows gesture was caused by failed eyebrows detection, while they were not correctly segmented by the histogram analysis. In fact, in that case, eyebrows were hardly seen in the original image. The best results of gesture recognition were produced for people with dark hair color, while their face features were also darker than for blonde ones.

5 Conclusions

The designed system shows that vision-based systems recognizing face gestures provide many data that can be used for human-computer interfacing. However, the problem of the face features analysis and gesture recognition is nontrivial and requires high computation power of the computers on which such system is expected to be implemented. The main problems of the image processing are the proper lighting conditions. In the presented case the individual characteristics of skin and hair color of the users have also significant influence on the performance of the system. It is extremely difficult to implement the system that will fit to all face types, skin and hair colors, and, what is more, will operate in difficult lighting conditions. Designed system solves some of the problems presented above, like light conditions, however the influence of some of them remains unchanged.

The designed system can be very useful for people with physical disabilities, who do not have control over their hands and cannot operate computer in normal manner. iViso can be used also for rehabilitation purposes for people that suffer from facial paralysis. The system can be implemented in all cases where there is no possibility of

physical contact between human and computer input device or such a contact is prohibited due to contamination possibilities, for example in operating room where the surgeon operates on patient and requires some information on the screen, while his hands are occupied by tools.

References

1. Ekman, P., Friesen, W.: Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto (1978)
2. Tian, Y.L., Kanade, T., Cohn, J.F.: Recognizing action units for facial expression analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23(2), 97–115 (2001)
3. Liao, C., Chuang, H., Duan, C., Lai, S.: Learning spatial weighting via quadratic programming for facial expression analysis. In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 86 (2010)
4. Li, Y., Kang, S., Kim, Y., Jung, S.: Development of a facial expression recognition system for the laughter therapy. In: *IEEE Conf. on Cybernetics and Intelligent Systems (CIS)*, p. 168 (2010)
5. Faria, P.M., et al.: Interface Framework to Drive an Intelligent Wheelchair Using Facial Expressions. *Industrial Electronics*, 1791–1796 (2007)
6. Lyons, M.J., Haehnel, M., Tetsutani, N.: The Mouthesizer: A Facial Gesture Musical Interface, p. 230. *Siggraph*, Los Angeles (2001)
7. Królak, A., Strumiło, P.: Eye-Blink Controlled Human-Computer Interface for the Disabled, *Human-Computer Systems Interaction. Adv. in Soft Computing* 60, 123–133 (2009)
8. Arce, G.R.: Nonlinear Signal Processing: A Statistical Approach. Wiley, New Jersey (2005)
9. Gonzalez, C.R., Woods, E.R.: Digital Image Processing, 2nd edn. Prentice Hall, New Jersey (2000)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518 (2001)

Assessing Safety of Object Pushing Using the Principle of Reversibility

Yuri Gavshin and Maarja Kruusmaa

Tallinn University of Technology, Centre for Biorobotics,
Akadeemia tee 15a-111, 12618 Tallinn, Estonia
{yury,maarja}@biorobotics.ttu.ee
<http://www.biorobotics.ttu.ee>

Abstract. This article presents an implementation of an innovative safety module for a robot control architecture. It applies the principle of reversibility to assess intrinsic safety of actions and to adapt robot's behavior. The underlying idea is that all reversible actions are intrinsically safe. A practical experiment is conducted to demonstrate that a robot control architecture can develop complex safe behaviors. This is accomplished by using the safety module in conjunction with human-based knowledge and sufficiently high level of perception. A robot is placed in a room with a movable object while the safety module analyzes movements of the robot and the object. As the result, the robot can identify, for example, that pushing object into a corner is irreversible and thus unsafe.

Keywords: Reversibility, safety, abstract principles, roboethics.

1 Introduction

The latest statistical report in [1] shows that the number of autonomous service robots for home and professional use is growing. A multi-purpose robot-assistant is the vision for the future. This makes solving the issue of robot safety an important priority for robotics research. In its traditional formulation, the safety in robotics is not viewed in the context of autonomous decision-making, but rather as a responsibility of the designer [2]. Alternatively, robot safety is considered in a wider philosophical context – the overview of roboethics in [3] identifies most urgent, evident and sensitive ethical problems related to several sub-fields of robotics. Although it reports contributions from dominant moral theories, the overview does not report any work implementing those principles in practice.

A major effort in bringing theory to implementation is exploring the idea of embedding ethical behavior into a military robot, in the form of ethical rules [4]. In contrast to such autonomous, but pre-programmed robot control, we use abstract principles as a basis for safe behavior. Our underlying idea is that all actions that are reversible, are also intrinsically safe. Most of the harmful actions the robot can do (for example, falling down the stairs, breaking an object, etc.) are also irreversible. There can be irreversible actions that the robot is allowed

or expected to do (e.g. a vacuum cleaning robot cleans a floor irreversibly) but then the robot is designed for doing those actions, and it is an informed decision of the user to use this robot for these purposes.

In our previous work ([5], [6]) we used the principle of reversibility to develop safe behaviors by suppressing irreversible actions. The observed behaviors were obstacle avoidance and locality. However, further applicability of our approach was limited by trivial state identification logic – plain sensor data from the sensors was used. In this paper we are combining this abstract principle with human-based knowledge. It includes, but not limited to, environment modelling, model-aware state identification, localization and planning algorithms.

We argue that such a combination enables development of smarter behaviors, similar to other hybrid intelligent algorithms and applications [7]. One of the strongest points of the reversibility principles governing robot behavior is the ability to behave safely without any prior knowledge. Scalability and applicability of such pure approach can be extended with human-based knowledge. Abstract principles, on the other hand, can govern robot behavior in unexpected situations, where no pre-defined rules can be applied.

In this paper we report results of the experiment where the simulated and the real robots learn to reversibly manipulate an object by pushing it back and forth. The principle of action reversibility is used to identify intrinsically-safe decisions. The robot is allowed to make such decisions autonomously, while explicit authorization is required for irreversible ones. We provide a formal framework to describe reversible actions and assess safety of system decisions, allowing a test robot to process its experience with the environment.

Next section describes the control system architecture and its safety module together with a reversibility-based sub-module to assess decision safety and alter system behavior. Section 3 contains the structure and the theoretical framework. Further, experimental setup with implementation details are presented. In sections 5 we report results of the experiment and the last section contains conclusions with plans for the future.

2 Safety Module for Cognitive Robot Control Architecture

As a testing ground for our research experiments to study the safety module, we created the *PAHPAM* system (Programmable Architecture for Hierarchical Perception, Actuation and Modeling). Since this system is not a primary objective of this article, we will describe only the most relevant of its aspects.

The *PAHPAM* has three standard levels of a general reactive-deliberative architecture – reactive, hybrid and deliberative. Hardware-specific functionality is implemented as a subcomponent of the reactive level, which is located on the robot, while hybrid and deliberative ones can also run on a separate more powerful machine. There are quite many assumptions and prerequisites for the whole system and the safety module to work. We have solved those prerequisites in a minimalistic fashion by hard-coding the routines we cannot generalize and

formalize, since most of the problems are difficult research problems by themselves and are yet to be solved. We use Monte-Carlo localization algorithm to solve localization problem. Object identification is simplified by the round shape of the object and the IAV method [8] is used to identify such circular objects in laser rangefinder scans. State identification problem is solved by using only the relevant information with a balanced level of detail. Important part of our robot control system is a model of the environment for analysis, planning and learning. The model doesn't have to be ideal – in case of inconsistencies between the modeled and the actual movement, the model can be updated and a new plan can be created.

The safety module ensures that system's decisions are safe and within predefined bounds – similar to the *ethical governor* in [9]. Before making decision, the system checks with the safety module whether the decision is allowed from the safety perspective (see Fig. 1 for activity diagram). A set of rules/patterns to explicitly (dis)allow specific decisions in specific states is a first stage of safety assessment inside the module. If the specified state-action pair matches those patterns, the answer is generated based on this pre-programmed or learned knowledge. Context- and task-specific overrides (for example, to enable irreversible, but useful actions) should be added to this set of rules/patterns.

When no rule/pattern can be applied and the robot is going to make an autonomous unauthorized decision, a reversibility-based analysis module is used to assess intrinsic safety of making the action from the state in question. Such safety architecture allows to make a hybrid system in terms of cooperation between system's designer knowledge in form of rules with patterns to apply them and a reversibility-based logic used as a backup, when no rules can be applied.

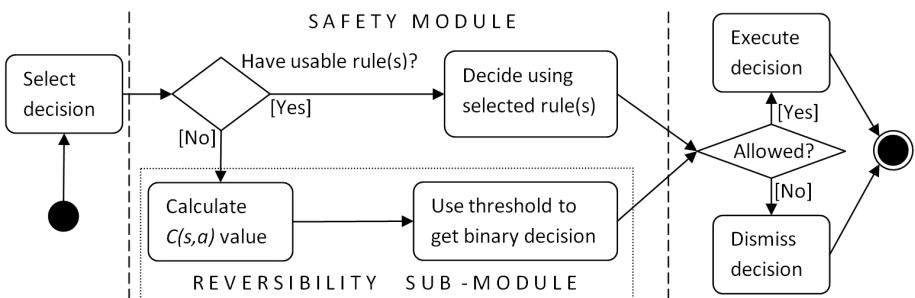


Fig. 1. The activity diagram for proposed safety module

3 Reversibility Models for Reversibility Based Sub-Module

Reversibility MDP-Model (RMM) is a finite Markov Decision Process with a reversibility function $C(s, a) \leq 0$. MDP is a 4-tuple $(S, A, P(\cdot, \cdot), R(\cdot, \cdot))$, where S is a finite set of states, A_s is a finite set of actions available from state

s , $P_a(s, s')$ is the probability that action a in state s will lead directly to state s' , $R_a(s, s')$ is the (expected) immediate cost of making action a from state s , followed by a transition to state s' with probability $P_a(s, s')$.

$C(s, a)$ is the total expected “cost” of $s \rightarrow s' \rightarrow s$ transition, i.e. reversing an action a made from state s . $C(s, a) = -\infty$ for absolutely irreversible actions and $C(s, a) = 0$ for perfectly reversible ones. To calculate it, we search for a path $p = (a_0, a_1, \dots, a_n)$, where $a_0 = a$ to make the $s \rightarrow s' \rightarrow s$ transition. This is done by iterating through the set P_s of possible paths; for every candidate path $p = (a_0, a_1, \dots, a_n)$, the candidate $C_p(s, a)$ value is calculated as follows:

$$C_p(s, a) = \min\left(\sum_0^n R_{a_i}(s_i, s'_i) \cdot P_{a_i}(s_i, s'_i) : s'_n = s\right). \quad (1)$$

If none of the possible $s'_n = s$, then $C_p(s, a) = -\infty$. If there are no suitable candidate paths found, $C(s, a) = -\infty$, otherwise the maximum (or “sufficiently high”) $C_p(s, a)$ value of one of the candidate paths is returned as a result. A value is “sufficiently high”, when $C_p(s, a) \geq C_{min}$. For the binary decision, action a from state s is reversible, if $C(s, a) \geq C_{rev}$. Both C_{min} and C_{rev} threshold values are set by a cognitive system or a designer, based on the knowledge about the context.

In other words, with $C_p(s, a)$ we predict a “cost” of making a sequence of actions $p = (a_0, a_1, \dots, a_n)$, taking in consideration only the possible outcomes, where final state is s . Path with the smallest $C_p(s, a)$ is the easiest way to reverse action a from state s and its cost is returned as $C(s, a)$ assessment value. A trivial candidate for a path is a single action a itself: if $C_{p=(a)}(s, a) \geq C_{min}$, then $C_{p=(a)}(s, a)$ is returned as $C(s, a)$ value.

The main purpose of the reversibility-based module is to assess the safety of intrinsic decisions through the study of their reversibility. This sub-module, as the safety module in general, uses knowledge from the current context chosen by a cognitive system. Every context has two reversibility models: $RMMi$ for internally simulated and $RMMa$ for the actual experience. The purpose of such internal simulation is to plan robot’s actions and identify (ir)reversible actions before actually making them for the first time.

$RMMi$ and $RMMa$ share the same sets S and A , generated by state and action identification modules of a cognitive system. Probability function $P_a(s, s')$ is derived from observed actual transitions for $RMMa$ or simulated ones for $RMMi$. Similarly, reward function $R_a(s, s') \leq 0$ is derived from the simulated or actually observed cost (energy, time, damage, etc) of the action.

The on-line nature of assesment considerably limits the number of candidate paths to try. To overcome this, we use C_{min} threshold to stop search process when at least one $C_p(s, a) \geq C_{min}$ value is found. Additionally, our implementation of reversibility and $C(s, a)$ calculation is based on cyclical state-action transitions. Such cycles can be given in advance by a human, deduced theoretically from an internal model of the environment or identified from statistics of actual state-action transitions.

A simple example of such a cycle is a composite action “go 1 metre forward”, then ‘go 1 metre backward’, or vice-versa. A set of cycles can be used as candidate paths for $C(s, a)$ calculation – the first action of starting sequence of actions in the cycle is expected to be undone by the rest of the cycle.

4 Experimental Setup

The purpose of the experiment is to demonstrate how the aforementioned action cycles can be identified and then used to assess action safety on-line through $C(s, a)$ calculation. The main purpose of the experiment is to better explain our approach and to show how the reversibility principle works in more complex scenarios than we have previously used. Our experiments in [5] and [6] showed how a robot can identify irreversible actions in the context of self-movement and demonstrated safe behaviors by avoiding such actions. In this experiment we want a robot to be able to undo the change to the environment – a round movable object. A practical example of such a behavior could be the vacuum cleaner, which cleans not only the free space, but also under movable objects placing them back after cleaning the area initially occupied by the object. As a result, after cleaning the floor, the room layout would stay the same, unless instructed otherwise.

In our experiment we use MetraLabs’ Scitos G5 robot; test-runs are made on the actual robot and in a simulated environment, which is a copy of the actual “room” (see Fig. 2). The robot control framework is connected to *Player* server [10], which in turn controls either the actual robot or its model in *Stage* simulator [10]. In our setup the “green” object is the only round item in the environment and its approximate radius is known. The round object’s size and its position in robot’s coordinates are identified from laser rangefinder scans, filtering out the occasional “wrong” objects by radius threshold; laser sensor position and settings are known. In this experiment we use the following set of

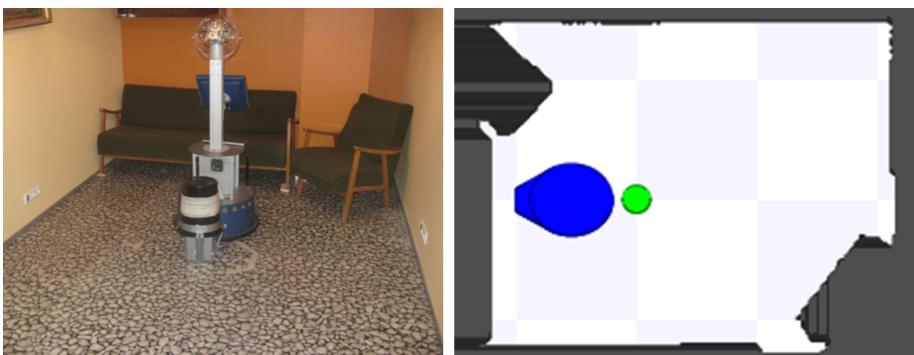


Fig. 2. The physical experimental setup of the room (*left*) and the simulation in Player/Stage (*right*); the grid-map is generated from the real room

actions: “F” – move 0.6 metres forward, “B” – move 0.6 metres backward, “f” – move 0.15 metres forward, “b” – move 0.15 metres backward, “L” – rotate 60 degrees left and “R” – rotate 60 degrees right. States of the robot and the object are identified by X and Y coordinates, rounded to the nearest multiple of 0.13 metres and orientation, rounded to the nearest multiple of 0.13 radians; since the “green” object is round, its orientation is always 0.

We start our experiment with a free movement of the robot to collect statistics (average and standard deviation of covered distance and rotated angle for both local and global odometry) for the six actions used. If the robot pushes the “green” object we collect statistics about the initial and final distance between centers of the robot and the obstacle for the actions that change the state (position) of the object. For simplicity, we analyze only the “F” action effect on the object in this experiment. Additionally, we narrow analysis to specific relative position of the object in robot internal frame coordinates – when object is placed directly in front of the robot, 0.5 meters from its center of rotation.

After the data is collected we can use it to simulate different paths internally, taking only robot and object interaction into account. Next, we search for the cycles – paths that put the “green” object back to its initial state. This part can be done offline – search through the space of possible paths is a time-consuming process. Currently, on our test machine with Radeon HD5770 video, GPU-based exhaustive analysis of 11-step paths (which gives the first valid cycles) takes 10 seconds. We are searching for cycles by simple iteration over all possible paths, starting from the shortest ones. We limit the search space by setting the maximum allowed runtime (10 minutes) and length of the path (20 actions). The search is executed in a separate thread, which allows to use new path as soon as it is identified and the search can be stopped, if needed.

In the final part of our experiment the safety module is used to govern behavior of a robot on-line. The previously identified cycles that start with “F” are simulated internally, now taking immovable obstacles into account. $P_a(s, s')$ is calculated by executing internal simulation with obtained physical movement parameters. Possible next states with possibilities are received by analyzing 10 such actions with normally distributed random error, using standard deviation obtained in the beginning of the experiment. $R_a(s, s')$ is 0 minus the length of the movement in meters or rotation angle in radians; if collision is detected by simulation code, then the $R_a(s, s')$ is additionally multiplied by 100. Threshold values were set as follows: $C_{min} = 2$, $C_{rev} = 20$.

5 Results

The data acquisition part of the experiment revealed virtually no difference between simulated and real parameteres of physical movements in this task of a robot moving itself and a round obstacle. Therefore, most robot test-runs were performed in Player/Stage simulation, where data acquisition is fast and automated. The experiments showed that many paths can be found to successfully undo the movement of the “green” object by “F” action of the robot. The paths

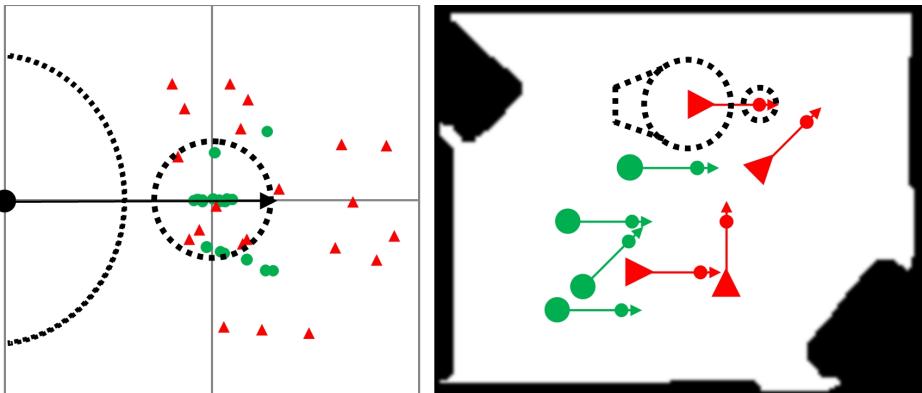


Fig. 3. The result of wall-free trials to reverse “F” action in a robot-centered reference frame are *on the left*, grid size is 0.5 metres. The result of the trials in the room with walls in the global reference frame are *on the right*.

go around the object and push it from the opposite side, some make “b” action before going around and some don’t.

Fig. 3 on the left shows the situations learned by the end of the trial. The picture is drawn in robot-centered coordinates with the black arrow representing “F” action. The green dots are the positions of the object, from where the pushing is considered reversible. Red triangles are positions of objects from where the pushing of the objects is considered irreversible. It is easy to see that the situations where the object is in front of the robot have been successfully reversed. However, more complicated situations, where the robot touches the object with its side and the objects slides away, are harder to undo. Whereas the two additional line-like clusters of green dots are unexpected – the robot pushes the object away from X axis of initial robot’s pose and then pushing it back while finishing the path around the object. It shows that the robot is even able to learn to reverse actions influenced by rather complicated physics of sliding and friction.

The result of safety module governing the “F” action in the global reference frame is shown in Fig. 3 on the right – robot’s pose with object position and action length is overlaid upon the map of the room. Reversible object pushing poses, identified by the green enlarged dots with arrows are the ones made towards the object near the center of the room while the robot is closer to room’s walls. The red combinations of enlarged triangles with arrows represent irreversible situations and the robot has correctly identified that there is no room to maneuver around the object to push it back.

6 Conclusions

Based on the results of the experiment we conclude that non-trivial and quite complex cycles of actions can be successfully identified, allowing the robot to

manipulate objects in a reversible manner – pushing them from one side and undoing such action by driving around the objects and pushing them back from the opposite side. Suppression of the irreversible actions while taking immovable objects into account results in further increase of behavioral complexity – robot “understands” that pushing object into a corner is irreversible and thus unsafe.

Future Work. Our short-term plans are to optimize the path search and simulation algorithms. In the long-term we are going to use the same approach of using abstract principles to develop safe behaviors, but moving further towards more complex and real-life problems/scenarios.

Acknowledgments. This research was supported in part by scholarship from Estonian Information Technology Foundation.

References

1. IFR Statistical Department, http://www.worldrobotics.org/downloads/PR_2010-09-14_service_EN.pdf
2. Veruggio, G., Operto, F.: Roboethics: Social and ethical implications of robotics. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, pp. 1499–1524. Springer, Heidelberg (2008)
3. Sawyer, R.J.: Robot ethics. *Science* 318(5853), 1037 (2007)
4. Arkin, R.C.: Governing lethal behavior: Embedding ethics in a hybrid deliberative/reactive robot architecture. *Tech. Rep. GIT-GVU-07-11*, Georgia Tech. (2007)
5. Kruusmaa, M., Gavshin, Y., Eppendahl, A.: Don't do things you can't undo: Reversibility models for generating safe behaviours. In: ICRA 2007, pp. 1134–1139 (2007)
6. Gavshin, Y., Kruusmaa, M.: Comparative experiments on the emergence of safe behaviours. In: TAROS 2008, pp. 65–71 (2008)
7. Corchado, E., Abraham, A., de Carvalho, A.C.P.L.F.: Hybrid intelligent algorithms and applications. *J. Inf. Sci.* 180(14), 2633–2634 (2010)
8. Xavier, J., Pacheco, M., Castro, D., Ruano, A., Nunes, U.: Fast line, arc/circle and leg detection from laser scan data in a player driver. In: ICRA 2005, pp. 3930–3935 (2005)
9. Arkin, R., Ulam, P.: An ethical adaptor: Behavioral modification derived from moral emotions. In: CIRA 2009, pp. 381–387 (2009)
10. Gerkey, B., Vaughan, R., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: ICAR 2003, pp. 317–323 (2003)

Class Prediction in Microarray Studies Based on Activation of Pathways

Henryk Maciejewski

Institute of Computer Engineering, Control and Robotics,
Wrocław University of Technology,
ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Henryk.Maciejewski@pwr.wroc.pl

Abstract. This paper presents a novel approach to building sample classifiers based on microarray gene expression studies. This approach differs from standard methods in the way features are selected. Standard methods focus on features (genes) with most differential expression between classes of samples compared, while the proposed approach takes into account apriori domain knowledge of relationships between features, available e.g., in the form of pathway or gene-ontology databases. Features for classification are then selected on the basis of activation of pathways (gene sets) rather than mutually unrelated genes with very high individual predictive power. Performance of the proposed method is illustrated on the basis of sample microarray studies.

Keywords: Classification, feature selection, gene expression microarray, pathway analysis.

1 Introduction

Identification of gene expression patterns to be used as predictors for sample classification seems to be one of the most promising albeit challenging applications of gene expression microarrays or similar massive throughput techniques. Many attempts to build diagnostic or prognostic classifiers have been reported (e.g., [8][11][12]), and vast literature has recently appeared devoted to methodological issues related to feature selection and prediction based on high dimensional data (see e.g., [6] for an analysis of different approaches). Generally, current approaches to class prediction focus on sets of features which show the strongest relationship with the target and thus promise the highest predictive power. Hence the common approach is to use top differentially expressed (DE) genes as features, without taking into account potential relationships among features. Although intensively studied, the problem of class prediction based on massive throughput data is still not satisfactorily solved. The main issues are related to instability of models and to problems with proper verification of predictive performance. These problems were analyzed in our previous works ([13][14]), where different methods of ranking genes were compared in terms of predictive performance of classifiers built using top-ranking genes. We also showed that such

classifiers are generally very sensitive to the selection of model's parameters (such as the number of features, etc.), ie., small changes in parameters produce unstable estimates of prediction error.

Building on our previous work, this paper proposes a different approach to class prediction based on an enhanced procedure of feature selection. Feature selection will include domain knowledge about potential relationships among features (genes). Such knowledge of groups of functionally related genes is available in databases e.g., KEGG, Gene Ontology or Biocarta, and is now being actively developed. The proposed method is motivated by recent developments where the real difference between the groups compared (e.g., disease vs. control) is attributed to coordinated but possibly small difference in expression of functionally related genes rather than very strong difference in a few most DE genes (see e.g., [17] where the authors argue that “An increase of 20% in all genes encoding members of a metabolic pathway may dramatically alter the flux through the pathway and may be more important than a 20-fold increase in a single gene”). The proposed method aims to base class prediction on activation of pathways which may bring stabilization of features and presumably improved performance. It can be envisaged that the proposed classifier can be later fused with standard classifiers or alternatively pathway based features can be combined with data-driven features to form a robust hybrid system for class prediction on the basis of high-dimensional data ([4][18]).

First, we summarize different approaches to analysis of activation of pathways (or other functionally related or apriori given groups of features). Next, the proposed method of pathway based classification is presented. Finally, a numerical example is given based on two microarray studies to provide a preliminary analysis of how the method compares to standard approaches.

2 Analysis of Pathway Activation

The problem of discovery of groups of genes (such as signalling pathways) which are differentially expressed between classes of samples has been actively investigated in bioinformatics literature. Several methods have been proposed, e.g., GSEA (Gene Set Enrichment Analysis, [16][17]), GSA (Gene Set Analysis, [7]), globaltest ([9]), or methods proposed in [15]. These methods differently define the null hypothesis of no association between an apriori defined subset of genes and the target variable. Some of the methods analyze only expression of genes in a given subset, without taking the remaining genes into account (*self-contained* methods), while other methods compare expression of genes in the set against expression of the rest of the genes (*competitive* methods). Methodological validity of these different approaches was analyzed in [10]. It should be mentioned that gene set analysis was performed primarily to extract biological insight from massive throughput results and to improve interpretability of underlying differences between classes of samples.

In this work we use gene set analysis as a tool to generate features for classification of samples. We identify the most differentially expressed (or activated)

pathways and then use genes in these pathways as features. For the purposes of this work, gene set analysis will be carried out with the globaltest method. This choice is motivated by an interesting property of the globaltest which attempts to express activation of pathways (gene sets) in terms of a regression model with class label as the target and gene expressions as independent terms. More specifically, logistic regression is assumed, such that

$$E(Y_i|\beta) = h^{-1} \left(\alpha + \sum_{i=1}^m x_{ij} \beta_j \right). \quad (1)$$

where h is the logit function, $x_{i\cdot}$ denotes vector of expression of m genes in the gene set for sample i , with class label Y_i , and β_j is the coefficient for gene j .

The p-value of the test of no association between the target and the members of the set is then used as the measure of gene set expression, and thus it can be a useful measure for pathway-based feature selection.

3 Classification Based on Activation of Pathways

We assume that the results of a massive throughput study are given as a matrix $X_{n,p}$ which represents p features (gene expressions) measured for n samples, with class designation for a sample i given in Y_i , $i = 1, 2, \dots, n$. We also assume that a set PWDB of d subsets (denoted PW_i , $i = 1, 2, \dots, d$) of features is specified apriori, these may represent domain knowledge of groups of related features (e.g., genes in a pathway, genes with common GO term, or with related chromosome location). The proposed method allows to (a) build the sample classifier given X , Y and PWDB, and (b) estimate the expected prediction error for new samples (denoted EPE). Due to relatively small n and $p \gg n$, the EPE will be estimated by data reuse techniques, such as cross validation (CV) where the data are repeatedly split into training and test partitions, with the EPE calculated as the average misclassification rate over all the test partitions. More specifically, we will use *internal* cross validation where subsequent iterations involve a feature selection step. Although frequently overlooked in literature (where feature selection is realized once and preceeds the CV procedure), this policy is important to obtain a reliable measure of classifier performance, as shown by Markowetz in [15]).

The proposed algorithm can be summarized in the following steps.

1. Leave out sample i , $i = 1, 2, \dots, n$ for model testing, ie., remove row i from X and element i from vector Y and denote the remaining matrix and vector as X^i and Y^i .
2. Using the training data (X^i, Y^i) calculate the p-value with the global test for each of the PWs in PWDB. Order PWs by the increasing p-value: $PW_{(1)}$, $PW_{(2)}$, etc.
3. Remove columns from X^i related to features not present in $PW_{(1)}$, denote this matrix as X_{tr}^i .

4. Using the training data (X_{tr}^i, Y^i) fit a predictive model f and classify the sample Y_i as $\hat{Y}_i = f(Y_i)$.
5. Repeat steps 1 through 4 for $i = 1, 2, \dots, n$.
6. Calculate the expected misclassification rate as $EPE = \sum_{i=1}^n I(\hat{Y}_i \neq Y_i)$.

In the following section, we will compare performance of this approach with the standard method of class prediction where top most differentially expressed genes are selected as features. It should be noted that the proposed approach differs from current approaches primarily in the way features are generated: we propose to use information on pathway activation for this purpose. The remaining elements of our procedure, especially the use of internal cross validation are taken from current, state-of-the-art approaches to class prediction. It should be also noted, that in step 4 of the proposed procedure we are not specific on what type of predictive model is fitted to the data. For the purposes of the numerical example, two classifiers were selected: Support Vector Machine (SVM) and logistic regression. In the next section, we compare the proposed approach with the standard method not only in terms of the EPE , but also in terms of stability of the feature sets selected in subsequent steps of internal CV. The purpose of this is to empirically verify a supposition that analysis of gene sets may yield more reproducible features when data slightly changes (such as in subsequent steps of internal cross-validation). It is often observed that small changes in data produce quite different feature sets when standard approach is used, which may account for unstable behaviour of classifiers built from microarray data ([14]).

Table 1. Coincidence matrix for the PW and STD methods for logistic regression and SVM models. Rows are labeled by the true class and columns by the predicted class.

PW Reg		PW SVM		STD Reg		STD SVM	
True	0	True	0	True	0	True	0
0	31	6	0	32	5	0	31
1	5	37	1	7	35	1	8

4 Numerical Example – Performance of Classifiers and Stability of Results

The first example is based on a subset of the ALL data, published in [3], and representing B-cell leukemia patients. We build the classifier to distinguish between leukemia and control samples (dataset includes 79 samples, 37 leukemia and 42 control, designated in the original data as ‘BCR/ABL’ and ‘NEG’, respectively). Prior to analysis, the dataset was preprocessed (which involved logging and non-specific filtering to remove genes with very low or roughly constant expression across samples, technically genes with IQR below 0.5 or signal below 100 in more than 75% of samples were removed). This reduced the number of features from original 12625 to 2391. In the study we include results for median-centered data

Table 2. Comparison of features selected by the PW method (left) and STD method (right). Results of the test of differential expression are shown as the Welch test statistic and raw and multiple testing adjusted p-values.

gene	teststat	rawp	adjp	gene	teststat	rawp	adjp
1636_g_at	-9.1304	1e-04	1e-04	1636_g_at	-9.1304	1e-04	1e-04
39730_at	-8.6041	1e-04	1e-04	39730_at	-8.6041	1e-04	1e-04
1635_at	-7.1679	1e-04	1e-04	1635_at	-7.1679	1e-04	1e-04
37027_at	-5.7094	1e-04	9e-04	40202_at	-6.3309	1e-04	1e-04
2039_s_at	-4.7615	1e-04	0.0147	37027_at	-5.7094	1e-04	9e-04
37001_at	-3.2675	0.0022	0.6541	40480_s_at	-5.4687	1e-04	0.0012
577_at	3.0297	0.0039	0.8252	39837_s_at	-5.4503	1e-04	0.0012
1007_s_at	-2.8701	0.0049	0.9119	36591_at	-5.4169	1e-04	0.0012

and for original non-centered data. We tested the SVM and logistic regression classifiers. As PWDB the KEGG pathway database was used with 210 pathways.

We first observe that in 79 rounds of the CV procedure, the pathway ‘04360’ was always selected as the winner, with $pValue=7.05e-10$ and with 26 genes. We thus first compare the classifier based on this pathway with the classifier based on 26 most DE genes. Results are shown in Table 2. We observe slightly better performance of the pathway (PW) method than the standard method (STD), which may be surprising if association of the features with the target is analyzed - Table 2. Apparently, very high association of the STD features (most of them are significant under multiple testing correction! - see right panel of Table 2, with $adjp < 0.05$) does not guarantee the highest predictive performance. All these results were obtained for median-centered data. For non median-centered data, the overall misclassification rate for the PW method remains similar (12 misclassified items), while performance of the STD method deteriorates remarkably (18-19 misclassifications).

The overall performance of the pathway-based (PW) and standard (STD) methods is summarized in Fig 2, with the analysis done for changing feature vector sizes. The PW model was tested for up to 26 features, as this was the number of genes in the winning pathway. We observe that with growing number of features, performance of both models deteriorates (the effect can be attributed to overfitting). However, the set of PW-based features seems to outperform the STD features, although individual STD features show stronger association with the target (see Table 2). This preliminary result confirms the supposition formulated earlier, that the gene sets which promise best predictive power do not necessarily consist of the most DE genes.

In Fig. 2 stability of features selected in iterations of CV procedure is summarized. The figure presents how frequently (the value on the X axis) a set of features of size given on the Y axis was selected in 79 CV iterations. We observe that the PW method yields very stable features, despite the fact that features

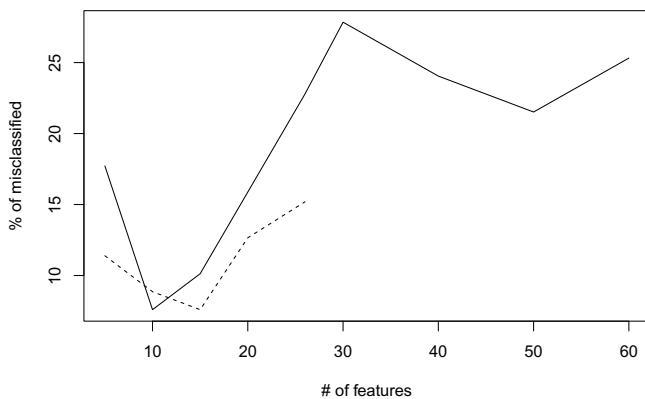


Fig. 1. Misclassification rate for the ALL data as a function of number of features. Solid line - STD method, dashed line - PW method.

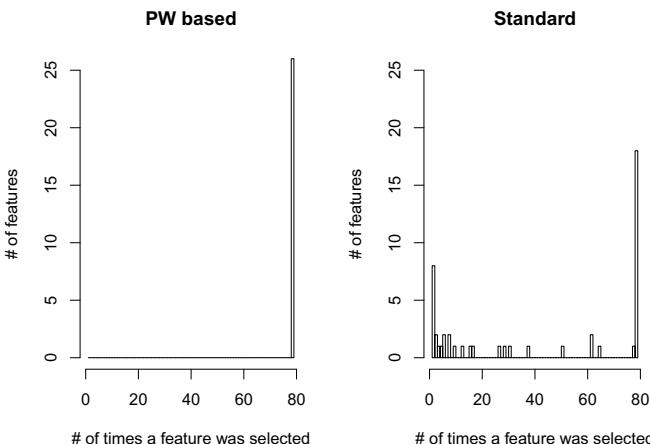


Fig. 2. Stability of features selected in subsequent CV iterations. See text for interpretation.

were selected on changing data. In STD procedure, 18 features were repeatedly selected, the rest were constantly changing. This may be the reason for overall worse performance of the STD classifier.

The second example is based on the chronic lymphocytic leukemia data (available in the CLL Bioconductor package). The data contains samples labeled as either progressive (14 samples) or stable (8 samples) in regards to disease progression. Data preprocessing was done as in the ALL example, which reduced the number of genes to 1687. Performance of classifiers is summarized in Fig. 3. First, we observe that this dataset is more difficult to classify, which could be expected due to (a) small number of samples and (b) relatively large p-value (≈ 0.01)

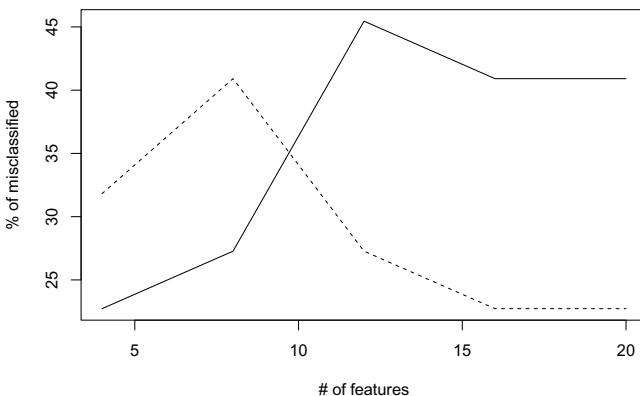


Fig. 3. Misclassification rate for the CLL data as a function of number of features. Solid line - STD method, dashed line - PW method.

calculated by the globaltest to verify association of all genes with the target. The same p-value for the ALL data equals ≈ 0.0003 , and for Golub's leukemia data $\approx 10^{-14}$, [9]. (Due to this strong association, Golub's data is considered easy to classify). Secondly, with growing number of features, the PW-based classifier again seems to outperform the standard classifier, consistently with the previous example. It should be noted that in this example we used genes from two winning PWs as features (ie., $PW_{(1)} \cup PW_{(2)}$), as the winning $PW_{(1)}$ (ID=00670) contains only 4 genes.

5 Conclusions

In this work an idea was proposed on how apriori information on sets of functionally related genes can be incorporated in class prediction based on high dimensionality data (such as results of gene expression microarrays). The analysis of sample microarray studies suggests that the pathway based features may promise improvement in performance and stability of classifiers. However, these results should be regarded as preliminary, as it seems highly unlikely that differences between classes can be attributed to features from just one (albeit winning) pathway. Hence further research is necessary to (a) include relevant genes from important pathways or other apriori specified sets, and (b) comprehensively evaluate this approach.

This work was sponsored by the grant MNiSzW N516 510239.

References

1. Adewale, A.J., et al.: Pathway analysis of microarray data via regression. *J. Comput. Biol.* 15(3), 269–277 (2008)
2. Allison, D.B., et al.: Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics* 7, 55–65 (2006)

3. Chiaretti, S., Li, X., Gentleman, R., et al.: Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood* 103, 2771–2778 (2004)
4. Derrac, J., García, S., Herrera, F.: A First Study on the Use of Coevolutionary Algorithms for Instance and Feature Selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
5. Dinu, I., et al.: Gene-set analysis and reduction. *Briefings in Bioinformatics* 10(1), 24–34 (2008)
6. Dudoit, S., Fridlyand, J., Speed, P.: Comparison of discriminant methods for classification of tumors using gene expression data. *Journal of American Statistical Association* 92, 77–87 (2005)
7. Efron, B., Tibshirani, R.: On testing the significance of sets of genes. *Ann. Appl. Stat.* 1(1), 107–129 (2007)
8. Glas, A.M., et al.: Converting a breast cancer microarray signature into a high-throughput diagnostic test. *BMC Genomics* 7, 278 (2006)
9. Goemann, J.J., et al.: A global test for groups of genes: testing association with a clinical outcome. *Bioinformatics* 20(1), 93–99 (2004)
10. Goeman, J.J., Buehlmann, P.: Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics* 23(8), 980–987 (2007)
11. Khan, J. et al.: Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Med.* 7, 673–679 (2001)
12. Lin, Y.H., et al.: Multiple gene expression classifiers from different array platforms predict poor prognosis of colorectal cancer. *Clin. Cancer Res.* 13, 498–507 (2007)
13. Maciejewski, H.: Quality of feature selection based on microarray gene expression data. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part III. LNCS, vol. 5103, pp. 140–147. Springer, Heidelberg (2008)
14. Maciejewski, H., Twaróg, P.: Model instability in microarray gene expression class prediction studies. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2009. LNCS, vol. 5717, pp. 745–752. Springer, Heidelberg (2009)
15. Markowetz, F., Spang, R.: Molecular diagnosis. Classification, Model Selection and Performance Evaluation, Methods Inf. Med. 44, 438–443 (2005)
16. Mootha, V.K. et al.: PGC-1 alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics* 34(3), 267–273 (2003)
17. Subramanian, A., et al.: Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* 102(43), 15545–15550 (2005)
18. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants – Evolutionary and Neural Methods of Training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 590–597. Springer, Heidelberg (2010)

A Hybrid Approach for ECG Classification Based on Particle Swarm Optimization and Support Vector Machine

Dawid Kopiec and Jerzy Martyna

Institute of Computer Science, Jagiellonian University, ul. Prof. S. Łojasiewicza 6,
30-348 Cracow, Poland

Abstract. In this paper, we describe a hybrid framework based on Particle Swarm Optimization (PSO) and Support Vector Machine (SVM) for the ECG signal classification. By means of a specially prepared pre-processing method we extracted the most significant features from the 12-lead ECG recording from the standard ECG database. In order to reduce the dimension of the input data a particle swarm optimization (PSO) was used. The numerical results indicated that the presented classifier achieved 94.16% recognition rate.

1 Introduction

The electrocardiogram (ECG) is an important tool providing very useful information about the state of the heart. The ECG signal is described as a recurrent wave sequence of P-, QRS- and T-waves associated with each beat. In order to analyse the qualified cardiac period all the locations of the ECG signals components should be specified, such as the P and T waves, and QRS complexes.

Various ECG signal classification techniques have been recently discussed. Lu [10] proposed neuro-fuzzy system to diagnose *acute myocardiac infarction*. Gerardo et al. [2] utilized hybrid neuronal-fuzzy networks to quantify and characterize the *heart rate variability*. Statistical methods, such as Markov model was also suggested by Messadeg [11] to classify some cardiac. The multilead approach for ECG delineation based on the wavelet transform was proposed by Llamedo Soria et al. [9].

Recently, a novel machine-learning method, called the support vector machine (SVM), has been developed in the field of the pattern recognition, regression estimation, etc. The SVM was introduced by V. Vapnik [15]. Compared with the traditional neural networks, the SVM can obtain a global optimal solution and avoid the curse of dimensionality. In particular, the SVM has the ability to divide samples into two or more samples with the widest margin between them and the extension of this concept to a higher dimensional setting by means of using the kernel function in order to represent similarities in this setting.

Therefore, the SVM method is used more and more frequently for detecting ECG signals. Among others, S. Osowski et al. (2004) [13] developed an expert system based on the SVM for reliable heartbeat recognition purposes. M.H. Song

et al. [14] suggested an algorithm for the *arrhythmia* classification, which involved the reduction of feature dimensions by means of the linear discrimination analysis and an SVM based classifier.

The main goal of this paper is to use the SVM as a tool for the ECG classification, and in particular, as a tool for detecting QRS-complexes, P- and T-waves in the 12-lead ECG signal with the use of specially prepared preprocessing methods. These methods are suitable for a low signal-to-noise ratio of the ECG waveforms. Moreover, the applied classifier achieved 99% detection rate.

The paper is organized as follows: in section 2 support vector machines are presented. Section 3 provides a definition of a singular value decomposition (SVD) used in the preprocessing of the ECG waveforms and properties SVD. Section 4 explains the particle swarm optimization (PSO) used for reducing the dimensional size of the input data records. In section 5, we present a computational example of the ECG signal analysis conducted by means of our method. Section 6 concludes the paper.

2 A Brief Review of the SVM Method

The SVM method [15] has been used for classification, micro-array gene expression data, etc. We recall that the SVM maps the input patterns into a higher dimensional feature space by means of using as a nonlinear mapping chosen a priori. Thus, in this higher dimensional feature space is constructed a linear decision surface. Let us consider m -dimensional training data x_i ($i = 1, 2, \dots, n$) and their class labels y_i , where $y_i = 1$ and $y_i = -1$ for classes 1 and 2, respectively. For the linearly separable input data, the following relation gives the equation for the hyperplane separating of two different classes

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = \sum_{j=1}^k w_j \phi_j(\mathbf{x}) + w_0 = 0. \quad (1)$$

where $\phi(\mathbf{x}) = [\phi_0(x), \phi_1(x), \dots, \phi_k(x)]^T$ with $\phi_0(x) = 1$. $\mathbf{w} = [w_0, w_1, \dots, w_k]^T$ is the weight vector of the network.

All operations in learning and testing mode are realized in SVM using the kernel functions, which are defined as $K(\mathbf{x}, \mathbf{x}_i) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x})$.

The solution is given by means of the optimal weight vector w_{opt} as follows

$$w_{opt} = \sum_{i=1}^{N_s} \alpha_{si} d_{si} \phi(x_{si}). \quad (2)$$

where index s points to the set of N_s support vectors. The following relation

$$d_i \left(\sum_{j=1}^K w_j \phi_j(x_i) + w_0 \right) \geq 1 - \xi_i. \quad (3)$$

is fulfilled with the equality sign. The variables ξ_i are called slack variables. In turn, the output signal $y(\mathbf{x})$ of the SVM is given as the function of kernels, namely

$$y(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_{S_i} d_i(K \mathbf{x}_{S_i}, \mathbf{x}) + w_0 . \quad (4)$$

The most popular kernel functions used are the linear kernel, sigmoid kernel, polynomial kernel, RBF kernel, etc.

3 The Singular Value Decomposition (SVD) Method

The preprocessing stage of ECG waveforms is achieved here by means of using several methods. Among others was used the SVD method. It is a matrix factorization approach known from linear algebra, which reveals many important properties of the matrix. In this method the singular value decomposition of any $n \times m$ matrix A has the following form [4], namely

$$A = U \cdot S \cdot V^T . \quad (5)$$

where U is an $n \times n$ orthonormal matrix, whose columns are called the left singular vector of A , and V is an $m \times m$ orthonormal matrix, whose columns are called the right singular vectors of A . Matrix S is an $m \times m$ diagonal matrix. The diagonal elements of matrix S are listed in the descending order $s_1 \geq s_2 \geq \dots s_m \geq 0$ and called the singular values of A .

4 The Particle Swarm Optimization

The concept of the particle swarm optimization (PSO) was introduced by Kennedy and Eberhart [6], [7] as an efficient search and optimization technique. It was used here to reduce the dimensional size of the input data records.

In the PSO, a population of conceptual 'particles' is initialized with random positions X_i and velocities V_i , and function f is evaluated by means of using the particle's positional coordinates as the input values. In an n -dimensional search space, each particle has a position $X_i = \{X_{i1}, x_{i2}, \dots, X_{ij}, \dots, X_{in}\}$, a velocity $V_i = \{V_{i1}, \dots, V_{ij}, \dots, V_{in}\}$, where $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$. At each time-step the positions and velocities are adjusted, and the function is evaluated with new coordinates. The basic update equations for the d -th dimension of the i -th particle in the PSO may be given by

$$V_{id}^{k+1} = \omega V_{id}^k + C_1 \phi_1 (P_{lid} - X_{id}^k) + C_2 \phi_2 (P_{gd} - X_{id}^k) . \quad (6)$$

$$X_{id}^{k+1} = X_{id}^k + V_{id}^{k-1} . \quad (7)$$

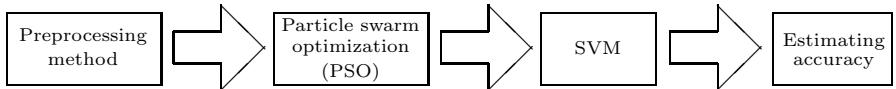


Fig. 1. A flowchart of the analysis of the ECG signals

where k denotes the iteration number, variables ϕ_1 and ϕ_2 are random positive numbers. They are drawn from a uniform distribution and defined by an upper limit ϕ_{max} , which is a parameter of the system. C_1 and C_2 are called acceleration constants, parameter C_1 is called a *cognition learning factor*, and C_2 is referred to as a *social factor*. ω is called an inertia weight and is employed in order to control the impact of the previous history of velocities on the current one. P_{li} is the local best solution found so far by the i -th particle, while P_g represents the positional coordinates of the fittest particle found so far in the entire community.

5 The Experiment Results

In our analysis we used the PTB Diagnostic ECG Database [8], which is publicly accessible in the PhysioNet [3]. This data set contains 549 records of heart overhauls of 290 patients divided into nine well-known diseases (see Table 1). The test group consisted of 209 men and 81 women. The average age of the men and women was 55.5 and 61.6 years, respectively.

Figure 1 shows the flowchart of our analysis. In phase 1 the ECG signals were preprocessed with the help of several preprocessing methods. The second phase, performed by means of the PSO method, was devoted to reducing the dimensional size of the input data. In the third phase the SVM method was used in order to analyse the ECG waveforms. The fourth phase was devoted to estimating the accuracy of our analysis.

The heart overhauls were represented in the ECG records in the form of 12-lead electrocardiogram signals, namely: $i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v6$. Each lead was described in terms of 1000 Hz frequency and the resolution of 16 bits. The time of the overhaul of each patient was increased from 32 to 120 seconds here. An example of 12-lead ECG waveforms is shown in Fig. 2. After refusing the all cases with 16 heart beats, nine groups of cardiac diseases were obtained (see Table 1). All of them are described here by means of 428 learning records. The data set prepared this way was analysed with the help of the preprocessing method. The software tool used in this study was the RapidMiner. We extended it some scripts described in the Python language.

We applied several preprocessing methods, namely: 1) the Fourier transform (FT), 2) the Haar wavelet, 3) the SVD matrix factorization, 4) SVD factorization for single ECG channel, 5) the averaging of the signal value, 6) the oscillation of a single ECG channel, 7) the oscillation of one signal channel discretized by means of the Discrete Fourier Transform (DFT).

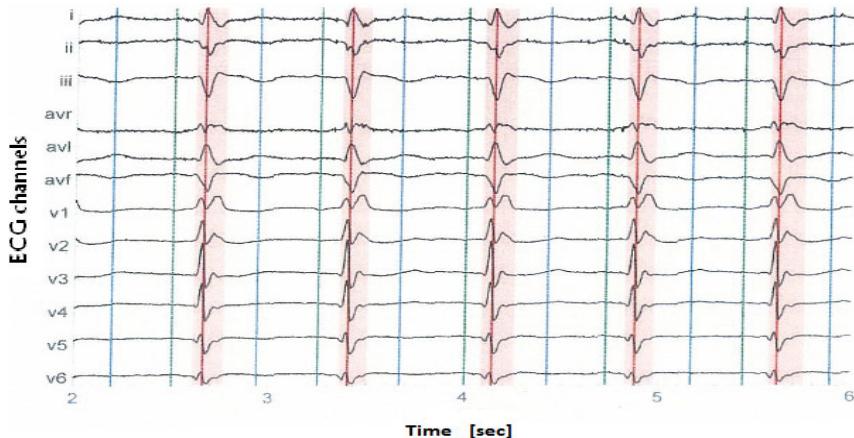


Fig. 2. Component wave detection in record of the PTB Diagnostic ECG Database

Ad 1) The Fourier transform of the ECG signal provides a Fourier array with a frequency range [10, 109] for the first channel, with frequency [10, 59] for the next two channels, and with frequency [10, 30] for all remaining channels here. The FT allows us to obtain 400 values of the data set record.

Ad 2) The Haar wavelet was used only for the first ECG signal. The remaining ECG signal channels were represented by successive terms of array. By means of using the Haar wavelet preprocessing method we obtained 180 array values.

Ad 3) The SVD matrix factorization allows us to detect and extract important relationships in the data. Here, we obtained 80 values for the first channel and 50 for the next channels. On the basis of the SVD matrices we selected the three most singular values. The SVD method provides us with 621 values.

Ad 4) The SVD matrix factorization for a single channel. This method allows us to obtain greater accuracy in the first ECG channel. By means of using the SVD we managed to build a matrix from which only 5 singular values are selected. This preprocessing method provides 455 values.

Ad 5) The averaging of the signal value was used for each ECG channel separately. When applying this preprocessing method we obtained 280 values for the ECG signal.

Ad 6) The oscillation of a single ECG channel signal determines 360 values.

Ad 7) The oscillation of a single ECG channel discretized by the DFT transform provides 360 values here.

In the second phase, we employed the PSO method in order to reduce the data input size dimension. By means of this method we increased the generalization of the classifier. Moreover, the PSO allows us to obtain at least one hundred values of the input data set.

The classifier proposed for the 12-lead ECG signal was obtained as a result of the RBF kernel function. The optimal value of σ was found in various training sessions. The σ value was determined as equal to 0.4998 and as 0.4385 for the

Table 1. Heart data sets used for testing the SVM method

No.	Data set	No. of patients	No. of train data
1	<i>Healt</i>	52	156
2	<i>Myocardial infarction</i>	148	104
3	<i>Cardiomyopathy/Heart failure</i>	18	44
4	<i>Bundle branch block</i>	15	43
5	<i>Dysrhythmia</i>	14	32
6	<i>Myocardial hypertrophy</i>	7	21
7	<i>Valvular heart disease</i>	6	10
8	<i>Myocarditis</i>	4	12
9	<i>Other distempers</i>	4	6

Table 2. Obtained confusion matrix

No.	Predicted class									Recognition (%)
	1	2	3	4	5	6	7	8	9	
1	153	16	5	8	8	0	2	0	2	78.87
2	2	85	3	2	2	0	1	0	0	89.47
3	0	1	35	0	0	0	1	0	0	94.59
4	1	2	0	33	0	0	0	0	1	89.19
5	0	0	1	22	0	0	0	0	0	95.65
6	0	0	0	0	0	21	0	0	0	100.00
7	0	0	0	0	0	0	6	0	0	100.00
8	0	0	0	0	0	0	0	12	0	100.00
9	0	0	0	0	0	0	0	0	3	100.00

PSO with the given size data input records and for the PSO without the given size data, respectively. The performance of the SVM classifier was estimated for all the classified heart diseases through the use of the confusion matrix (see Table 2). It can be seen that four of the eight diseases are predicted with 100% classification accuracy here. The total recognition rate (efficiency) of our SVM classifier is equal to 94.16%.

In the fourth phase we used the 0.632 bootstrap method in order to estimate the accuracy of our analysis. This method works as follows: the data set is sampled d times, with a replacement, resulting in a training set of d samples. As it turns out, on average 63.2% of the original data will end up in the bootstrap, and the remaining 36.8% will form the test set.

Thus, the overall accuracy (OA) of the data is then determined as

$$Acc(S) = \sum_{i=1}^k (0.632 \times Acc(S_i)_{test_set} + 0.368 \times Acc(S_i)_{train_set}) . \quad (8)$$

where $Acc(S_i)_{test_set}$ is the accuracy of the model obtained with a bootstrap sample i when it is applied to test set i . $Acc(S_i)_{train_set}$ is the accuracy of the model obtained with a bootstrap sample i when it is applied to the original set of data records.

Table 3. Overall accuracy of all the ECG channel signals

Preprocessing method	ECG channel group				
	(avl, avr, avf)	(i, ii, iii)	(v1, v2, v3)	(v4, v5, v6)	(i, avf, v3)
	OA (%)	OA (%)	OA (%)	OA (%)	OA (%)
1	61.68	61.25	58.43	62.86	63.32
2	54.92	56.56	63.78	70.32	61.26
3	44.38	43.72	53.25	48.85	44.60
4	47.43	43.46	47.19	45.55	43.46
5	61.71	61.23	70.10	72.89	66.14
6	60.04	61.91	65.67	67.78	61.91
7	79.66	79.89	80.16	80.87	79.89

We computed the overall accuracy of all the ECG channel signals. The obtained results of the OA parameter for each ECG channel signal are shown in Table 3. Further, we presented the graph obtained for each ECG channel signal depending on parameters C and γ . The first of them is a penalty parameter that determines the trade-off between the training error and the VC dimension of the model (Fig. 3), and the second is a parameter of the RBF function (Fig. 4).

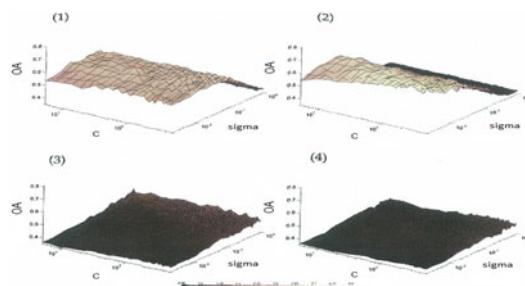
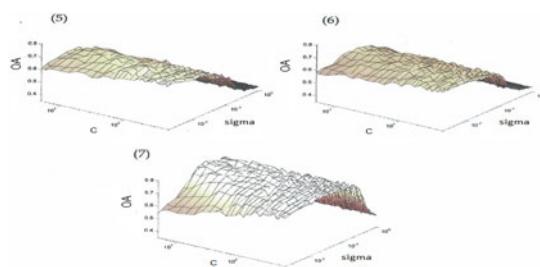
**Fig. 3.** Overall accuracy (OA) in dependence on C and σ parameters for 1 - 4 preprocessing methods, respectively**Fig. 4.** Overall accuracy (OA) in dependence on C and σ parameters for 5 - 7 preprocessing methods, respectively

Table 4. Comparative results of the ECG waveform classifiers

Method	Efficiency (%)
Proposed classifier	94.16
SOM-SVD [5]	92.2
Fuzzy Hybrid NN [12]	96.06
Discrete Wavelet Transforms [1]	96.6

We have not numerical results belong to other classifiers for the same data. But we can provide some results as efficiency (%) presented in the literature using different techniques for analysis of ECG waveforms. Comparison were done for following classifiers: expert system using Kohonen and SVM (SOM-SVD) [5], fuzzy hybrid neural network (FHNN) [12], Discrete Wavelet Transforms (DWT) [1] (see Table 4).

6 Conclusion

In this paper, we examined the possibility of using the SVM method and several preprocessing techniques for analysing 12-lead ECG waveforms. We tested our method using the PTB ECG Database covering different samples of QRS-complexes, and P- and T-wave morphologies. The preprocessing methods we used included the SVD matrix, the Haar wavelet, the Fourier transform, etc. The best result of the ECG analysis was obtained for the DFT transform.

References

1. Froese, T., et al.: Comparison of Extrasystolic ECG Signal Classifiers Using Discrete Wavelet Transforms. *Pattern Recognition Letters* 27, 393–407 (2006)
2. Gerardo, A.Z., et al.: Cardiac Sudden Death Risk Detection Using Hybrid Neuronal-Fuzzy Networks. In: 3rd Int. Conf. Electrical and Electronics Eng. (2006)
3. Goldberger, A.L., Amaral, L.A.N., Glass, L.: PhysioBank, PhysioToolkit, PhysioNet: Components of a New Research Resource for Complex Physiologic Signals (2000), <http://circ.ahajournals.org/cgi/content/full/101/23/e215>
4. Golub, G.H., van Loan, C.F.: Matrix Computations. John Hopkins University Press, Baltimore (1996)
5. Hu, Y.H., et al.: A Patient-adaptive ECG Beat Classifier Using Mixture of Experts Approach. *IEEE Trans. Biomed. Eng.* 44(9), 891–900 (1997)
6. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proc. of IEEE Int. Conf. on Neural Networks, pp. 1942–1948 (1995)
7. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann, Academic Press (2001)
8. Koch, H., Bousseljot, R., Kreiseler, D.: The PTB Diagnostic ECG Database, <http://physionet.org/physiobank/database/ptbdb/>
9. Llamedo, M., Martinez, J., Laguna, P.: A Multilead Wavelet-Based ECG Delineator Based on the RMS Signal. *Computers in Cardiology* 33, 153–156 (2006)
10. Lu, H.L., Ong, K., Chia, P.: An Automated ECG Classification System Based on a Neuro-Fuzzy System. *Computers in Cardiology* 27, 387–390 (2000)

11. Messadeg, D.J., Snani, C., Bedda, M.: An Approach for ECG Classification Using Wavelets and Markov Model. *Inf. and Comm. Tech.* 1, 1910–1913 (2006)
12. Osowski, S., et al.: Beat Recognition Using Fuzzy Hybrid Neural Network. *IEEE Trans. Biomed. Eng.* 48(11), 1265–1271 (2001)
13. Osowski, S., et al.: Support Vector Machines Based Expert System for Reliable Heartbeat Recognition. *IEEE Trans. Biomed. Eng.* 51, 582–583 (2004)
14. Song, M.H., Lee, J., Cho, S.P., Lee, K.J., Yoo, S.K.: Support Vector Machine Based Arrhythmia Classification Using Reduced Features. *International Journal of Control, Automation and Systems* 3(4), 571–579 (2005)
15. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, New York (1995)

Fuzzy Modeling of Digital Products Pricing in the Virtual Marketplace

Jaroslaw Jankowski, Jaroslaw Watrobski, and Mateusz Piwowarski

Faculty of Computer Science and Information Systems

West Pomeranian University of Technology

ul. Zolnierska 49, 71-410 Szczecin, Poland

{JJankowski,JWatrobski,MPiwowarski}@wi.zut.edu.pl

Abstract. There is observed the evolution of Internet systems that combine features of both technical and economic aspects. In this context, there appears a problem with respect to solutions related to modeling and the management of different aspects of system organization. The article thus employs hybrid system based on fuzzy modeling to identify dependencies between user's characteristics and digital goods evaluation for the purpose of developing a dynamic pricing system.

Keywords: hybrid systems, web social platforms, fuzzy modeling.

1 Introduction

Complex Internet platforms commonly act as dual systems by integrating technologies and methods of data processing with economic aspects. Many studies have emerged in both areas, including P. R. Messinger, E. Strouliab [16] and volumes edited by J. Spence [20] and Y. Sivan [19] published by Virtual Worlds Institute. W. B. Bainbridge emphasizes that a number of research methodologies, including formal experimentation and quantitative analysis of economic markets or social networks is explored in that area [3]. One of the functionality aspects of such systems involves digital product distribution platforms and marketplace environments [11]. It should be noted that this sector is developing very fast and is now one of the most promising directions for e-business. Global sales of virtual products are predicted to rise to 6 billion dollars per year by 2013 [22]. Applications have started to appear that integrate data processing for the purposes of electronic market monitoring, including multilevel price discrimination. When we look at the complex structure of this type of systems, different analytical approaches can be applied here. Integration of methods and their interactions must focus on global system behaviour, with the possibility to analyse components separately. Hybrid artificial intelligence systems can be applied here with broad areas of applications with backgrounds related to computational intelligence (fuzzy systems, rough sets, evolutionary algorithms, etc.) [5]. These techniques are applied in solving some real world problems in science, business, technology, and commerce [1]. Growth of web applications creates new areas of research related to hybrid systems, like analysis of navigation patterns [24], infrastructure [18] and web traffic [21]. Integration of

these methods with social platforms or virtual worlds was not discussed in literature yet as widely as other areas. These applications can be the next step towards the better understanding of the complexity of the systems, and they can help with solving decision problems related to them concerning areas of communication, social networks, online marketing and others. Proposed in this paper approach integrates several methods and systems within social platform towards supporting decision processes related to products pricing. It also shows new applications areas for hybrid intelligent systems. The main aim of research, of which results are included in this article, was the modelling of dependencies between digital product pricing and users' characteristics presented in an economic and social scope. When introducing new products into the virtual system it is hard to determine which price levels are acceptable by the users and what is the actual pricing of digital products (so called cold start effect). In order to conduct market sampling the auction system was used, by means of which the test set of products was introduced. This enabled the acquisition of knowledge about users' preferences and constituted a supply of a fuzzy system. As a result, the hybrid system was acquired, which integrates input from the auction system and historical data connected with users, and provides the basis to determine the pricing level for products for different sets of input parameters which can be used in the system of price discrimination. The research was conducted in a series of phases, which included preparing a methodological framework for fuzzy models, monitoring the system to integrate it with the online environment and data processing using fuzzy model validation.

2 Problem Statement and Areas of Research

The research focused mainly on systems of micro transactions that are presently integrated with many Internet applications and are constantly becoming more widely introduced business models in many areas. In the case of social networking platforms, which can form the basis of many systems that shape virtual economic systems, such analyses also entail the evaluation of connections within the social web. In the social network context, users are assumed to be provided with interactive messages that are adapted to his/her preferences. The following variables are taken into account, including the set of n digital products $O = [o_1, \dots, o_n]$ and the set of m users $U = [u_1, \dots, u_m]$ to whom interactive messages are sent in the form of a message sequence k_1, k_2, \dots, k_z . Using $d(k_i, p_j)$ for function modeling, the adaptation of message k_i to a vector of user preferences p_j may influence message recipients. There is a possibility of creating a certain kind of behaviour based on the determining supply vector $s = [s_1, s_2, \dots, s_n]$ and by defining for each user a price vector $p_{t,u}$ that changes over time. The price vector $p_u(t) = [p_{1,u}(t), p_{2,u}(t), \dots, p_{n,u}(t)]$ depends on the set of object parameters that describe user characteristics $u_j = [c_{j,1}, c_{j,2}, \dots, c_{j,3}]$ as well as object characteristics. The sale platforms have been developing, and users buy certain products in newly established environments of marketplace [11]. Such approach is nowadays implemented by many platforms, as well as systems. The effectiveness of sale systems is dependent on pricing politics that have been introduced, and on preferences of the audience. This was presented by Faber Novel Consulting that showed the dependency of sales value on the assumed pricing levels [6]. Liao C. and others, present in their studies how crucial

the loyal consumers in online social systems are and how significant it is to strengthen such loyalty [13]. There can be distinguished three categories of e-clients that were determined by Hernandez-Ortega B. and J. Jimenez-Martinez, that is: potential client considering first purchase, new client not participating in many transactions, and experienced one with large number of conducted transactions [9]. Flavian C. conducted studies that show how the transaction influences consumers' satisfaction [7]. Other fields of research relate to next actions concerning purchases. Lin H. and Wang Y. conducted broad studies in the scope of different forms of purchases [14]. In that research the applications of neuro fuzzy inference models in modeling of preferences and characteristics of digital products purchasers, was not detected. Most commonly, the applied methods assume deterministic character of input data and are basing on average values of individual variables.

3 A Structure of Fuzzy Inference Models and Sampling System

In the online functioning of complex technical and economic systems, there are occurrences that are hard to analyze with the use of classical methods because of the difficulty in determining relations between the input and output of the system, as well as due to a lack of mathematical identification models. Establishing number of interactions that are of specific type (e.g. purchases, social connections, other types of activity) of individual user can acquire values that are hard to be precisely determined, and strict limits among various values do not exist. In that case, linguistic values assumption, and determination of e.g. small, medium and high activeness of users, can be presented with appliance of fuzzy numbers. In this study, a fuzzy inference model was created together with systems that focus on modeling and determining audience characteristics and the influence of these characteristics on digital products pricing. Used fuzzy inference systems are based on fuzzy sets as defined by L.A. Zadeh [23]. The input value for fuzzy systems undergoes a fuzzy operation (or fuzzification); after inputting a fuzzy set, the output results in N fuzzy sets, which then undergo defuzzification as a block. One of the more commonly used models is Mamdani's model [15], which has been widely discussed in the literature, for example, by A. Piegat [17] and R. Babuska [2]. The main assumption involves mapping model inputs in the form of a vector X for output Y^* by obtaining the largest possible similarity and minimization of mean-square error given all mappings X to Y according to the system. Transformation procedures operating on input and output sets may use the operator MIN or the algebraic product operator PROD defined by Larsen [12]. In the proposed solution, the integration of an inference subsystem with a real system, which is supported by an Internet platform system, was assumed. Due to the large support sets, manual system parameterization was relatively difficult. Thus, the adaptive neuro-fuzzy inference system proposed by J.S.R. Jang was used, which is a neuron back-propagation algorithm based on the measurement systems and inference model developed by Takagi-Sugeno [10]. These types of systems use the properties of both fuzzy systems and neuron networks, which are characterized by, among others, parallel data information processing and the possibility of adaptation [4]. In Fig. 1, the five-layer architecture of the ANFIS system, which uses the inference mechanism from the Takagi-Sugeno model, is presented.

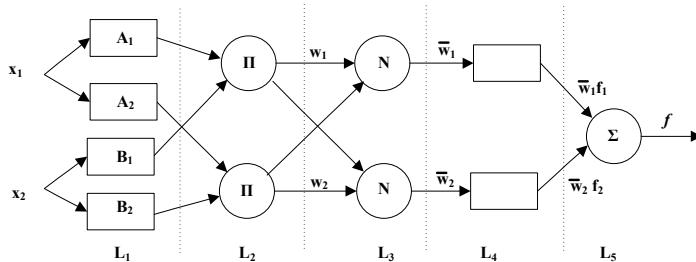


Fig. 1. Structure of adaptive neuro fuzzy inference system [11]

In the first layer, there is the introduction and parameterization of various premises. The second layer generates the support product and the levels for the activation of each rule are determined. In the output of the third layer, rule normalization occurs. The fourth layer determines the inference parameters, and the output of the fifth layer is an intensified signal. During the study, measurement systems were integrated with a real system, and a fuzzy subsystem was created. Fuzzy parameters were generated from time series data using method introduced by T. Frantti [8]. The hybrid system presented in the article can be defined as a part of interactive environment IE, implemented in the form of a social platform, where a set of n users, formalized as $U=u_1, u_2, \dots, u_n$, are social connections related and operate as potential customers interested in purchasing a digital product. For each user u_i , we can declare a set of m attributes such that $u_i=a_1, a_2, \dots, a_m$, which affect the decisions leading to a purchase. The operator of the social platform possesses a set of k available products $P(p_1, p_2, \dots, p_k)$ and must individualize the selling prices and evaluations for each product through discriminatory analysis. For a set of test products $S=s_1, s_2, \dots, s_l$, we define a set of samples l to analyze the preferences and pricing levels for each different product type. The samples are introduced via a data sampling control system DS, which uses electronic auctions to extract data about preferences, pricing levels and demand for different products. We define subsystems for the control system DS, i.e., $DS=[A, DB, MV]$, in the form of an auction platform A, a transactional database DB, and a fuzzy model validation subsystem MV. Data about product pricing and preferences allow us to define a preference vector for a user u , which is dependent on input attributes and product characteristics. For example, the process vector assigned for user u can be defined as follows: $p_u(t) = [p_{1,u}(t), p_{2,u}(t), \dots, p_{n,u}(t)]$. Values from the sampling system are transferred to the database DB and are then used as inputs for d fuzzy models in the MV subsystem, i.e., m_1, m_2, \dots, m_d . At the output, we obtain a set of rules R_s , which can be registered in the rule repository RDB. Rules from the RDB repository are extracted and used in the selection of the pricing level in accordance with the user input parameters $u_i=a_1, a_2, \dots, a_m$. The next part of the article is dedicated to the validation of the data sampling system and the fuzzy models for the inference module.

4 Integration of Sampling System with an Online Environment

In order to acquire users' preferences and to determine price levels the test set was prepared including nine digital products. Pricing in systems of that type mainly depend on

the uniqueness of a product and number of items made available to the users. Therefore, certain products within the system were provided in a limited number, which mainly determined the price offer and influenced the product's attractiveness. The results of the studies conducted in the scope of virtual world platform are presented below and they are one of the fields of virtuLabs, a project which is focused on development of methods and analytical systems dedicated for Internet platforms. In the first phase the prime auction was conducted during which participants bid objects entered into the system. The transactions were made for nine digital objects available in limited versions marked with the following symbols s_1-s_9 . Achieved pricing and number of top offers for individual objects is presented in the Tab. 1, where Q - quantity of incoming offers, E - average object price, L - limit for digital objects distribution.

Table 1. Offers distribution in the primary auction

Product	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
Q	1251	69	564	816	34	313	170	228	653
E	221	193	151	105	157	170	195	137	98
L	5	9	13	13	28	37	49	62	99
	6100	2835	1500	2000	1111	2000	1064	500	781
	5848	2639	1500	1500	700	907	985	500	689
	5848	2500	1500	1330	700	675	700	250	680
	5000	2180	1500	1302	450	645	500	203	500
	4500	2000	1500	1300	411	550	472	200	50
	...								

Participants were provided with 315 virtual objects in the scope of prime auction. In the next period, users were able to sell digital objects on secondary auctions. In the analyzed period 842 auctions were started. In the next phases the transaction fluidity significantly improved. At Fig. 2 there is a quantitative arrangement of offers in the individual sub periods.

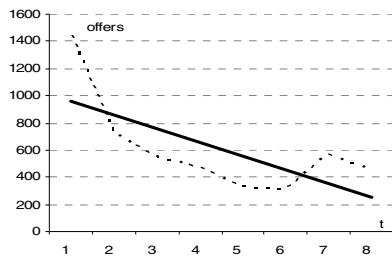


Fig. 2. Quantitive arrangement of offers

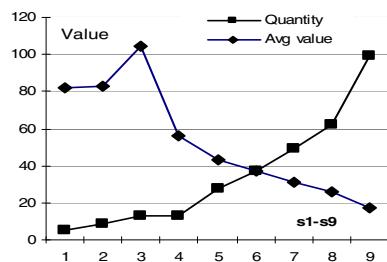


Fig. 3. Dependency of average pricing on number of products on the market

The biggest number of offers was placed for object s_1 which due to a limited number was more attractive and obtained the highest pricing. Average pricing was inversely proportional to the number of available objects in the marketplace system, as

shown in Fig. 3. Most commonly the subjects of transactions were objects s7, s8 and s9. During the auction their participants placed 7325 purchase offers. The average pricing was inversely proportional to number of available objects in marketplace system. In the analyzed period there was a big variability of a number of new auctions and auctions finalized with transactions. In the whole period, 351 auctions were successful. The growth of price was significant, because it appeared together with a decrease of new auctions, which resulted in lower supply and higher number of offers devoted to individual auction. The conducted market sampling shows the big changeability of pricings and confirms the need to introduce the price discrimination methods, the aim of which is to match product price with users' characteristics and preferences. In the next phase the basic aim was to determine the influence of features characterising the user in economic and social scope on the pricing of digital products. In the process of parameterization of system inputs, the parameters characterising a certain user in the economic and social scope were assigned to each pricing.

5 Modeling of the Digital Products Pricing with Fuzzy System

Data from sampling systems were connected with user's parameters in order to determine dependencies between user's parameters and product pricings. The main task of the inference model is to reflect the pricing level depending on the users' parameters in economic and social scope. The acquired inference system indicates how the pricing level depends on the users' parameters and how it can be used in the system of price discrimination, which on the basis of users' parameters will generate price offers. The main goal was to detect relations between users' parameters and levels of accepted prices and willingness to buy virtual objects. The parameters that characterize users were divided into two groups: social and economic. On this basis two particle models were established, which aimed to explain the dependency of purchase decision making on parameters and user. In the first phase the users' characteristic representation model was established in a scope of economic conditions, and number of activated services and their type as a vector $E_i = [e_1, e_2, e_3]$, where e_1 - number of transactions, e_2 - number of virtual currency units, e_3 - number of premium services activations. For calculation purposes the MatlabTM environment was used. In the next stage neuro fuzzy model with three inputs and one output was prepared. Input parameters reflected the individual components of vector $E_i = [e_1, e_2, e_3]$, which values were acquired from empirical data. Training set included data 173 transactions and connected users' characteristic vectors. Fig. 4 shows the arrangement of membership function for individual system inputs. The accuracy of the model was tested on four different shapes of membership function, but the highest accuracy was obtained for Gauss function, and such arrangement was assumed in final model. For every input the number of membership functions was also analyzed. Satisfying results were obtained in 3 membership functions for individual inputs. Training was tested on 20, 50, 10, 200 and 300 epochs. Teaching over 100 of epoch provided improvements and was stopped in order to avoid an over-fitting effect. Fig. 5 shows the response system surface depending on input, for input e_1, e_3 which shows the dependencies between inputs and output of system and determines the influence of individual components of users characteristics vector on digital object pricing. On the systems output pricing levels

were presented for digital products in the amount from 0 to 2000 of virtual currency units (VC). The response depended on the number of Premium account activations and number of transactions previously realized within the system. The surface of system responses indicates the highest pricing at the level of 2000 VC of products made by a number of users who activated up to 50 Premium accounts and who realized a small number of transactions. With a number of transactions from 2000 to 3000 the value of pricing decreases to the level of 1500 VC. The acquired response surface provides the basis to determine price levels for users with different values of input parameters.

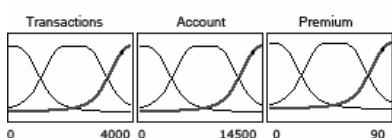


Fig. 4. Membership functions for input variables

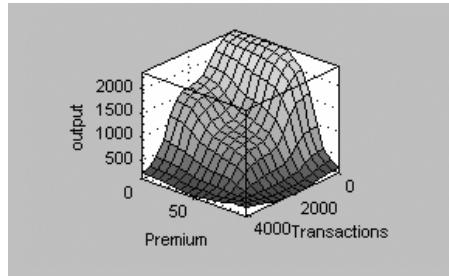


Fig. 5. Surface plot showing relationship between input and output parameters for E_i model

Second model includes social characteristics of users that relate to activeness and localization in social network. Social characteristics of i -th user are described by a vector $S_i = [s_1, s_2, s_3]$, where s_1 - level of social network connections, s_2 - users activeness in internal communication (out coming messages), s_3 - users activeness in internal communication (incoming messages). Fig. 6 shows five layer structure of ANFIS model, where for each block training with a back propagation and hybrid method was applied. Better results were obtained with back propagation method.

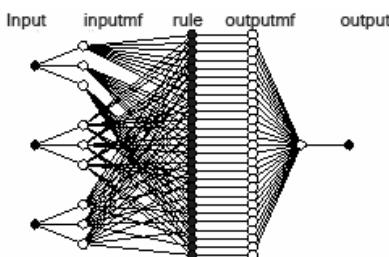


Fig. 6. Five layers structure of reasoning system

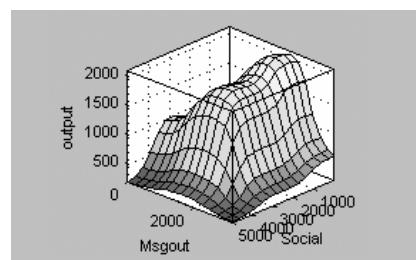


Fig. 7. Surface plot showing relationship between input and output for S_i model

Data from 173 registered transactions in digital object distribution system was taken into consideration. Fig. 7 presents system answer for trained network with similar parameters as for economical parameters. For a chosen input level, this model determines values for output and can be used to make predictions about users in real system. The results indicate the presence of connections between user parameters, user decision-making and the level of object pricing. Research has shown that fuzzy models can be a useful part of inference systems dedicated to modelling relation between user parameters and digital good pricing in a form of hybrid system. A two-layer integrated auction platform can be used as a market monitoring system and can thus help determine user preferences. Such an approach is especially relevant for social networking platforms, virtual worlds, and online games in which operators introduce digital products such as avatar extensions. Knowledge achieved in such a process can be used when introducing the next round of products in order to increase revenue levels and user satisfaction. The acquired results confirm the appearance of dependencies between pricing levels for digital products and users' parameters considered in economic and social scope. The acquired answer surfaces for both approaches enable us to reflect on the dependencies in a system and provide possibilities to generate dedicated price offers on the basis of data acquired from the inference system.

6 Summary

Presented approach to modeling the relation between digital products pricing and users' characteristics by taking into account the uncertainty of measurement data is based on hybrid system. The review of the literature suggests that existing solutions for dynamic pricing systems tend to be deterministic and not many areas related to social platforms and hybrid intelligent systems are explored. Proposed solutions identify input data sources that can be used for fuzzy representation and the nature of the data justifies the use of fuzzy modeling. Thus, this article presents a system of inference based on this approach. The presented market sampling system uses an auction platform to allow data retrieval in order to model the characteristics of consumers; moreover, the model has been empirically verified. Data obtained after fuzzification were used as an input layer for a fuzzy model. Specifically, parameters reflected the relationship between user characteristics and the level of pricing for digital products. Moreover, the model takes into account the specificities of a highly dynamic environment. The proposed procedure for using a fuzzy model as a part of hybrid system to understand online virtual environments can be used in other areas. The obtained results show that adopting such a solution in the context of dynamic pricing systems has both practical and theoretical potential.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72(13-15), 2775–2784 (2009)
2. Babuska, R.: Fuzzy Modeling for Control. Kluwer Academic Publishers, Boston (1998)
3. Bainbridge, W.S.: The Scientific Research Potential of Virtual Worlds. *Science* 317(5837), 472–476 (2007)

4. Cichocki, A., Unbehauen, R.: Neural Networks for Optimization and Signal Processing. John Wiley & Sons, Chichester (1994)
5. Corchado, E., Abraham, A., Ponce, A.C., Ferreira de Carvalho, L.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)
6. FaberNovel Consulting: Social Network websites: best practices from leading services. Research paper, FaberNovel Consulting (2007)
7. Flavian, C., Guinaliu, M., Gurrea, R.: The role played by perceived usability, satisfaction and consumer trust on website loyalty. *Information and Management* 43(1), 1–14 (2006)
8. Frantti, T.: Timing of fuzzy membership functions from data. Technical report, University of Oulu, Oulu (2001)
9. Hernandez-Ortega, B., Jimenez-Martinez, J., Jose Martin-DeHoyos, M.: Differences between potential, new and experienced e-customers, Analysis of e-purchasing behavior. *Internet Research Journal* 18(3), 248–265 (2008)
10. Jang, J.S.R.: ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Transactions on Systems, Man and Cybernetics* 23(3), 665–685 (1993)
11. Korolom, M.: The future of virtual goods,
<http://www.hypergridbusiness.com/2009/09/the-future-of-virtual-goods>
12. Larsen, P.M.: Industrial Applications of Fuzzy Logic Control. *International Journal of Man-Machine Studies* 12(1), 3–10 (1980)
13. Liao, C., Palvia, P., Lin, H.N.: The roles of habit and web site quality in e-commerce. *International Journal of Information Management* 26(6), 469–483 (2006)
14. Lin, H., Wang, Y.: An examination of the determinants of customer loyalty in mobile commerce contexts. *Information and Management* 43(3), 271–282 (2006)
15. Mamdani, E.H.: Applications of Fuzzy Set Theory to Control Systems: A Survey. In: Gupta, M.M., Saridis, G.N., Gaines, B.R. (eds.) *Fuzzy Automata and Decision Processes*, pp. 77–88. North-Holland, Amsterdam (1977)
16. Messingera, P.R., Strouliab, E., Bonea, K.M., Niud, R.H., Smirnova, K., Perelgute, S.: Virtual worlds - past, present, and future: New directions in social computing. *Decision Support Systems* 47(3), 204–228 (2009)
17. Piegat, A.: *Fuzzy Modeling and Control*. Physica Verlag, Heidelberg (2001)
18. Schuler, W.H., Bastos-Filho, C.J.A., Oliveira, A.L.I.: A novel hybrid training method for hopfield neural networks applied to routing in communications networks. *International Journal of Hybrid Intelligent Systems* 6(1), 36–41 (2009)
19. Sivan, Y.: Technology, Economy and Standards in Virtual Worlds. *Journal of Virtual Worlds Research* 2(3), 1–8 (2009)
20. Spence, J.: Virtual Worlds Research: Past, Present and Future. *Journal of Virtual Worlds Research* 1(1), 1–12 (2009)
21. Wanga, X., Abrahamb, A., Smitha, K.A.: Intelligent web traffic mining and analysis. *Journal of Network and Computer Applications* 28, 147–165 (2005)
22. Yarow, J.: U.S. Virtual Goods Revenue Ready To Explode,
<http://www.businessinsider.com/chart-of-the-day-us-virtual-goods-revenue-estimates-2009-11>
23. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8(3), 338–353 (1965)
24. Zehraoui, F., Kanawati, R., Salotti, S.: Hybrid neural network and case based reasoning system for Web user behavior clustering and classification. *International Journal of Hybrid Intelligent Systems* 7(3), 171–186 (2010)

An Algorithm Based on Genetic Fuzzy Systems for the Selection of Routes in Multi-Sink Wireless Sensor Networks

Lliam B. Leal, Marcus Vincius de S. Lemos, Raimir Holanda Filho,
Ricardo A.L. Rabelo, and Fabio A.S. Borges

Mestrado em Informtica Aplicada – Universidade de Fortaleza (UNIFOR)
Fortaleza – CE – Brasil

Centro de Ensino Unificado de Teresina – CEUT

Teresina – PI – Brasil

{lilihamleal,marvinlemos,fabbioanderson}@gmail.com,
raimir@unifor.br,
ricardor_usp@yahoo.com.br

Abstract. Wireless sensor networks (WSNs) are composed of sensor nodes in order to detect and transmit features from the physical environment. Generally, the sensor nodes transmit information to the special node called sink. Some recent researches have led to the selection of routes in sensor networks with multiple sink nodes. The approach proposed by this paper presents the application of Genetic Fuzzy System (GFSs) for the selection of routes in WSNs, in order to make the communication between multiple sensor nodes and sink nodes. The results obtained through simulations demonstrated a sensor network with a longer lifetime, through the choice of the adequate sink used for sending packets through the network in order to find the best routes.

Keywords: Fuzzy Inference System, Genetic Algorithms, Wireless Sensor Networks, Fuzzy Genetic Systems, Wireless Sensor Networks Routing.

1 Introduction

In recent years we have witnessed a considerable increase in researches involving Wireless Sensor Networks (WSNs) due to their applicability to various areas like safety, health, agriculture, smart environments, industrial automation, among others [19], [10]. Some of the most common paradigms of communication in WSNs involve the communication of multiple sensor nodes placed in an observation area reporting information to a special node named sink (many-to-one). In [1] it is said that the use of an unique sink represents a bottleneck in a network, especially for applications in real time and continuous data flow. In this sense, some researches have directed studies to the use of multiple sinks, in which communication addresses an alternative paradigm composed of multiple sensor nodes to multiple sinks (many-to-many). The approach used in this work focuses

on the communication between the sensor nodes and the multiple sink nodes of a WSN.

The methodology implemented for selection routes in WSN combines fuzzy inference systems of Mamdani [1] and Genetic Algorithms (GA) [3]. We employed a fuzzy inference system of Mamdani, in that we tried to model approximate reasoning [12] in order to imitate the human ability to make reasonable decisions in an environment of imprecision, uncertainties and noise. Fuzzy inference systems of Mamdani are used to determine the most appropriate sink node through consideration of some characteristics of the sensors network, such as energy and number of hops. Genetic Algorithms are employed to obtain the optimal adjustment of Mandanis fuzzy inference system parameters. By applying GAs, we intend to achieve both a fuzzy database and a fuzzy rules base to maximize performance of the application of Mamdanis inference system in the selection of routes in WSN. Therefore, the obtained fuzzy system should be able to help the routing algorithm to choose the best communication path between the sensor nodes and the sink nodes so as to maximize the network lifetime. It is important to highlight that GAs are global optimization algorithms, based on natural selection mechanisms and genetics, which have shown themselves efficient in a wide variety of problems, in that they override many limitations found in the traditional search/optimization methods [4]. The systems achieved from the integration between the fuzzy inference models and Genetic Algorithms are called Genetic Fuzzy Systems (GFSs) [5].

2 The Algorithm Proposed to Assist the Choice of the Best Route in Multi-Sink Environments

2.1 General Aspects

In our proposal we consider a network in which the sensor nodes are positioned so as to reach the coverage of the whole area and the multiple sinks are arranged according to the network design. It is noteworthy that the sink nodes are devices with superior characteristics to the sensor nodes, having no energy limitations.

Each sink node is responsible for receiving notification messages of events occurring anywhere on the network. For such, each sensor node must select the most appropriate sink at a given moment, taking characteristics of the network into consideration, such as energy, number of hops, collisions and delay. We must emphasize that on this paper we considered initially only two characteristics, energy and number of hops, main metrics considered in WSNs [2].

It is important to highlight that choosing a routing protocol is necessary, since our solution will work together with it. The choice of the routing protocol is made according to the needs of the network designer, being required only that the protocol be capable of providing multiple paths (multipath). To test, we used the routing protocol Directed Diffusion [9].

In our experiments, we use the Sinalgo simulator [17]. Sinalgo is a framework implemented in Java language, for testing and validating the networks algorithms. Contrary to other tools such as the *Network Simulator 2* [13], which

allows the simulation of networks in several layers of the protocol stack. Our approach focuses the use of Sinalgo to verify the efficiency of the algorithms to selection of routes.

3 Genetic Fuzzy System

3.1 Aspects of Implementation

The implemented fuzzy inference system has two input linguistic variables, Energy and number of hops, and a linguistic output variable, the Fuzzy Level (FL). The syntax of the rules of the fuzzy system is represented by the following linguistic conditional declarations:

- Rule 1: If (Energy is A_1) and (Number of hops is B_1), Then (FL is C_1) or
- ...
- Rule n_r : If (Energy is A_p) and (Number of hops is B_q), Then (FL is C_r) .

where: n_r is the number of rules; A_j , B_k , C_l are the primary terms (linguistic values) associated with the linguistic variables, Energy, number of hops and FL, respectively; p , q and r correspond to the number of primary terms of the linguistic variables Energy, Number of hops and FL, respectively.

Thus, we have a Mamdanis fuzzy system, consisting of N_r disjunctive inference rules. The rules are inferred in a parallel way, without considering the order in which they are implemented. The inference of each rule consists of the evaluation of the antecedent, followed by the application of the implication operator to determine the consequent fuzzy set. The defuzzification procedure obtains the FLs numeric value. The implication operator used was Mamdanis minimal operator; the aggregation of the consequents was performed by the maximum operator and the defuzzification method employed was the area center method [15].

The approach adopted in this paper proposes the use of GAs to determine the number of primary terms for the linguistic variables, the adjustment of the membership functions associated with each primary term, and the set of linguistic rules of Mamdanis fuzzy system, simultaneously.

The first aspect to be considered in the use of GAs for the solution of a problem is the chromosomal representation of the problem, so that the GAs can solve it appropriately [16]. In this study the chromosomes have information related to the number of primary terms and the membership functions of the input and output linguistic variables, as well as information on the rules bases of the fuzzy inference system.

The chromosome structure to store the parameters of the fuzzy inference system was divided into three parts. The first part codifies the amount of primary terms of the linguistic variables. This part of the chromosome has 9 bits, where each set of 3 bits represents the number of primary terms of a linguistic variable. The second part stores information of the fuzzy rules base. According to experience of authors in relation to the behavior of WSNs, the top number of primary terms of each linguistic variable was defined as being equal to 5, and the amount of rules is then set by a multiplication between the numbers of primary

terms of the input linguistic variables (Energy and Number of hops). Thus, the maximum number of rules in the system is equals to 25. Therefore, the second part is represented by 75 bits, and each set of 3 bits represents the primary term associated with a rule of the inference system. The third part codifies the parameters of the membership functions of each linguistic variable of the system. As membership functions have continuous parameters that require a great precision, the representation of the chromosome by real numbers was adopted [18], instead of binary representation used in the two first parts. The membership functions used for the linguistic variables are triangular (Figure 1).

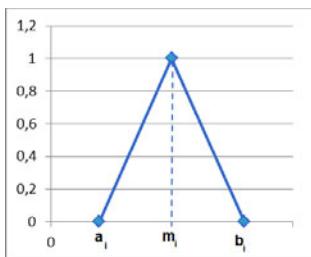


Fig. 1. Triangular membership function

However, instead of the chromosomes storing the values a_i , m_i and b_i for each membership function, they store the adjustment coefficients (parameters) δ_i and η_i . Each membership function i can be assembled from the following equations [14]:

- $a_i = (a_i + \delta_i) - \eta_i$,
- $m_i = (m_i + \delta_i)$,
- $b_i = (b_i + \delta_i) + \eta_i$.

The coefficient δ_i is responsible for moving the membership function to the right or to the left, while the coefficient η_i can expand or shrink the support of the membership function [20] in relation to its discourse universe. As each linguistic variable has 5 primary terms at most, and each membership function associated with a primary term is represented by the two parameters δ_i and η_i , 10 genes, at most, are necessary to represent the membership functions of a linguistic variable of the applied fuzzy inference system. Therefore, the third part of the chromosome has size 30. As each chromosome stores the information about the parameters of the inference fuzzy system, we have one chromosome of 114 ($9 + 75 + 30$) genes. The GA implemented has adopted a population of 80 individuals (chromosomes).

After the chromosome representation definition, the GA design focuses on the specification of an evaluation function. Since the goal of the fuzzy system is to help the routing algorithm choose the best way to communicate between the sensor nodes and the sink nodes, this work employed the time it takes for the

first sensor node of the network to die as evaluation function. Therefore, there is a maximization problem whose objective is to find a configuration to the fuzzy system that selects the best routes to maximize the lifetime of the network.

After the attribution of the fitness index by the evaluation function, the selection process chooses a subset of individuals of the current population to compose an intermediate population in order to apply the genetic operators. The selection method adopted in this work was the tournament method. It is worth mentioning that the size of the adopted tournament was set equal to 2. In combination with the selection module an elitist strategy was used with the maintenance of the best individual from one generation to the other.

To evolve the population go through an evolution one should apply the genetic operators (crossover and mutation). In this study, we used discrete crossover [6]. The mutation genetic operator [8] is necessary to introduce and maintain genetic diversity of the population through randomic changes of genes within the chromosomes, what provides a means for incorporation of new genetic characteristics within the population. Therefore, mutation ensures the possibility of reaching any point of the search space, besides to avoid to get trapped in local minima. However, mutation is applied less frequently than crossover in order to preserve the exploration-exploitation-relationship [7].

Several criteria may be applied to finalize the execution of a GA. In this work the maximum limit of generations fixed in 100 interactions was adopted as stopping criterion.

4 Simulations, Results and Discussions

This section presents comments about the simulations carried out to validate the selection of route algorithm proposed using Genetic Fuzzy Systems. This section also shows the results obtained through simulations.

4.1 Network Characteristics

The main characteristics of the network are:

1. Topology: the simulated network is steady and includes only two types of nodes: sink nodes and sensor nodes. The sensor nodes have similar characteristics, featuring a flat network, where each node of the network has a single identifier and steady radio range.
2. Scenarios used: in our simulations we used three scenarios with similar environment. All scenarios are usually composed of 100 sensor nodes distributed uniformly throughout the network and for the scenarios that use multiple sinks we used 4 sink nodes located at the end of the network initially, with a range area of 700 m x 700 m. We configured a sensor node located in the center of the network to transmit packets continuously at a ratio of one packet per second in the three scenarios.

Next we describe the simulated scenarios:

- Scenario 1: in this scenario we used the Directed Diffusion routing protocol in its traditional form, without the use of fuzzy inference systems and without multiple sinks.
- Scenario 2: the application through Mamdanis fuzzy inference system influences the routes formation through the Directed Diffusion protocol. Nevertheless, the adjustments of the fuzzy inference system is made in an empiric way.
- Scenario 3: just as in scenario 2, the application of Mamdanis fuzzy inference system influences the formation of routes through the Directed Diffusion protocol. Notwithstanding, in this scenario, the parameters of the fuzzy inference system are not adjusted empirically anymore, but by using Genetic Algorithms, characterizing the approach as fuzzy-genetic.

4.2 Results and Discussions

The results of the proposed approach application are presented based on three metrics. The metrics are:

- First node death time: it expresses the death time of the first node in the network. Here, we intend to analyze how long sensor nodes remain living;
- Network death time: it records how long the network remains alive, that is, how long the network will be able to keep the necessary communications active.
- Sink nodes number: in this metric, we evaluate how the amount of sink nodes influences our work;

We evaluated the behavior of the sensors in each scenario according to the three metrics addressed. For the first metric, first node death time in the network, we can see from Figure 2 that in scenario 1, where we used the Directed Diffusion protocol in its traditional form, without the use of fuzzy inference systems and without multiple sinks. The results showed weakness in the lifetime of the network, where a sensor node participates actively in a route in use. In scenario 2 we have a solution that uses fuzzy inference systems adjusted empirically, associated with the use of multiple sinks. In this scenario, the gain compared with scenario 1 is approximately 234%. Finally, in scenario 3 we have the representation of the proposed solution that uses a genetic fuzzy system and multiple sinks in which the gain when compared with scenario 1, approaches 373%. This gain was due to the use of multiple sinks, increasing the number of routes each node can use for the transmission of data over the network and the use of the genetic fuzzy system which makes the node avoid the bad routes (low energy, long distances).

We represented in Figure 3 the network death time, time when the network communication is discontinued due to the death of some sensor nodes that compose the routes of the sender. In the column representing scenario 1 (without fuzzy inference system and without multiple sinks), we have that the network

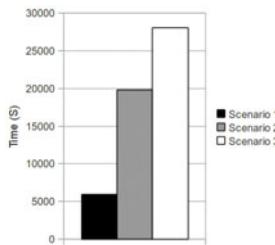


Fig. 2. Death time of the first node of the network

death time corresponds to the networks first node death time because this network does not have multiple routes and multiple sinks. In scenario 2, the network remains active for a long period of time. Approximately 478% higher than the scenario 1, this gain was achieved by using multiple routes, associated with the use of empirical fuzzy inference in the best routes selection process at a certain moment. The great difficulty encountered in this simulation scenario was due to the empirical adjustment of the inference system that consumed much time, in that they were carried out based on trial and errors. Still, it was not possible to firmly tell whether the chosen parametres of fuzzy inference system were the best. We obtained in scenario 3, an uptime of the network higher than all simulated scenarios; about 541% higher than scenario 1, this gain was promoted through the use of genetic fuzzy system and multiple sinks.

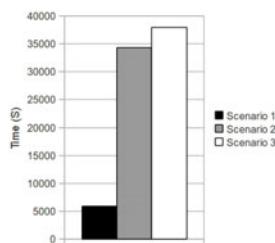


Fig. 3. Death Time the Network

5 Conclusions

This paper proposes an algorithm for the selection of routes with the use of fuzzy inference systems for the WSN with multiple sinks. The main objective of the algorithm is to extend the networks lifetime which can be achieved by making the nodes avoid bad routes during the sending of packets through the network. For such, we applied fuzzy inference systems in order to build a knowledge base which makes the process of finding the best routes possible.

Results obtained through simulations, demonstrated that our approach is feasible increasing the networks lifetime satisfactorily. In future works we intend to

analyze the behavior of algorithms in more complex scenarios, as well as select new linguistic variables for the fuzzy inference.

When you opt for a fuzzy inference system to determine the selection of routes in a wireless sensor network with multiple sink nodes, you have an action/control strategy that can be monitored and interpreted, even from the linguistic viewpoint. Another advantage found in using fuzzy inference systems during the development of this work refers to the inclusion of the authors experience in the definition of some parameters of the implemented fuzzy system. That experience could be used directly to aid the construction of the rules base and the initial definition of the primary terms (fuzzy sets) of the linguistic variables.

References

1. Boukerche, A., Martirosyan, A.: An energy efficient and low latency multiple events' propagation protocol for wireless sensor networks with multiple sinks. In: PE-WASUN 2007: Proceedings of the 4th ACM Workshop on Performance Evaluation of Wireless ad Hoc, Sensor, and Ubiquitous Networks, pp. 82–86. ACM, New York (2007)
2. Garca-hernndez, C.F., Ibargengoytia-gonzlez, P.H., Garca-hernndez, J., Prez-daz, J.A.: Wireless sensor networks and applications: a survey (2007)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
4. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. Wiley, New York (1998)
5. Herrera, F.: Genetic Fuzzy Systems: Taxonomy, Current Research Trends and Prospects, vol. 1, pp. 27–46. Springer, Heidelberg (2008)
6. Herrera, F., Lozano, M., Snchez, A.M.: Hybrid crossover operators for real-coded genetic algorithms: an experimental study, vol. 9, pp. 280–298 (2005)
7. Herrera, F., Lozano, M., Verdegay, J.: Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis, vol. 12, pp. 265–319. Springer, Heidelberg (1998)
8. Hinterding, R., Gielewski, H., Peachey, T.: The Nature of Mutation in Genetic Algorithms. In: Proceedings of the Sixth International Conference on Genetic Algorithms, Citeseer, pp. 65–72 (1995)
9. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking, vol. 11, pp. 2–16. IEEE Press, Piscataway (2003), <http://dx.doi.org/10.1109/TNET.2002.808417>
10. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J.: Wireless sensor networks for habitat monitoring. In: WSNA 2002: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 88–97. ACM, New York (2002)
11. Mamdani, E.H.: Application of Fuzzy Algorithms for Control of Simple Dynamic Plant, vol. 121, pp. 1585–1588 (1974)
12. Mamdani, E.H.: Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis, vol. 26, pp. 1182–1191. IEEE Computer Society, Washington, DC, USA (1977)
13. NS2 (2008), http://nsnam.isi.edu/nsnam/index.php/main_page
14. Park, D., Kandel, A., Langholz, G.: Genetic-based New Fuzzy Reasoning Models with Application to Fuzzy Control, vol. 24, pp. 39–47 (1994)

15. Ross, T.J.: *Fuzzy Logic with Engineering Applications*. Wiley, Chichester (2004)
16. Shi, Y., Eberhart, R., Chen, Y.: Implementation of Evolutionary Fuzzy Systems, vol. 7, pp. 109–119 (1999)
17. sinalgo (2010), <http://disco.ethz.ch/projects/sinalgo/index.html>
18. Srikanth, T., kamala, V.: A real coded genetic algorithm for optimization of cutting parameters in turning, vol. 8, pp. 189–193 (2008)
19. chee Tseng, Y., chiun Wang, Y., wu Yeh, L.: ipower: An energy conservation system for intelligent buildings by wireless sensor networks (2009)
20. Zimmermann, H.J.: *Fuzzy Set Theory – and its Applications*. Kluwer Academic Publishers, Dordrecht (2001)

Hybrid Analytical and ANN-Based Modelling of Temperature Sensors Nonlinear Dynamic Properties

Lidia Jackowska-Strumillo

Computer Engineering Department, Technical University of Lodz,
ul. Stefanowskiego 18/26, 90-924 Lodz, Poland
lidia_js@kis.p.lodz.pl

Abstract. The paper presents new methods for modelling of temperature sensors' dynamics by means of Artificial Neural Networks (ANN) and hybrid analytical-neural approach. Feedforward multilayer ANN and a moving window method, as well as Recurrent Neural Networks (RNN) are applied. The proposed modelling techniques were evaluated experimentally for two small platinum Resistance Temperature Detectors (RTDs) immersed in oil. Experiments were performed in temperature range, for which the sensors characteristics are nonlinear. The proposed ANN-based and hybrid analytical-neural models were validated by means of computer simulations on the basis of the quality of dynamic errors correction. It was shown that in the process conditions for which classical methods and linear models fail, the application of ANNs and hybrid techniques which combine soft and hard computing paradigms can significantly improve modelling quality.

Keywords: dynamic sensor's model, artificial neural networks, hybrid analytical-neural model.

1 Introduction

Dynamic measurements play an important role in many industrial processes. Determination of a dynamic model of a sensor is essential for designing its optimal corrector and improving the accuracy of dynamic measurements, as well as optimizing the process control. Particularly, temperature sensors working in harsh environmental conditions usually have poor dynamic properties and their application is limited to measurements of slow temperature changes. Theoretical and experimental methods of temperature sensor identification and dynamic error correction were proposed by many researchers, but most of the studies concentrate on linear models of the sensors and their correctors [1-3]. However, such approaches simplify the modelling problem and give satisfactory approximations only in narrow temperature ranges. In wider temperature ranges, nonlinear dynamic models of the sensors, e.g. analytical [4] or ANN-based [5], must be taken into account. This is particularly important in scientific research and industrial applications in which temperature measurement and control is a critical task, e.g. in electronics, chemistry, food engineering and biotechnology.

Hybrid modelling approach using *a priori* knowledge about the object and soft computing can significantly improve modelling quality [6-7]. Therefore hybrid

analytical-neural dynamic model of temperature sensor is proposed in this paper. Hybrid modelling approach is compared with dynamical temperature sensors' models based only on *Artificial Neural Networks* (ANN).

2 Linear Dynamic Models of RTD

In traditional modelling approach a sensor's model structure is evaluated under the laws of thermokinetics, which are complex nonlinear partial differential equations [2].

For small temperature ranges and constant sensor working conditions a simplified linear and lumped parameter model of a temperature sensor given by the operator transfer function (1) is used in practice [1].

$$G_T(s) = \frac{\Theta_T(s)}{\Theta_O(s)} = \frac{1}{\prod_{i=1}^n (1+sN_i)} \quad (1)$$

where: $\Theta_O(s)$ – Laplace transform of the changes in the temperature of the medium surrounding the sensor, $\Theta_T(s)$ – Laplace transform of the changes in the temperature of the sensing element evoked by the change of temperature Θ_O , N_i – thermal time constants.

Parameters of model (1) – thermal time constants are identified in an experimental way on the basis of the sensor's input and output signals. The order of the model n depends on the sensor's size and construction, heat exchange conditions, and also on the accuracy required in a given application. For low inertial RTD with a thin sheath the model of the second order usually yields a high approximation accuracy [2].

On the basis of *a priori* knowledge and experimentally identified thermal time constants the final sensor's model, given by equation (2) can be proposed.

$$\Theta_O(t) = \Theta_T(t) + (N_1 + N_2) \frac{d\Theta_T(t)}{dt} + N_1 N_2 \frac{d^2\Theta_T(t)}{dt^2}. \quad (2)$$

Equation (2) was used by the author to develop an algorithm for modelling the sensor's dynamics, which was applied in a computerised measuring system. A discrete time sensor model corresponding to differential equation (2) is given by the following difference equation:

$$\Theta_O(k) = \Theta_T(k) + \frac{N_1 + N_2}{2\Delta t} [\Theta_T(k+1) - \Theta_T(k-1)] + \frac{N_1 N_2}{\Delta t^2} [\Theta_T(k+1) - 2\Theta_T(k) + \Theta_T(k-1)], \quad (3)$$

where Δt is the sampling interval.

After transformation of equation (3) we obtain:

$$\Theta_T(k+1) = -a_1 \Theta_T(k) - a_2 \Theta_T(k-1) + b_1 \Theta_O(k), \quad (4)$$

where: $a_1 = b_1 (1 - 2N_1 N_2 / \Delta t^2)$, $a_2 = b_1 [2N_1 N_2 - (N_1 + N_2)\Delta t] / 2\Delta t^2$ are the autoregressive and $b_1 = 2\Delta t^2 / [2N_1 N_2 + (N_1 + N_2)\Delta t]$ exogenous coefficients.

Difference equation (4) is the sensor's ARX model (*AutoRegressive with eXogenous variables*), which allows to predict the next sample of the sensor's output signal $\Theta_T(k+1)$ on the basis of the current input signal sample $\Theta_O(k)$ and current and previous output samples $\Theta_T(k)$, $\Theta_T(k-1)$.

3 Modelling of Sensors' Dynamics by the Use of ANN

A general model of a nonlinear discrete-time dynamical *Single-Input Single-Output* (SISO) system can be defined by the following difference equation [8]:

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)], \quad (5)$$

where: $u(k)$ and $y(k)$ represent the input-output pair of the SISO system at time k , f is a nonlinear function and the integer numbers m, n satisfy the condition $m \leq n$.

This nonlinear model is based on relations between the sensor's input vector U to its output vector Y . The predicted value of the system output signal is calculated from the current system input and output signals and a number of previous consecutive signal samples. This model is a generalization of the linear sensor's model given in (4). ANN-based dynamic models of the systems, equivalent to equation (5), were proposed by Narendra and Parthasarathy in 1990. They proposed two main classes of ANNs for this purpose [8]:

- *multilayer FeedForward Networks* (FFN) – with so called *moving window* method, in which the network is connected to the current and a number of past samples of system states,
- *Recurrent Neural Networks* (RNN).

The ANN-based identification scheme proposed for identification and modelling of industrial temperature sensors featuring nonlinear dynamic properties is shown in Fig. 1. A serial-parallel method and FFN of *MultiLayer Perceptron* (MLP) type were used for the sensors identification and the backpropagation training algorithm with the Levenberg-Marquardt optimization method were applied to minimize the *Mean-Square Error* (MSE) cost function. A *Tapped Delay Line* (TDL) is defined as a series

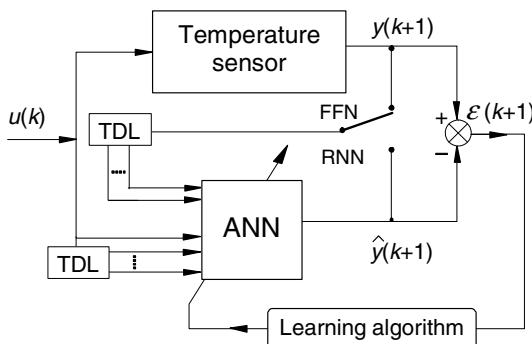


Fig. 1. A schematic diagram of ANN-based sensor's identification

of time delay units [8]. The FFN and RNN obtained from FFN by closing the feedback connections were used for the sensors' modelling. An iterative algorithm with a criterion of minimum of MSE between the sensor's output and the network output was used for RNN training. This method of temperature sensors dynamics modeling was reported in [5].

For the considered application, the key requirement is proper preprocessing of the data for the networks' training and testing. A sequence of increasing and decreasing step functions with different magnitudes was proposed by the author (in Fig. 4) as the interrogation signal.

4 Hybrid Modelling Approach

ANN-based sensors' models are built on the basis of relations between the sensor's input and output vectors, i.e. the so called black-box modelling approach. ANNs are an important alternative to analytical methods, as they allow to built nonlinear dynamical sensor's model neglecting details of laws of thermokinetics and RTD construction. However, ANN-based models have complex structure and large number of parameters, which do not have their physical representation.

In hybrid modelling approach that is proposed the so called grey-box modelling is applied. In this approach a sensor's model structure is determined from *a priori* knowledge and only unknown model parameters are obtained experimentally using soft computing methods. The structure of the proposed model is shown in Fig. 2.

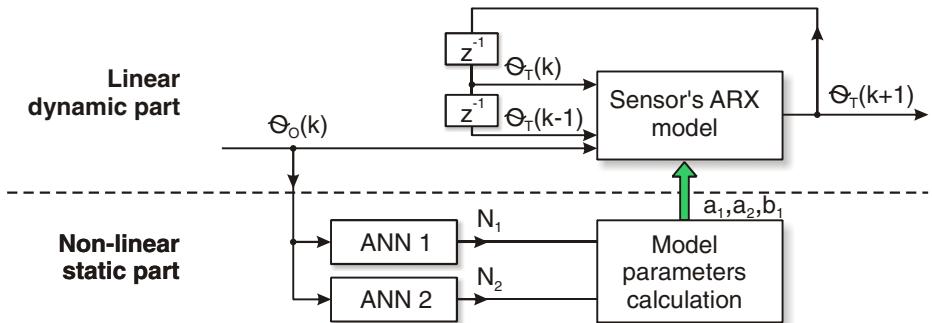


Fig. 2. Schematic diagram of the simplified nonlinear hybrid analytical-neural model of a temperature sensor

This model consist of linear dynamic part and non-linear static part. It can be viewed as many interconnected ARX models (Eq. 4). Each of these partial linear dynamical models is dedicated to a small range of the medium temperature. With each temperature change, a new set of model parameters (a_1, a_2, b_1) is calculated. For stationary systems, the nonlinear static part of the model can be determined off-line. In the proposed hybrid modeling approach the relations between thermal time constants and medium temperature Θ_o are approximated by means of ANN on the basis of experimental results.

5 Experimentation

Verification of the proposed modelling methods was performed experimentally for two small Pt 100 industrial resistance temperature detectors in steel sheaths:

TOPC 4 – diameter $d = 4$ mm, length $l = 115$ mm, class B, range $(-50 \div 200)^\circ\text{C}$,
 TOPI 6 – the parameters: $d = 6$ mm, $l = 175$ mm, class B, range $(-200 \div 600)^\circ\text{C}$.

The sensors were tested in a thermostat in well stirred oil. In these working conditions they can be considered as stationary nonlinear objects. A series of experiments of sensor identification was performed using the step response method for various magnitudes of temperature step interrogations. For each single interrogation step, the oil temperature was constant, so the sensor's dynamics were approximated by the simplified linear model given in (1) and $n = 2$. Nonlinear dependence of the experimentally determined sensors' time constants N_1, N_2 on oil temperature (shown in Fig. 3) was approximated by MLP neural networks.

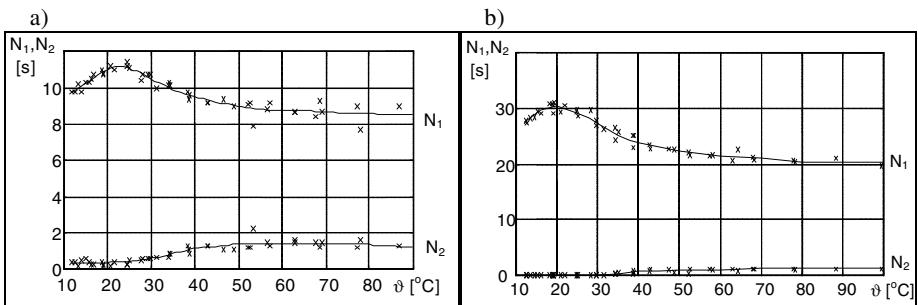


Fig. 3. Sensor's equivalent time constants N_1, N_2 versus oil temperature approximated by MLP networks: a) for the TOPC 4 sensor: MLP (1-6-1) for N_1 and MLP (1-3-1) for N_2 ; b) for the TOPI 6 sensor: MLP (1-4-1) for N_1 and MLP (1-2-1) for N_2 .

The nonlinear sensor model shown in Fig. 2 was employed to compute a series of the sensors' step responses. Static nonlinearity in the model was approximated by the trained ANN and linear dynamics by the ARX model given by equations (4). The obtained sensors' step responses were used in the ANN based approach as the

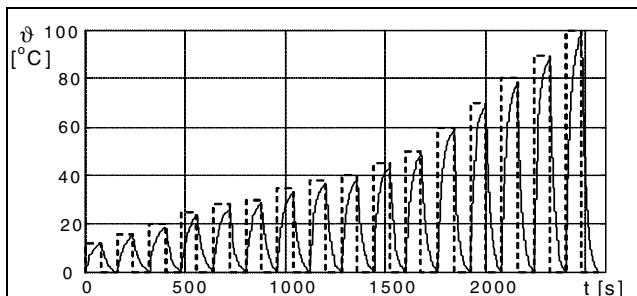


Fig. 4. Experimental data for networks' training and testing for the TOPI 6 sensor: temperature step interrogation – dashed line, sensor's output – solid line

networks training and testing data sets. An example of the training data for the TOPI 6 sensor is shown in Fig. 4. For both the sensors, two from the step responses were used for network testing and the rest were used for the networks' training.

Different two-layer FFN and RNN were trained as the sensors' models. The network structures are described schematically as (v - x -1) i.e., v inputs, x neurons with a sigmoidal activation function in the hidden layer and one output. The main criterion for choosing the best network structure was the minimum of the root mean square error E_{RMSE} computed for the testing data, but also E_{RMSE} for the training data, maximum absolute error E_{max} , mean absolute error E_{MAE} , mean error E_{mean} and the network structure were taken under consideration. Errors obtained for a chosen example networks are shown in table 1.

Table 1. Sensors' dynamical models – structures of chosen ANN and the corresponding modelling absolute errors: E_{RMSE} – root mean square error, E_{max} – maximum absolute error, E_{MAE} – mean absolute error, E_{mean} – mean error, N_{ep} – number of ANN training epochs.

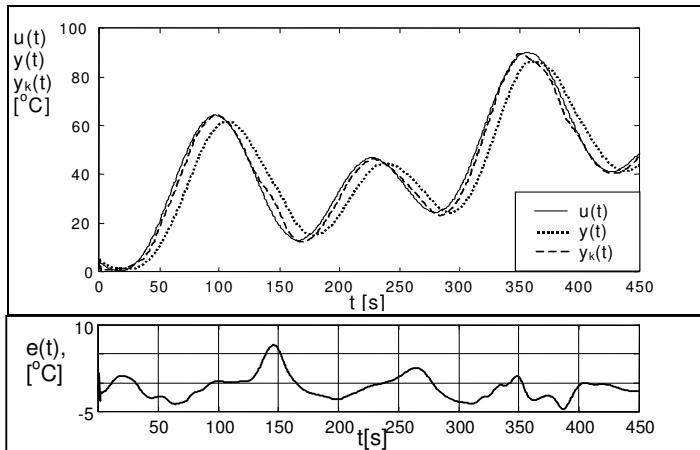
No	sensor	ANN structure	N_{ep}	training data		testing data		
				(–)	E_{RMSE} (°C)	E_{RMSE} (°C)	E_{max} (°C)	E_{MAE} (°C)
1	TOPC 4	MLP (3-6-1)	7	0,11	0,09	0,47	0,07	– 0,03
2	TOPC 4	MLP (5-20-1)	5	0,08	0,07	0,49	0,03	– 0,004
3	TOPI 6	MLP (3-4-1)	6	0,33	0,21	0,97	0,14	0,02
4	TOPI 6	MLP (5-22-1)	7	0,13	0,11	0,99	0,04	0,006
5	TOPC 4	RNN (5-20-1)	5	0,08	0,57	1,8	0,42	– 0,03
6	TOPI 6	RNN (3-4-1)	6	0,33	1,4	7,3	1,1	0,36

Very accurate ANN models were obtained for both the tested sensors in the temperature range $0\div100$ °C by the use of feedforward MLP networks (Table 1). For the (3- x -1) structures the current network input $X = [u(k), y(k), y(k-1)]$ and for the (5- x -1) structures $X = [u(k), u(k-1), y(k), y(k-1), y(k-2)]$. The predicted temperature value $y(k+1)$ is computed at the network output, as given in equation (5).

On the basis of the trained FFN the corresponding RNN were constructed by the method shown in Fig. 1. From the large set of obtained RNN only a few of them, after closing the feedback connections, were capable to modelling sensors dynamics properly. The errors obtained for the two best RNN chosen from the all tested networks are shown in Table 1 (No 5 & 6). The modelling errors for the testing data are a few times greater for the RNN than for the corresponding FFN.

The RNN and hybrid sensors models were verified by computer simulations for poly-harmonic interrogating signals via dynamical error correction. In this study, serial correction method was applied, in which ANN-based corrector implements an inverse dynamical model of the sensor. This correction method was reported in [9]. Results of the sensors modelling and correction by the (6-16-1) MLP FFN are shown in Fig. 5 and listed in table 2.

a) Neural sensor model (RNN) – ANN-based corrector (MLP)



b) Hybrid sensor model – ANN-based corrector (MLP)

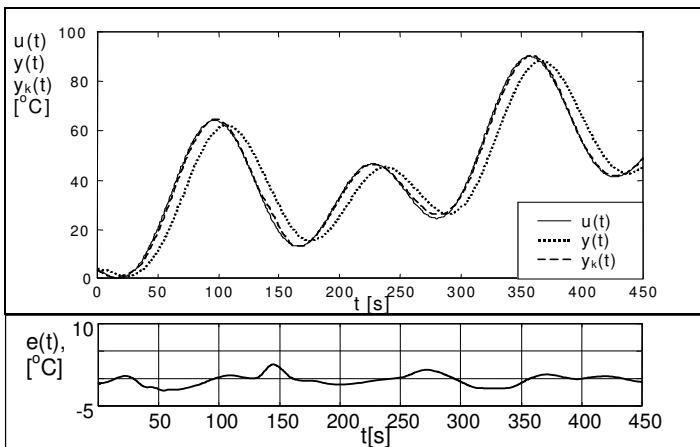


Fig. 5. Results of neural correction and modelling for TOPC 4 sensor; by means of RNN: interrogating poli-harmonic signal $u(t) = 0.01t + 2\sin(\omega t) + 2\sin(3\omega t) - 2\sin(7\omega t)$ – solid line, sensor's output computed by RNN (5-20-1) – dotted line, correction by the (6-16-1) MLP FFN – dashed line; $e(t)$ – correction error $e(t) = y_k(t) - u(t)$

Table 2. Dynamical errors correction for poly-harmonic interrogating signal for the RNN and hybrid sensors models and the MLP FFN correctors; E_{RMSE} – root mean square error, E_{max} – maximum absolute error, E_{MAE} – mean absolute error

No	sensor	sensor's model	corrector's structure	E_{RMSE} (°C)	E_{MAE} (°C)	E_{max} (°C)
1	TOPC 4	RNN (5-20-1)	MLP (6-16-1)	2.21	1.74	6.57
2	TOPC 4	hybrid	MLP (6-16-1)	0.96	0.78	2.74
3	TOPI 6	RNN (3-4-1)	MLP (6-18-1)	1.93	1.52	4.97
4	TOPI 6	hybrid	MLP (6-18-1)	0.61	0.44	2.08

6 Conclusions

Temperature sensors immersed in oil exhibit strongly nonlinear dynamic properties in a wide temperature range. Therefore the linear modelling and correction methods are not applicable in such process conditions.

The performed experiments proved that MLP feedforward and recurrent neural networks are capable of modelling the temperature sensor's nonlinear dynamics. RNN-based models are less precise than FFN, however, they can be applied as an independent nonlinear sensors' simulators.

In the reported application it was shown that hybrid analytical-neural modelling approach using *a priori* knowledge about the object has significantly improved modelling accuracy.

References

1. Michalski, L., Eckersdorf, K., McGhee, J.: Temperature Measurement. John Wiley and Sons Ltd., Chichester (1991)
2. Jackowska-Strumillo, L., Sankowski, D., McGhee, J., Henderson, I.A.: Modelling and MBS experimentation for temperature sensors. *Measurement* 20(1), 49–60 (1997)
3. Cimerman, F., Blagojevic, B., Bajsic, I.: Identification of the dynamic properties of temperature sensors in natural and petroleum gas. *Sensors & Actuators A* 96, 1–13 (2002)
4. Minkina, W.: Theoretical and experimental identification of the temperature sensor unit step response non-linearity during air measurement. *Sensors & Actuators A* 78, 81–87 (1999)
5. Jackowska-Strumiłło, L.: ANN based modelling and correction in dynamic temperature measurements. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 1124–1129. Springer, Heidelberg (2004)
6. Xie, W.F., Zhu, Y.Q., Zhao, Z.Y., Wong, Y.K.: Nonlinear system identification using optimized dynamic neural network. *Neurocomputing* 72, 3277–3287 (2009)
7. Mantovanelli, I.C.C., Rivera, E.C., Da Costa, A.C., Filho, R.M.: Hybrid Neural Network Model of an Industrial Ethanol Fermentation Process Considering the Effect of Temperature. *Applied Biochemistry and Biotechnology* 817, 136–140 (2007)
8. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks* 1(1), 4–27 (1990)
9. Jackowska-Strumiłło, L.: Correction of non-linear dynamic properties of temperature sensors by the use of ANN. In: Rutkowski, L., Kacprzyk, J. (eds.) Advances in Soft Computing – Neural Networks and Soft Computing, pp. 837–842. Physica-Verlag, Heidelberg (2003)

An Improved Annealing Algorithm for Throughput Maximization in Static Overlay-Based Multicast Systems

Michał Kucharzak and Krzysztof Walkowiak

Department of Systems and Computer Networks

Wrocław University of Technology, Poland

{michal.kucharzak,krzysztof.walkowiak}@pwr.wroc.pl

Abstract. Overlay-based multicast has been proposed as an efficient solution to ensure multicast communication in the Internet. In contrast to IP multicast, it uses a virtual topology created in an overlay network. For a few past years overlay-based multicast has attracted research community and many new protocols have been proposed. Thereby, a lot of research has been focusing on various aspects of such systems, however most of previous works concern on minimization of flow costs. In this paper, we take a look at the problem of modelling and optimization of overlay-based multicast with a special focus on maximization of system's throughput. We employ an integer program of fractional tree packing problem and propose an improved annealing algorithm for solving the problem. Efficiency of the algorithm is compared to optimal results as well as random ones and a basic simulated annealing.

Keywords: overlay multicast, simulated annealing, tree packing.

1 Introduction

Over the years, a lot of research, development and testing efforts have been devoted to support multicast communication. Initially, network-aware protocols implemented in the IP layer has been developed [6]. However, IP multicast is afflicted with many technical problems derived from scalability, addressing scheme management, flow or congestion control and even its 20-years development it is far from being globally and widely deployed. To tackle some of the problems, overlay-based multicast has been proposed. It comprises systems which realize multicast routing by forming virtual topologies (mainly trees) on the top of network and transportation layers. This virtual-oriented concept expands end-system multicast [4][2] and provides potential scalability and ease of deployment of new, network-layer-independent protocols at relatively low costs. Contrariwise to IP multicast, in overlay system multicast data packets are replicated and forwarded by participating nodes instead of IP routers and logical multicast flows are realized as multiple unicast flows at the level of the network layer.

Most of previous works on overlay multicast focus mainly on flow costs or delay minimization [14][11][13]. On the other hand, most of past papers on

throughput optimization take into account network-aware systems with IP routers and switches [29][16]. Therefore, in this paper, we introduce throughput maximization problem in an overlay-based multicast systems. We employ a relaxed version of the NP-hard Steiner tree packing problem [9][3] and formulate the tree packing problem in static overlay-based multicast systems with limited number of trees. By static, we assume systems with constant set of nodes, without joining and leaving users. To solve the optimization problem we propose an annealing-based metaheuristic. However many intelligent hybrid algorithms are derived from evolutionary techniques [5][7], we employ simulated annealing and apply additional improvements including repetition of an initial solution selection and a suspension time in current temperature state. Results confirm that applying such kind of intelligence to simulated annealing provides additional gain.

The main contribution of the paper is a formulation of the tree packing problem dedicated to maximization of system's throughput in overlay-based multicast networks and as well as a new improved annealing algorithm.

2 Problem Description

We consider an overlay system and the problem of information multicasting, when every user participating in the system obtains the same amount of data from a single source. Based on natural relaxation of the Steiner tree packing problem proposed in [9] and known as fractional Steiner tree packing problem, we propose spanning tree packing problem applying features of overlay networks in order to maximize throughput in such systems.

2.1 Spanning Tree Packing in Static Overlay-Based Multicast

The basic formulation includes a predefined set of trees $t \in \{1, \dots, T\}$ and a corresponding set of integer variables x_t , which represent flow allocated to tree t . Trees are represented by vector β which contains parent-relation for all nodes. For example, $\beta_t = [1, 1, 1, 3, 4, 3]$ refers to a tree in which a node 1 is a parent of nodes 2 and 3, node 3 is a parent of 4 and 6, and parent of 5 is 4. In addition, the root node is coded as its own parent. The problem comprises finding a maximal throughput assignment to at most Q trees selected from predefined set of trees regarding available capacities of participants' access link. Every tree t represents a fractional multicast structure.

Problem Formulation

indices

$w, v = 1, 2, \dots, V$ vertices (overlay nodes)

$t = 1, 2, \dots, T$ predefined trees

constants

β_{tv} parent node of v in tree t

- d_v available download capacity of peer v
- u_w available upload capacity of peer w
- r root node
- Q limit of trees that can be selected
- M_t maximum throughput of tree t

variables

- x_t throughput assigned to tree t (e.g. streaming rate in kbps, integer)
- $y_t = 1$, if tree t can carry any traffic; 0, otherwise (binary, auxiliary variable)

objective

$$\text{maximize} \quad F = \sum_t x_t . \quad (1)$$

constraints

$$\sum_t \sum_{v:\beta_{tv}=w, v \neq w} x_t \leq u_w \quad w = 1, \dots, V . \quad (2)$$

$$\sum_t x_t \leq \min\{\min_{v \neq r}\{d_v\}, u_r\} . \quad (3)$$

$$\sum_t y_t \leq Q . \quad (4)$$

$$x_t \leq M_t y_t \quad t = 1, \dots, T . \quad (5)$$

The main goal of the problem (1) is to maximize overall system's throughput by assigning flows to trees $t \in \{1, \dots, T\}$. Constraints (2) refer to upload capacity limits and allocated flow on all trees $\sum_t x_t$ containing arcs from w to any other node ($v : \beta_{tv} = w, v \neq w$) cannot exceed given upload limit u_w of peer w , i.e. total outgoing flow from w is limited to its upload capacity u_w . We assume that every node is connected to the system and incoming flow to every node (except the root) is exactly of the same value. Thereby, the formula (3) provides that the overall throughput $\sum_t x_t$ cannot exceed download capacity d_v of any participant v or upload capacity of the source (u_r). In accordance to practical application, due to coding, control or management mechanisms but also due to CPUs power we introduce an additional constraint (4) which limits the number of used trees to given Q . Finally, throughput x_t equals to 0 if tree t is not used ($y_t = 0$), otherwise it cannot be greater than the maximum possible throughput M_t .

Let v^{*t} refer to the number of v 's children in tree t .

$$M_t = \min\{\min_{v \neq r} d_v, \min_{v:v^{*t} \neq 0} \frac{u_v}{v^{*t}}\} \quad t = 1, \dots, T . \quad (6)$$

In contrary to problems defined for IP-based systems, where the maximum throughput of a tree is limited to the minimal link capacity, in overlay systems the upper bound of tree's throughput depends on either the minimum download limit of any node or upload limit of v (u_v) and number of its children to be served v^{*t} (equation (6)). That means, the more children are assigned to v , the smaller fraction of u_v might be dedicated to them. Moreover, regarding overlay

networks, limited capacities of access links might be considered as the only bottlenecks in the system and the underlay core network is usually considered as over-provisioned [115].

3 Improved Annealing Algorithm

An Improved Annealing Algorithm (IAA) derives its idea from simulated annealing (SA) introduced by Kirkpatrick in [10]. A basic version of SA was inspired by analogy to the physical annealing process in which heated solid is slowly cooled so that in the end its structure has an optimal energy configuration. Regarding the model of the problem, a *solid structure* refers to x_t (a solution) and an *energy configuration* is its utility value $\sum_t x_t$. The IAA employs several improvements over basic SA and its pseudocode is shown in Fig. 1.

```

1  set TEMPERATURE, COOLING_FACTOR, TEMP_MIN
2  set INITIAL_SEARCH, SUSPENSION_TIME,
3  BEST_SOLUTION := getInitialSolution(INITIAL_SEARCH)
4  CURRENT_SOLUTION := SOLUTION
5  while TEMPERATURE > TEMP_MIN do
6      for 1:SUSPENSION_TIME
7          NEIGHBOUR_SOLUTION := getNeighbourSolution(CURRENT_SOLUTION)
8          ΔE := getCost(NEIGHBOUR_SOLUTION) - getCost(CURRENT_SOLUTION)
9          if exp(ΔE/TEMPERATURE) > random(0,1) then
10              CURRENT_SOLUTION := NEIGHBOUR_SOLUTION
11              if getCost(BEST_SOLUTION) < getCost(CURRENT_SOLUTION) then
12                  BEST_SOLUTION := CURRENT_SOLUTION
13              end if
14          end if
15      end for
16      coolingProcess(TEMPERATURE,COOLING_FACTOR)
17  end while

```

Fig. 1. The Improved Annealing Algorithm pseudocode

Parameters in line 1 are essential in annealing-based methods. TEMPERATURE is gradually decreased during the process being multiplied by COOLING_FACTOR $\in (0, 1)$ in each iteration. The algorithm stops working after reaching TEMP_MIN threshold. Additionally, the IAA comprises INITIAL_SEARCH used for tuning a starting structure (solution), SUSPENSION_TIME which defines iterations of staying in instantaneous temperature before proceeding cooling. In line 3 an initial BEST_SOLUTION is generated randomly. A function getInitialSolution chooses the best solution from the subset of size INITIAL_SEARCH of the search space. Initially chosen BEST_SOLUTION becomes instantaneous CURRENT_SOLUTION simultaneously.

The main loop of the algorithm (lines 5-17) emulates annealing which processes until temperature condition is satisfied. Regarding CURRENT_SOLUTION that represents an instantaneous structure of x_t , a function getNeighbourSolution generates randomly a new nearby solution of it (line 7). Depending on the cost difference (ΔE) between a neighbour and the current solutions, CURRENT_SOLUTION is replaced by NEIGHBOUR_SOLUTION with probability given by $\exp(\Delta E / \text{TEMPERATURE})$. In case when the neighbour is better than the current solution, the probability is

not less than 1, otherwise if ΔE is sufficiently small (the neighbour is not much worse than the current solution) and TEMPERATURE is sufficiently high, the neighbour solution might be considered as current solution. This concept provides escaping from a potential trap of sticking in a local optimum.

Essentially, two main improvements over basic SA are implemented: the first one concerns selection of an initial solution, and the second introduces time suspension at instantaneous temperature level.

One of the crucial procedure of annealing-based algorithms is a scheme of a neighbour selection shown in Fig. 2.

```

1   NEW_SOLUTION := CURRENT_SOLUTION
2   removeRandomTrees(NEW_SOLUTION)
3   decreaseAssignedThroughputRandomly(NEW_SOLUTION)
4   while isTreeAssigningPossible() do
5       TREE := selectRandomTree()
6       assignMaximumThroughput(TREE)
7       addTreeToSolution(TREE)
8   end while
9   while isThroughputAssigningPossible() do
10      assignMaximumThroughputToRandomTree(NEW_SOLUTION)
11   end while
12   return NEW_SOLUTION

```

Fig. 2. Pseudocode of the `getNeighbourSolution` procedure

The procedure removes some trees and randomly decreases remaining throughput assigned previously to `CURRENT_SOLUTION` (lines 2–3). Next, a new tree is selected randomly and maximum throughput possible is added. The loop 4–8 processes until any additional tree cannot be allocated with nonzero throughput or if limit of used trees has been reached. Finally, in loop 9–11, throughput of trees in `NEW_SOLUTION` is completed until no more can be added. The `getNeighbourSolution` is implemented by SA and IAA heuristics in the same form.

4 Numerical Experiments

Benchmark values. To compare the quality of IAA, optimal results were obtained with CPLEX [8] and its C++ interface. All computations were performed on a PC, Intel Core2 Duo, 2.13 GHz, 4GB RAM, Windows 7 Professional. A random algorithm (RA) and a basic simulated annealing were implemented in order to observe a gain of improvements of the IAA. Considering RA, random throughput is assigned to randomly selected tree until any additional throughput cannot be allocated. All algorithms were implemented in C++ in MS VS2008.

Instances. Generated instances contain upload limits proportionally distributed from values {512, 1024, 1536 kbps}, where limit of the root node u_r equals to 1536 kbps and download capacities $d_v = 4096$ kbps are the same for all nodes in the system.

Results. First experiments comprise a scenario with $V = 100$, $T = 500$. Higher values of parameters: TEMPERATURE, COOLING_FACTOR, INITIAL_SEARCH and SUSPENSION_TIME provide larger subspace of examined solutions but also require increased computational time. Therefore, it is challenging to determine the parameters with regard to efficient searching in reasonable time or search space. Table 1 presents parameters' values that provide the most satisfactory results among all tested values which ensure testing the search space of size of 10000.

Table 1. Tuning parameters

Parameter	Best value	Tested values
TEMPERATURE	50	10, 50, 100, 200, 500, 1000, 2000, 10000, 20000, 50000
COOLING_FACTOR	0.995	0.9, 0.95, 0.975, 0.98, 0.985, 0.99, 0.992, 0.993, 0.995, 0.999, 0.9993, 0.9995
INITIAL_SEARCH	500	2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000
SUSPENSION_TIME	12	1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20

Table 2 compares algorithms' performance for the given scenario. All heuristics search a solution subspace of similar sizes. RA checks 10000 randomly selected feasible solutions; SA uses TEMPERATURE=1000 and COOLING_FACTOR=0.999931; IAA uses values of parameters indicated in Tab. 1. Columns of the table represent: consecutive limits Q , the best results found in 1000 independent repetitions of each heuristic, average results, average computational time.

Table 2. Results comparison. Scenario $V=100$, $T=500$, $Q=2,\dots,6$

Q	Results			Average (standard deviation)			Time (s)				
	OPT	RA	SA	IAA	RA	SA	IAA	OPT	RA	SA	IAA
2	358	358	358	358	341 (9.0)	348 (8.1)	352 (8.0)	8.0	1.0	6.7	7.1
3	453	426	453	453	369 (15.4)	404 (20.9)	421 (15.4)	70.8	1.3	8.4	7.9
4	510	438	480	503	389 (15.5)	436 (18.8)	461 (15.4)	489.6	1.4	11.9	11.2
5	535	437	495	512	404 (12.4)	451 (23.9)	480 (14.6)	2113.0	1.7	12.5	13.0
6	—	446	520	532	414 (12.6)	466 (22.6)	512 (14.6)	—	1.8	14.4	15.7

For the case where $Q = 2$, all heuristics can reach an optimal threshold, however the best average result is provided with the IAA and the gap to optimum equals to 1.7% in relation to 2.8% and 4.7% of SA and RA, respectively. Increasing Q has an impact on larger solution space and affects bigger difference in result effectiveness. For $Q = 3$, annealing algorithms found an optimal solution, in contrast to RA which yielded a 6% gap. In all cases, the results provided by the IAA are better than RA's and SA's results and relative gaps to optimal for $Q = 5$ equal to 4.3%, 7.5%, and 18.3% for IAA, SA and RA respectively. For $Q = 6$, CPLEX solver encountered an out-of-memory error, thus solving large size problems is often ineffective and computational time required for performing optimization becomes impractical and often unacceptable.

Next experiments focus on instances including $V = 40$, $T = 400$ and limited number of children per node. A subset of a searched space is decreased to about 1000: RA performs 1000 iterations; SA's temperature = 1100 and cooling factor = 0.993; IAA's parameters are 10, 0.99, 70 and 4 for temperature, cooling factor, initial search and suspension time, respectively. Fig. 3 presents quality of results in relation to optimal.

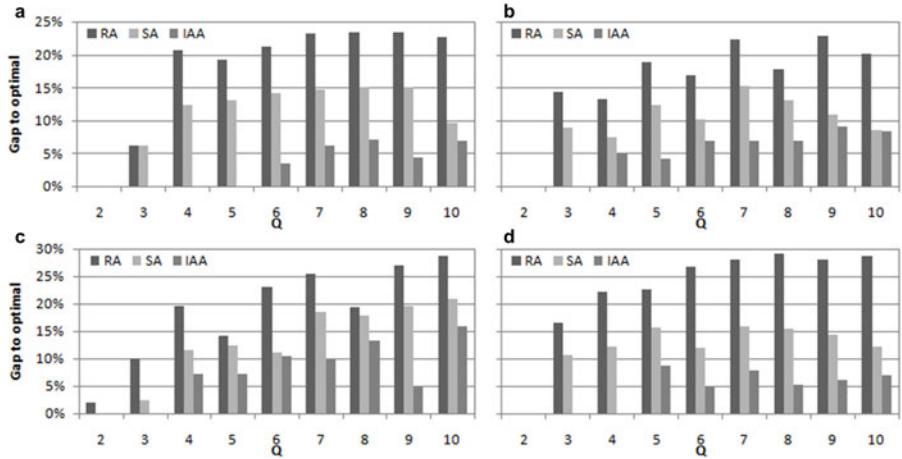


Fig. 3. Optimization results. Trees generated with limited number of children per node: (a) 2; (b) 4; (c) 10; (d) 20.

The IAA outperforms other algorithms and often reach optimal results, especially for small Q . In the worst cases the gap does not exceed 16% and, on average, equals to 5%, while SA yields 11% and RA approximately approaches to 19%.

5 Concluding Remarks

This paper proposed the fractional spanning tree packing problem for throughput maximization in static overlay-based multicast systems. Moreover, an improved annealing algorithm was proposed and evaluated in relation to optimal results as well as to results obtained with a basic simulated annealing and a random algorithm. Results indicate that, improving an initial solution and providing additional repetitions at certain temperature levels in IAA positively influence on a final result quality.

Acknowledgements

This work is partially supported by The Polish Ministry of Science and Higher Education.

References

1. Akbari, B., Rabiee, H.R., Ghanbari, M.: An Optimal Discrete Rate Allocation for Overlay Video Multicasting. *Computer Communications* 31(3), 551–562 (2008)
2. Chen, S., Günlük, O., Yener, B.: The Multicast Packing Problem. *IEEE/ACM Trans. Netw.* 8(3), 311–318 (2000)
3. Cheriyan, J., Salavatipour, M.R.: Hardness and Approximation Results for Packing Steiner Trees Problems. In: Albers, S., Radzik, T. (eds.) *ESA 2004*. LNCS, vol. 3221, pp. 180–191. Springer, Heidelberg (2004)
4. Chu, Y.-h., Rao, S.G., Zhang, H.: A Case for End System Multicast. *Proceedings of ACM Sigmetrics*, pp. 1–12 (2000)
5. Corchado, E., Abraham, A., de Carvalho, A.: Editorial: Hybrid intelligent algorithms and applications. *Inf. Sci.* 180, 2633–2634 (2010)
6. Deering, S.E., Cheriton, D.R.: Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Trans. Comput. Syst.* 8, 85–110 (1990)
7. Derrac, J., García, S., Herrera, F.: A First Study on the Use of Coevolutionary Algorithms for Instance and Feature Selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) *HAIS 2009*. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
8. IBM ILOG CPLEX 12.1: User's Manual for CPLEX (2009)
9. Jain, K., Mahdian, M., Salavatipour, M.R.: Packing Steiner trees. In: *SODA 2003: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 266–274. Society for Industrial and Applied Mathematics, Philadelphia (2003)
10. Kirkpatrick, S.: Optimization by Simulated Annealing. *Numerische Mathematik* 220(4598), 671–680 (1983)
11. Kucharzak, M., Walkowiak, K.: Optimization of Flows in Level-Constrained Multiple Trees for P2P Multicast System. In: *The Second International Conference on Advances in P2P Systems, AP2PS 2010*, Florence, Italy (October 2010)
12. Sentinelli, A., Marfia, G., Gerla, M., Kleinrock, L., Tewari, S.: Will IPTV Ride the Peer-to-Peer Stream? Peer-to-Peer Multimedia Streaming. *IEEE Communications Magazine* 45(6), 86–92 (2007)
13. Walkowiak, K.: Network Design Problem for P2P Multicasting. In: *International Network Optimization Conference INOC 2009* (2009)
14. Wu, C., Li, B.: Optimal Peer Selection for Minimum-Delay Peer-to-Peer Streaming with Rateless Codes. In: *P2PMMS 2005: Proceedings of the ACM Workshop on Advances in Peer-to-Peer Multimedia Streaming*, pp. 69–78. ACM, New York (2005)
15. Wu, G., Chiueh, T.: Peer to Peer File Download and Streaming. RPE report, TR-185 (2005)
16. Wu, Y., Jain, K., Kung, S.-Y.: A Unification of Network Coding and Tree-Packing (Routing) Theorems. *IEEE/ACM Trans. Netw.* 14(SI), 2398–2409 (2006)

An Implementation of Differential Evolution for Independent Tasks Scheduling on GPU

Pavel Krömer, Jan Platoš, Václav Snášel, and Ajith Abraham

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic
{pavel.kromer,jan.platos,vaclav.snasel}@vsb.cz

Abstract. Differential evolution is an efficient meta-heuristic optimization method with solid record of real world applications. In this paper, we present a simple and efficient implementation of the differential evolution using the massively parallel CUDA architecture. We demonstrate the speedup and improvements obtained by the parallelization of this intelligent algorithm on the problem of scheduling of independent tasks in heterogeneous environments.

1 Introduction

Nowadays, many algorithms are redesigned and rewritten for the GPU since modern GPUs introduce massive parallelism for a budget price and new APIs simplify the development of parallel applications. Among others, many successful intelligent algorithms including genetic algorithms and genetic programming were implemented on GPU. In this paper we introduce a simple parallelization of the differential evolution using the nVidia CUDA technology and demonstrate its performance and quality on the problem of scheduling of independent tasks in heterogeneous environments.

In grid and distributed computing, mixed-machine heterogeneous computing (HC) environments utilize a distributed suite of different machines to perform different computationally intensive applications that have diverse requirements [1][2]. Task scheduling, i.e. mapping of a set of tasks to a set of resources, is required to exploit the different capabilities of a set of heterogeneous resources. It is known, that an optimal mapping of computational tasks to available machines in an HC suite is a NP-complete problem [3] and as such, it is a subject to various heuristic [2][4][5] and meta-heuristic [6][7][8][9] algorithms. This work fits into the framework of hybrid intelligent systems [10][11]. The implementation of the intelligent algorithm – the differential evolution – is effectively split between the CPU and GPU in order to achieve more efficient and robust tool.

1.1 GPU Computing

Modern graphics hardware plays an important role in the area of parallel computing. Graphics cards have been used to accelerate gaming and 3D graphics

applications, but recently, they have been used to accelerate computations for various topics, e.g. remote sensing, environmental monitoring, business forecasting, medical applications or physical simulations etc. Architecture of GPUs (Graphics Processing Unit) is suitable for vector and matrix algebra operations, which leads to a wide use of GPUs in the area of information retrieval, data mining, image processing, data compression, etc. Nowadays, the programmer does not need to be an expert in graphics hardware because of existence of various APIs (Application Programming Interface), which help programmers to implement their software faster. Nevertheless, it will be always necessary to follow basic rules of GPU programming to write a more efficient code.

Four main APIs exists today. The first two are vendor specific, i.e. they were developed by two main GPU producers - AMD/ATI and nVidia. The API developed by AMD/ATI is called ATI Stream and the API developed by nVidia is called nVidia CUDA (Compute Unified Device Architecture). Both APIs are able to provide similar results. The remaining two APIs are universal. The first one was designed by Khronos Group and it is called OpenCL (Open Computing Language) and the second was designed by Microsoft as a part of DirectX and it is called Direct Compute. All APIs provide a general purpose parallel computing architectures that leverage the parallel computation engine in graphics processing units.

The main advantage of GPU is its structure. Standard CPUs (central processing units) contain usually 1-4 complex computational cores, registers and large cache memory. GPUs contain up to several hundreds of simplified execution cores grouped into so-called multiprocessors. Every SIMD (Single Instruction Multiple Data) multiprocessor drives eight arithmetic logic units (ALU) which process the data, thus each ALU of a multiprocessor executes the same operations on different data, lying in the registers. In contrast to standard CPUs which can reschedule operations (out-of-order execution), the selected GPU is an in-order architecture. This drawback is overcome by using multiple threads as described by Hager et al. [12]. Current general-purpose CPUs with clock rates of 3 GHz outperform a single ALU of the multiprocessors with its rather slow 1.3 GHz. The huge number of parallel processors on a single chip compensates this drawback.

2 Differential Evolution

Differential evolution (DE) is a versatile and easy to use stochastic evolutionary optimization algorithm [13]. DE is a population-based optimizer that evolves real encoded vectors representing the solutions to given problem. The DE starts with an initial population of N real-valued vectors. The vectors are initialized with real values either randomly or so, that they are evenly spread over the problem domain. The latter initialization leads to better results of the optimization process [13].

During the optimization, DE generates new vectors that are perturbations of existing population vectors. The algorithm perturbs vectors with the scaled

difference of two (or more) randomly selected population vectors and adds the scaled random vector difference to a third randomly selected population vector to produce so called trial vector. The trial vector competes with a member of the current population with the same index. If the trial vector represents a better solution than the population vector, it takes its place in the population [13].

Differential evolution is parametrized by two parameters [13]. Scale factor $F \in (0, 1+)$ controls the rate at which the population evolves and the crossover probability $C \in [0, 1]$ determines the ratio of bits that are transferred to the trial vector from its opponent. The number of vectors in the population is also an important parameter of the population. The outline of the traditional DE is illustrated in [1].

Algorithm 1. A summary of basic differential evolution

```

1 Initialize population  $P$  consisting of  $M$  vectors;
2 Evaluate chromosomes in initial population using objective function;
3 while Termination criteria not satisfied do
4   for  $i \in \{1 \dots M\}$  do
5     Mutation. Create a trial vector  $v_t^i = v_r^1 + F(v_r^2 - v_r^3)$ , where  $F \in [0, 1]$  is
      a parameter and  $v_r^1, v_r^2, v_r^3$  are three random vectors from the population
       $P$ .
6     Validate the range of coordinates of  $v_t^i$ . Optionally adjust coordinates of
       $v_t^i$  so, that  $v_t^i$  is valid solution to given problem.
7     Perform uniform crossover. Select randomly one point (coordinate)  $l$  in
       $v_t^i$ . With probability  $1 - C$  let  $v_t^i[m] = v^i[m]$  for each  $m \in \{1, \dots, N\}$ 
      such that  $m \neq l$ 
8     Evaluate the trial vector. If the trial vector  $v_t^i$  represent a better
      solution than population vector  $v^i$ , replace  $v^i$  in  $P$  by  $v_t^i$ 
9   end
10 end
```

2.1 Differential Evolution for Scheduling Optimization

An HC environment is composite of computing resources (PCs, clusters, or supercomputers). Let $T = \{T_1, T_2, \dots, T_n\}$ denote the set of tasks that is in a specific time interval submitted to a resource management system (RMS). Assume the tasks are independent of each other with no intertask data dependencies and preemption is not allowed (the tasks cannot change the resource they have been assigned to). Also assume at the time of receiving these tasks by RMS, m machines $M = \{M_1, M_2, \dots, M_m\}$ are within the HC environment. For our purpose, scheduling is done on machine level and it is assumed that each machine uses First-Come, First-Served (FCFS) method for performing the received tasks. We assume that each machine in HC environment can estimate how much time is required to perform each task. In [2] Expected Time to Compute (ETC) matrix is used to estimate the required time for executing a task in a machine. An ETC matrix is a $n \times m$ matrix in which n is the number of tasks and m

is the number of machines. One row of the ETC matrix contains the estimated execution time for a given task on each machine. Similarly one column of the ETC matrix consists of the estimated execution time of a given machine for each task. Thus, for an arbitrary task T_j and an arbitrary machine M_i , $[ETC]_{j,i}$ is the estimated execution time of T_j on M_i . In the ETC model we take the usual assumption that we know the computing capacity of each resource, an estimation or prediction of the computational needs of each job, and the load of prior work of each resource.

The two objectives to optimize during the task mapping are makespan and flowtime. Optimum makespan (metatask execution time) and flowtime of a set of jobs can be defined as shown in [Equation 1](#).

$$\text{makespan} = \min_{S \in \text{Sched}} \{ \max_{j \in \text{Jobs}} F_j \}, \quad \text{flowtime} = \min_{S \in \text{Sched}} \{ \sum_{j \in \text{Jobs}} F_j \}. \quad (1)$$

where Sched is the set of all possible schedules, Jobs stands for the set of all jobs and F_j represents the time in which job j finalizes. Assume that $[C]_{j,i}$ ($j = 1, 2, \dots, n, i = 1, 2, \dots, m$) is the completion time for performing j -th task in i -th machine and W_i ($i = 1, 2, \dots, m$) is the previous workload of M_i , then $\sum (C_i + W_i)$ is the time required for M_i to complete the tasks included in it. According to the aforementioned definition, makespan and flowtime can be evaluated according to [Equation 2](#),

$$\text{makespan} = \min_{i \in \{1, \dots, m\}} \{ \sum C_i + W_i \}, \quad \text{flowtime} = \sum_{i=1}^m C_i. \quad (2)$$

Minimizing makespan aims to execute the whole metatask as fast as possible while minimizing flowtime aims to utilize the computing environment efficiently.

A schedule of n independent tasks executed on m machines can be naturally expressed as a string of n integers $S = (s_1, s_2, \dots, s_n)$ that are subject to $s_i \in 1, \dots, m$. The value at i -the position in S represents the machine on which is the i -the job scheduled in schedule S . Since the differential evolution uses for problem encoding real vectors, real coordinates must be used instead of discrete machine numbers. The real-encoded DE vector is translated to schedule representation by truncation of its elements. Assume schedule S from the set of all possible schedules Sched . For the purpose of differential evolution, we define a fitness function $\text{fit}(S) : \text{Sched} \rightarrow \mathbb{R}$ that evaluates each schedule in [Equation 3](#).

$$\text{fit}(S) = \lambda \cdot \text{makespan}(S) + (1 - \lambda) \cdot \frac{\text{flowtime}(S)}{m}. \quad (3)$$

The function $\text{fit}(S)$ is a sum of two objectives, the makespan of schedule S and flowtime of schedule S divided by number of machines m to keep both objectives in approximately the same magnitude. The influence of makespan and flowtime in $\text{fit}(S)$ is parameterized by the variable λ . The same schedule evaluation was used also in [\[9\]](#).

3 DE for the Scheduling of Independent Tasks Implemented on the GPU

The goal of the implementation of differential evolution on the CUDA architecture was achieving high parallelism while keeping the simplicity of the algorithm. The implementation consists of kernels (i.e. parallel methods) for generation of initial population (that is generation of pseudo random numbers), selection of random competitors for each vector in the opulation, DE processing including generation of trial vectors and crossover, validation of the range of coordinates of generated vectors, and the merger of parent and offspring populations. Besides these generic kernels implementing the DE process, an implementation of the fitness function evaluation was done in separate kernel. The overview of our DE implementation is shown in [Figure 1](#).

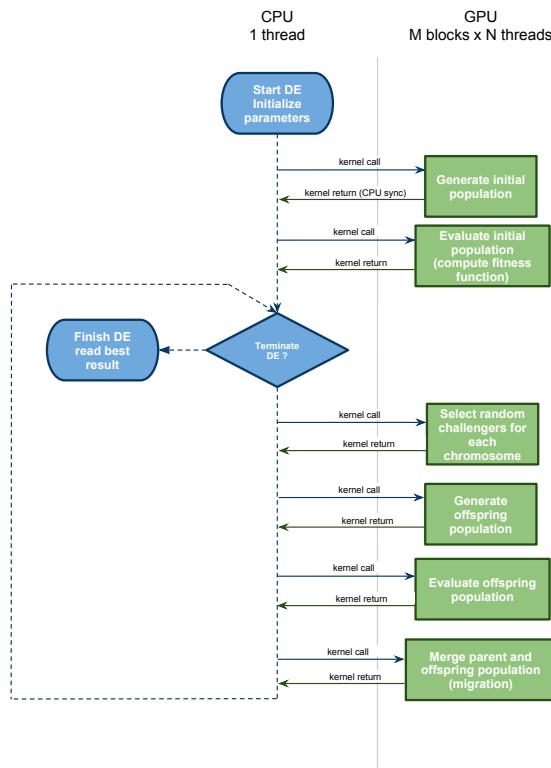


Fig. 1. The flowchart of the DE implementation on CUDA

4 Experiments

The kernels were implemented using the following principles: Each vector is processed by a thread group (set of threads sharing certain amount of memory,

also known as block). The number of thread groups is in CUDA currently limited to $(2^{16} - 1)^2$ and hence the maximum population size is in this case the same. Each vector coordinate is processed by a thread. The limit of threads per block depends on the hardware and it can be 512 or 1024. This limit enforces the maximum vector length. For the scheduling with given ETC matrices, vectors with 512 coordinates are needed. Each kernel call aims to process the whole population in one step, e.g. it asks the CUDA runtime to launch M blocks with 512 threads in parallel. The runtime executes the kernel with respect to available resources.

Such an implementation brings several advantages. First, all the generic DE operations can be considered done in parallel and thus their complexity reduces from $M \times N$ (population size multiplied by vector length) to c (constant, duration of the operation plus CUDA overhead). Second, this DE operates in a highly parallel way also on logical level. A population of offspring chromosomes of the same size as the parent population is created in a single step and later merged with the parent population. Third, the evaluation of vectors is accelerated by the implementation of the fitness function on GPU.

We have implemented the DE for scheduling of independent tasks on the CUDA architecture to evaluate the performance and quality of such a solution. We have measured the speedup obtained by the implementation and compared it to a single threaded version of the algorithm. The comparison of fitness computation time and speedup on GPU for different population sizes is illustrated in Fig. 2. We have compared a regular implementation on CPU (e.g. object oriented C++ code), optimized implementation on CPU (low level C code to achieve maximum performance) and GPU implementation that computed flowtime and makespan for the whole population of vectors. The experiment was conducted on a server

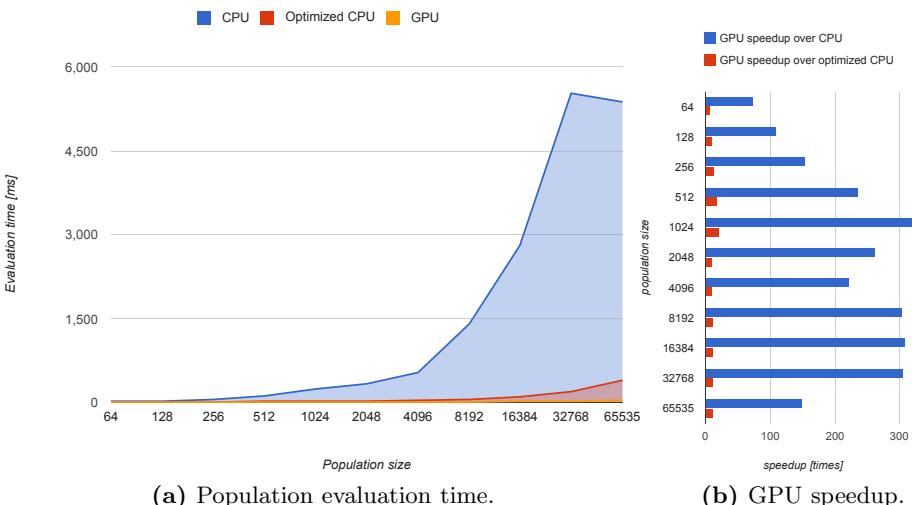


Fig. 2. Population evaluation time on CPU and GPU for the scheduling problem

with 2 dual core AMD Opteron processors at 2.6GHz and nVidia Tesla C2050 with 448 cores at 1.15GHz

The GPU implementation was 73 - 327 times faster than CPU implementation and 6 - 21 times faster than optimized CPU implementation of the same algorithm. This, along with the speedup achieved by parallel implementation of the DE process contributes to the overall improvement of results.

Second, we have investigated the quality of results obtained by the DE and compared it to the results obtained by a single-thread CPU implementation. Average fitness value of optimized schedules for different configurations of the algorithm are shown in [Table 1](#). The implementations are labeled as follows: type (CPU or GPU) / population size / limit of generations (thousands). We have used population sizes 1024 and 64 and the number of generations was selected so that the run time of all configurations was roughly the same. Other algorithm parameters were: $C = 0.8$, $F = 0.4$, and $T = 0.9$.

Table 1. Average fitness values of optimized schedules on CPU and GPU

ETC Matrix	Implementation				
	CPU/1024/300	CPU/64/300	GPU/1024/30	GPU/64/100	GPU/64/90
l-l-s	4026.055	4057.75	3964.43	3720.595	3746.435
l-l-i	2845.72	2832.415	2819.075	2689.105	2715.63
l-l-c	6807.95	6820.44	6691.08	6753.845	6760.52
l-h-s	180229	178137	174380.5	161464.5	159097.5
l-h-i	118625	116126.5	116943.5	110959.5	110964
l-h-c	412743	396145.5	389817.5	392084.5	388914.5
h-l-s	121705	124647	122083.5	112245	112203.5
h-l-i	86917.6	86305.25	86814.9	83255	82760.8
h-l-c	204818.5	205140.5	201239.5	200608.5	200650
h-h-s	5226555	5293475	5186865	4806145	4794115
h-h-i	3597320	3575925	3584020	3370300	3389225
h-h-c	11530150	11759800	11404550	11422600	11357100
time [sec]	31	31	29	34	31

In the table, the best average fitness is typeset in bold. In all cases, the schedule with best fitness was found by the GPU variant of the algorithm. Only in two cases, the fitness of schedules found by one of the CPU implementations was better than the GPU/1024/30 variant of the algorithm. Nevertheless, also in these cases the overall best schedule was found by an GPU implementation of the algorithm.

5 Conclusions

This paper introduces a GPU implementation of the differential evolution. The algorithm was described and the implementation on the nVidia CUDA platform was presented. The GPU implementation of differential evolution was used to

find good schedules for the independent tasks scheduling problem. With the help of the GPU, the fitness function for this problem is evaluated 6 - 21 faster in one case and 73 - 327 times faster in another case. In a direct comparison with CPU based implementation was shown that the differential evolution on GPU can find schedules with better average fitness.

Acknowledgement

This work was supported by the Ministry of Industry and Trade of the Czech Republic, under the grant no. FR-TI1/420.

References

1. Ali, S., Braun, T., Siegel, H., Maciejewski, A.: Heterogeneous computing (2002)
2. Braun, T.D., Siegel, H.J., Beck, N., Boloni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., Theys, M.D., Yao, B., Hensgen, D., Freund, R.F.: A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems (2001)
3. Fernandez-Baca, D.: Allocating modules to processors in a distributed system. *IEEE Trans. Softw. Eng.* 15, 1427–1436 (1989)
4. Munir, E., Li, J.Z., Shi, S.F., Rasool, Q.: Performance analysis of task scheduling heuristics in grid. In: 2007 International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3093–3098 (2007)
5. Izakian, H., Abraham, A., Snasel, V.: Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments. In: International Joint Conference on Computational Sciences and Optimization, CSO 2009, vol. 1, pp. 8–12 (2009)
6. Ritchie, G., Levine, J.: A hybrid ant algorithm for scheduling independent jobs in heterogeneous computing environments. In: Proceedings of the 23rd Workshop of the UK Planning and Scheduling Special Interest Group (2004)
7. YarKhan, A., Dongarra, J.: Experiments with scheduling using simulated annealing in a grid environment. In: Parashar, M. (ed.) GRID 2002. LNCS, vol. 2536, pp. 232–242. Springer, Heidelberg (2002)
8. Page, A.J., Naughton, T.J.: Framework for task scheduling in heterogeneous distributed computing using genetic algorithms. *Artificial Intelligence Review* 24, 137–146 (2004)
9. Carretero, J., Xhafa, F., Abraham, A.: Genetic algorithm based schedulers for grid computing systems. *Int. Journal of Innovative Computing, Information and Control* 3 (2007)
10. Abraham, A., Corchado, E., Corchado, J.M.: Editorial: Hybrid learning machines. *Neurocomput.* 72, 2729–2730 (2009)
11. Corchado, E., Abraham, A., de Carvalho, A.: Editorial: Hybrid intelligent algorithms and applications. *Inf. Sci.* 180, 2633–2634 (2010)
12. Hager, G., Zeiser, T., Wellein, G.: Data access optimizations for highly threaded multi-core cpus with multiple memory controllers. In: IEEE Int. Symposium on Parallel and Distributed Processing, IPDPS 2008, pp. 1–7 (2008)
13. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution A Practical Approach to Global Optimization. Natural Computing Series. Springer, Berlin (2005)

Collaborative Community Detection in Complex Networks

Camelia Chira and Anca Gog

Department of Computer Science, Babes-Bolyai University

1 Kogalniceanu, Cluj-Napoca 400084, Romania

{cchira,anca}@cs.ubbcluj.ro

Abstract. A collaborative evolutionary model is proposed to address the community structure detection problem in complex networks. The discovery of communities or organization of nodes in clusters (with dense intra-connections and comparatively sparse inter-cluster connections) is a hard problem of great importance in sociology, biology and computer science. Based on a natural problem-specific chromosome representation and fitness function, the proposed evolutionary model relies on collaborative selection and best-worst recombination to guide the search process efficiently towards promising solutions. The collaborative operators take into account information about an individual line best ancestor, global and worst individuals produced up to the current generation. The algorithm is able to detect non-overlapping communities in complex networks without the need to a-priori know the expected number of clusters. Computational experiments on several real-world social networks emphasize a good performance of the proposed algorithm compared to state-of-the-art models.

Keywords: complex networks, community detection, evolutionary algorithms, collaborative selection, collaborative recombination.

1 Introduction

The study of complex networks intensively preoccupied the scientific community in the recent years. Examples of complex networks in nature and society include metabolic networks, the immune system, the brain, the human social networks, communication and transport networks, the Internet and the World Wide Web. A complex system is characterized by the lack of central control and the fact that individual components are simple compared to the collective behaviour [3]. The study of real-world networks revealed features as degree distribution, average distance between vertices, network transitivity [2][20].

Another property intensively investigated is the community structure. A community in a network is a group of nodes densely connected but sparsely connected with the nodes belonging to other communities. The importance of community detection emerges from its many applications. For example, in social and biological networks it can help studying interactions between groups of people or

animals. Generally, a better understanding and visualization of network structure can be facilitated by an efficient community structure detection.

Many techniques have been proposed for identifying community structure in complex networks. Hierarchical (agglomerative and divisive) clustering [7] aims at discovering natural divisions of networks into groups based on similarity metrics for the strength of connection between vertices.

One of the most important contributions in this field comes from Girvan and Newman [6] who proposed a divisive algorithm that uses the edge betweenness as a weight measure of the edges. They introduced a measure of the quality for a certain partitioning called *modularity*. Given a division of a network in k communities, the modularity is defined as $Q = \sum_i (e_{ii} - a_i^2)$ where e is a symmetric matrix of size $k \times k$, each element e_{ij} represents the fraction of edges that connect nodes from community i to nodes in community j and $a_i = \sum_j e_{ij}$. Higher values of the modularity indicate stronger community structures.

Radicchi et al. [16] proposed a similar technique to [6] but they used a new metric, edge-clustering coefficient whose computation time is less than the betweenness centrality. The resistor network approach proposed by Newman and Girvan in [13] has been improved by Huberman and Wu in [10]. In [13] is described an algorithm for community detection using random walks. Balakrishnan and Deo proposed a new technique based on bibliometric similarity between all pairs of vertices in the graph [1]. Community detection based on extremal optimization [5] uses the network modularity proposed in [13].

The drawback of these techniques is the computational complexity that makes them unsuitable for very large networks. Evolutionary computation provides promising algorithms for addressing various NP-hard problems. Several evolutionary approaches to the problem of detecting community structures in complex networks have been proposed in the literature [18][15]. In [18], the authors propose a genetic algorithm based on the network modularity [13] as fitness function. Individuals are represented as arrays of integers, where each position corresponds to a node in the network and each value in the array represents the id of the corresponding community. In [15], a genetic algorithm called *GA-Net* is proposed to discover communities in networks. Individuals are represented as array of integers of size N (number of nodes in the network) where each position i , $i = 1 \dots N$ has a value j , $j = 1 \dots N$ with the meaning that nodes i and j will be placed in the same cluster. The concept of community score - a quality measure of a partitioning favouring highly intra-connected and sparsely inter-connected communities - is engaged as fitness function.

In this paper, a collaborative evolutionary algorithm based on the representation and fitness function from [15] is described for the community detection problem. The hybrid aspects [4] of the model refer to integrating a mechanism for facilitating the collaboration between individuals within a standard evolutionary model. This mechanism is inspired by the search strategies used in techniques such as Particle Swarm Optimization where the movement of a particle is guided by its own best known position in the search space as well as the swarm's best known position [11].

The main features of the proposed evolutionary approach refer to collaborative selection and recombination which sustain a balanced search process. Each individual has information about its best ancestor and the global optimal and worst solutions already detected. Selection of parents considers individuals which are not genetically related while the recombination operator takes into account the intermediary global best and worst solutions. Experiments on several real-life networks indicate a good and competitive behaviour of the introduced evolutionary model.

The paper is organized as follows: next section describes the main ideas behind collaborative evolutionary algorithms; section three presents the specific model to approach the community structure detection problem; section four discusses the computational experiments performed for a set of real complex networks; conclusions and some directions for further developments are drawn at the end of paper.

2 Collaborative Evolutionary Algorithms

An evolutionary algorithm based on collaborative operators is designed and investigated. The idea of collaborative evolutionary algorithms was introduced in [9]. The model proposed here further extends and modifies the evolutionary algorithm described in [9], the only common feature of the two models being the collaborative selection. In the *Collaborative Evolutionary Algorithm (CEA)* proposed here, individuals have knowledge about other individuals: the best potential solution (denoted by *GlobalBest*), the lowest fitness solution (denoted by *GlobalWorst*) already obtained in the search process and the individual's best ancestor (denoted by *LineBest*). The ancestors represent all individuals that have existed in one of the previous generations and have contributed to the creation of the current individual. If within a single ancestral line there are multiple ancestors with the same best fitness values, the closest ancestor is chosen. If within the two ancestral lines of an individual the best individuals have identical fitness, one of them is randomly chosen. In the initial population, the *LineBest* of each individual is the individual itself. *GlobalBest*, *GlobalWorst* and *LineBest* will guide the search process in the form of passing relevant genetic material to the individuals.

The n individuals within the population $P(t)$ are grouped by their *LineBest*. The clusters $A_1, \dots, A_k, k \leq n$, are formed so that they represent a partition of $P(t)$ and all the individuals that belong to one cluster have the same *LineBest*.

The concept of *Genetic Relatedness (GR)* describes the existing relation between two individuals that have a common best ancestor. Therefore, all the individuals belonging to a cluster are genetically related and every two individuals belonging to different clusters are not genetically related.

The GR – based Tournament Selection operator [8] chooses pairs of individuals that will be subject of recombination. The pair is selected in such a way that the two individuals are not genetically related. The aim of this selection is to force the combination of different genetic material in the crossover process, thus expanding the search to regions that have not been yet explored.

Two different clusters that will provide the pair of individuals are randomly selected. Then, an individual from each cluster is selected according to a tournament scheme.

The *Best-Worst Recombination (BWX)* scheme [7] is performed in an environment where each individual has extra knowledge about the best individual obtained so far in the search process and also about the worst individual obtained so far. On one hand, the goal is to use the good genetic material contained by the best individual obtained so far while performing recombination. On the other hand, it is aimed at avoiding transmitting to the offspring bad genetic material already contained by the worst individual obtained in the search process.

3 Proposed Collaborative Evolutionary Algorithm for Community Detection in Complex Networks

The CEA structure is engaged for dealing with the problem of finding communities in complex networks. The collaborative selection and BWX recombination operator described in section two are applied to a specific individual representation and fitness function.

In order to apply this algorithm to the specific problem of community detection in complex networks, the representation of an individual and the fitness function need to be specified. For the results reported in the current paper, these features are those proposed in [15].

Every individual is represented as an integer array of size N , where N is the number of nodes in the network. Each position i in the array assumes a value j (where j can be an integer number from 1 to N) which is translated to a partitioning in which nodes i and j belong to the same cluster. The number of clusters emerges after the chromosome is decoded to a partitioning in communities. The initial value of each chromosome is randomly generated with the restriction that each value j assigned to a position i means that edge (j, i) exists in the given network (it would be unfeasible to place two nodes in the same cluster from the beginning if they are not directly connected).

The community score proposed in [15] is engaged as the fitness function. Let A be the adjacency matrix of size $N \times N$ for a network with N nodes. The value a_{ij} of A at position (i, j) is 1 if there is an edge from i to j and 0 otherwise. The community score of a partitioning of A with k clusters S_1, \dots, S_k is defined as $\sum_{u=1}^k M(S_u) \times v_{S_u}$, where $M(S)$ is the power mean of S of order r (as defined in [15]) and v_S is the volume of sub-matrix $S = (I, J)$ (i.e. the number of entries a_{ij} with value 1 in the adjacency matrix A such that $i \in I$ and $j \in J$). The community score based fitness facilitates the detection of maximal and dense sub-matrices - corresponding to a partitioning with densely connected clusters and sparse connections between any two clusters.

Each individual in the initial population is created by selecting for each position $i, i = 1 \dots N$ a value j at random from the neighbours of i (i.e. such that $a_{ij} = 1$). Each individual carries information about its *LineBest* and has access to the current *GlobalBest* and *GlobalWorst* values. An elitist strategy is used

by which the best individuals from the current population are automatically selected for next population. The rest of the individuals are selected according to a roulette scheme. This intermediary formed population represents the pool from which parents are selected for recombination according to the collaborative selection scheme presented in section two.

The BWX operator is applied with a certain crossover probability and the best of the two offspring resulted replaces the first parent if it has a better fitness value. For every position in the parent chromosome, the corresponding value is inherited by the offspring if also found in *GlobalBest* at the same position, otherwise the value from the other parent is used. Furthermore, if *GlobalWorst* contains the same value at the current position the offspring receives a newly generated value from the available neighbours (without excluding the possibility of same value being selected). A simple mutation operator is applied with a certain probability to the offspring resulting from recombination. Mutation affects one randomly selected position from the chromosome by choosing a new value at random from the connecting nodes (weak mutation). The mutated offspring replaces the parent if the fitness is improved.

4 Computational Experiments

Experiments focus on the following real-world networks extensively engaged in testing community detection algorithms: Zachary's karate club network [19], American College Football Network [6] and Krebs network of political books [14]. The proposed CEA is applied for each network with the following parameters: population size is 500, number of generations is 50, crossover probability is 0.8, mutation probability is 0.2, elite size is 0.1 and the order r from the power mean $M(S)$ used in the fitness function is 0.5. Binary tournament is used for selecting an individual (to act as a parent) from a cluster. The best solution from 10 runs detected by our algorithm for each network is reported in Table II.

For each partitioning solution, the Normalized Mutual Information (NMI) - as given in [15,12] - is computed to evaluate the results. NMI represents a similarity measure between two partitions and is expressed as a real number between 0 and 1 with the following significance: a value of 1 for NMI means the two partitions compared are identical whereas a NMI value of 0 suggests that two completely different partitions are compared. In this paper, we compare the detected partitions for the considered networks to the known real non-overlapping partitions according to the results reported in [13]. Furthermore, the Newman modularity measure Q [13] is calculated and presented in Table II. The modularity quantifies the strength of the community structure: $Q = 0$ means no quality at all (similar to placing nodes in communities completely at random) whereas $Q = 1$ is the maximum possible value indicating a strong community structure. Newman and Girvan [13] emphasize that Q values for strong community structures partitionings are in practice within the interval $[0.3, 0.7]$.

The first network analysed is the Zachary's karate club network [19] which describes the friendship of 34 members of a karate club over two years. The

Table 1. Numerical results for several complex networks: Newman Modularity (Q) Normalized Mutual Information (NMI). Results are the best obtained over 10 independent runs.

Network	Q	NMI
Zachary's karate club network [19]	0.3990	0.8255
American College Football Network [6]	0.5925	0.9550
Krebs network of political books [14]	0.5015	0.6045

network has 34 nodes (representing the members) and 78 edges (representing the social interactions between members). The best community structure detected by the proposed algorithm is characterized by a NMI value of 0.8255 and $Q = 0.3990$.

The American College Football Network [6] contains 115 nodes representing teams from the schedule of Division I games in the 2000 season. Edges connect two teams having regular-season games. The known community structure for this network associates 8-12 teams to each conference with more frequent games between members of the same conference than between teams of different conferences [6]. The best partition detected by our algorithm has a modularity of $Q = 0.5925$ and $NMI = 0.9550$.

The Krebs network of political books [14] contains 105 nodes representing books on American politics bought from Amazon.com. Edges connect two books frequently purchased by the same person. The books were divided by Newman [14] in three groups according to their political alignment: liberal, conservative and a small group of books with other or no clear affiliation. Figure 1 presents a community structure having Newman modularity $Q = 0.5015$ and $NMI = 0.6045$ detected by the proposed algorithm.

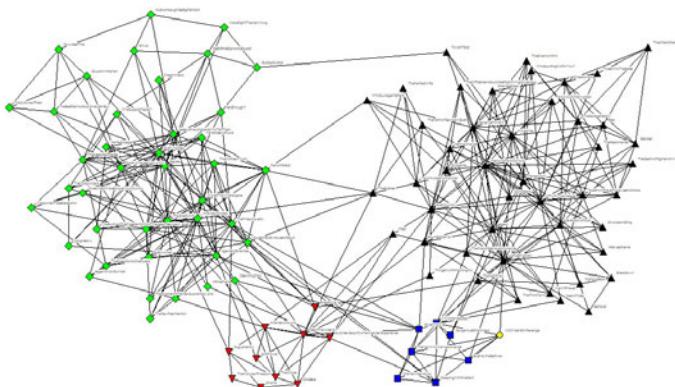


Fig. 1. Community structure for the Krebs network of political books

We further compare the obtained results for the American College Football and Krebs political books networks with those obtained by Girvan and Newman [6] and Pizzuti [15]. For these comparisons, the average value of NMI over 10 runs is considered and the results are presented in Table 2 (the compared values are those reported in [15]).

Table 2. Comparative results based on the average Normalized Mutual Information (NMI) obtained over 10 independent runs

Method	American College Football Network [6]	Krebs network of political books [14]
Girvan and Newman method [6]	0.8957	0.5107
GA-Net [15]	0.8825	0.5756
CEA	0.9317	0.5802

The comparative results indicate a good behaviour of the proposed collaborative evolutionary algorithm which was able to obtain better NMI values for the considered network data compared to state-of-the-art methods.

5 Conclusions and Future Work

A collaborative evolutionary algorithm is designed and implemented to detect communities in complex networks. The proposed algorithm relies on a problem-specific representation and fitness function engaged in the frame of a collaborative selection and recombination. The results obtained for several real network data are extremely encouraging, the algorithm being able to detect communities with a strong modularity and better normalized mutual information compared to well-known methods. Future work focuses on extending the numerical experiments to other network data and extendind the proposed model to deal with overlapping community detection in complex networks.

Acknowledgments. This research is supported by Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCSIS, Romania.

References

1. Balakrishnan, H., Deo, N.: Discovering Communities in Complex Networks. In: Proceedings of the ACM Southeast Regional Conference, pp. 280–285 (2006)
2. Barabasi, A.-L.: Linked: The New Science of Networks. Perseus, New York (2002)
3. Chira, C., Gog, A., Lung, R.I., Iclanzan, D.: Complex Systems and Cellular Automata Models in the Study of Complexity. Studia Informatica series, vol. LV(4), pp. 33–49 (2010)
4. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)

5. Duch, J., Arenas, A.: Community Detection in Complex Networks using Extremal Optimization. *Physical Review E* 72, 027104 -1 (2005)
6. Girvan, M., Newman, M.E.J.: Community Structure in Social and Biological Networks. *Proceedings of the National Academy of Sciences of the USA* 99, 7821–7826 (2002)
7. Gog, A., Dumitrescu, D., Hirsbrunner, B.: Best-Worst Recombination Scheme for Combinatorial Optimization. In: *Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM 2007)*, Las Vegas, USA, pp. 115–119 (2007)
8. Gog, A., Dumitrescu, D., Hirsbrunner, B.: New Selection Operators based on Genetical Relatedness for Evolutionary Algorithms. In: *Proceedings of IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore, pp. 4610–4614 (2007)
9. Gog, A., Dumitrescu, D., Hirsbrunner, B.: Community Detection in Complex Networks Using Collaborative Evolutionary Algorithms. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) *ECAL 2007. LNCS (LNAI)*, vol. 4648, pp. 886–894. Springer, Heidelberg (2007)
10. Huberman, B.A., Wu, F.: Finding Communities in Linear Time: a Physics Approach. *The European Physics Journal B* 38, 331–338 (2004)
11. Kennedy, J., Eberhart, R.C.: *Swarm Intelligence*. Morgan Kaufmann, San Francisco (2001)
12. Lancichinetti, A., Fortunato, S., Kertesz, J.: Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics* 11, 033015 (2009)
13. Newman, M.E.J., Girvan, M.: Finding and Evaluating Community Structure in Networks. *Physical Review E* 69, 026113-1 (2004)
14. Newman, M.E.J.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* 103, 8577–8582 (2006)
15. Pizzuti, C.: GA-Net: A Genetic Algorithm for Community Detection in Social Networks. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) *PPSN 2008. LNCS*, vol. 5199, pp. 1081–1090. Springer, Heidelberg (2008)
16. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and Identifying Communities in Networks. *Proceedings of National Academy of Science in USA* 101, 2658–2663 (2004)
17. Scott, J.: *Social Network Analysis: A Handbook*. Sage Publication, London (2000)
18. Tasgin, M., Bingol, H.: Community Detection in Complex Networks using Genetic Algorithm. *cond-mat/0604419* (2006)
19. Zachary, W.W.: An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* 33, 452–473 (1977)
20. Watts, D.: *Six degrees: The Science of a Connected Age*. Gardner's Books, New York (2003)

JCLEC Meets WEKA!

A. Cano, J.M. Luna, J.L. Olmo, and S. Ventura

Dept. of Computer Science and Numerical Analysis,
University of Cordoba, Rabanales Campus, Albert Einstein building,
14071 Cordoba, Spain
Tel.: +34957212218; Fax: +34957218630
`{i52caroa, i32luarj, juanluisolmo, sventura}@uco.es`

Abstract. WEKA has recently become a very referenced DM tool. In spite of all the functionality it provides, it does not include any framework for the development of evolutionary algorithms. An evolutionary computation framework is JCLEC, which has been successfully employed for developing several EAs. The combination of both may lead in a mutual benefit. Thus, this paper proposes an intermediate layer to connect WEKA with JCLEC. It also presents a study case which samples the process of including a JCLEC's EA into WEKA.

Keywords: WEKA, JCLEC, Evolutionary Algorithms, Data Mining.

1 Introduction

The huge amount of data in many society fields has attracted the need for obtaining useful information and knowledge from such data. Many application domains as marketing, sales, medical diagnosis or education, where it is essential to analyze many variables, have arised as typical domains to apply data mining (DM) techniques [3]. The main objective of such techniques is to support expert domains' decisions, especially border line decisions.

Likewise, tools for applying these techniques should allow expert domains to interact and visualize the DM process. Nowadays, there are a wide variety of DM tools as KEEL [1], WEKA [2], Knime [9], RapidMiner [10], etc., which have enough functionality to be used in a broad range of problems. WEKA has become one of the most popular DM workbenches and its success in researcher and education communities is due to its constant improvement, development, and portability. In fact, it can be easily extended with new algorithms. This tool, developed in the Java programming language, comprises a collection of algorithms for tackling several DM tasks as data pre-processing, classification, regression, clustering, association rules, also visualizing all the DM process. However, these algorithms are hardly used in situations where there are a huge amount of instances and attributes, situations where the execution becomes computationally hard. Instead, evolutionary algorithms (EAs) are very useful in this kind of circumstances, because they are powerful for solving problems that require a

considerable execution time. That is why it should be very interesting to harness the power of EAs also in the DM field, and WEKA appears to be a good platform to consider the inclusion of EAs.

Several customized EA classifiers have been previously integrated in WEKA, but there is no algorithm following an evolutionary schema in the standard WEKA distribution. Moreover, there is no evolutionary computation framework that has been included into this platform yet. An Open Source efficient and generic framework for evolutionary computation is JCLEC [2] (Java Class Library for Evolutionary Computation), which provides a high-level software environment to apply any kind of EA, with support for genetic algorithms (binary, integer and real encoding), genetic programming (Koza style, strongly typed, and grammar based) and evolutionary programming. This framework has been used in different problems, obtaining good results [28]. If we focus in the opposite direction, WEKA tool can help JCLEC by providing different data pre-processing algorithms to be applied before the EA. Because all of this, it seems very interesting to join the capabilities of WEKA with JCLEC framework, in order to allow the development of any kind of EA by using WEKA.

In this work, an intermediate layer that connects both WEKA and JCLEC is presented, showing how to include any evolutionary learning algorithm coded in JCLEC into WEKA. This enables the possibility of running this kind of algorithms in this well-known software tool as well as it provides JCLEC additional features and a graphical user interface.

This paper is structured as follows: Section 2 presents the architectonic design; Section 3 explains how to include a JCLEC EA in WEKA, illustrating this by means of an example; finally, some concluding remarks and future work are adumbrated.

2 Architectonic Design

WEKA's design allows to include new algorithms easily. Any new class is picked up by the graphical user interface without additional coding needed to deploy it in WEKA. To do so, new algorithms should inherit some properties from certain classes, which also indicate the methods that should be implemented. Though we focus on classification and association tasks, a similar approach could be followed for any other DM task such as clustering. New classification and association algorithms should extend the `AbstractClassifier` and `AbstractAssociator` classes, respectively. An abstract class called `ClassificationAlgorithm` that collects the common properties of classification EAs has been developed, which extends from `AbstractAlgorithm`. Thus, any classification EA will inherit from `ClassificationAlgorithm` the common properties and methods, and it will just specify its particular properties. Similarly, association algorithms also inherit from an abstract class called `AssociationAlgorithm`.

The architectonical design, developed to include JCLEC in WEKA, follows the schema represented in Figure 1, where it is depicted that WEKA provides two abstract classes from which any association or classification algorithm should

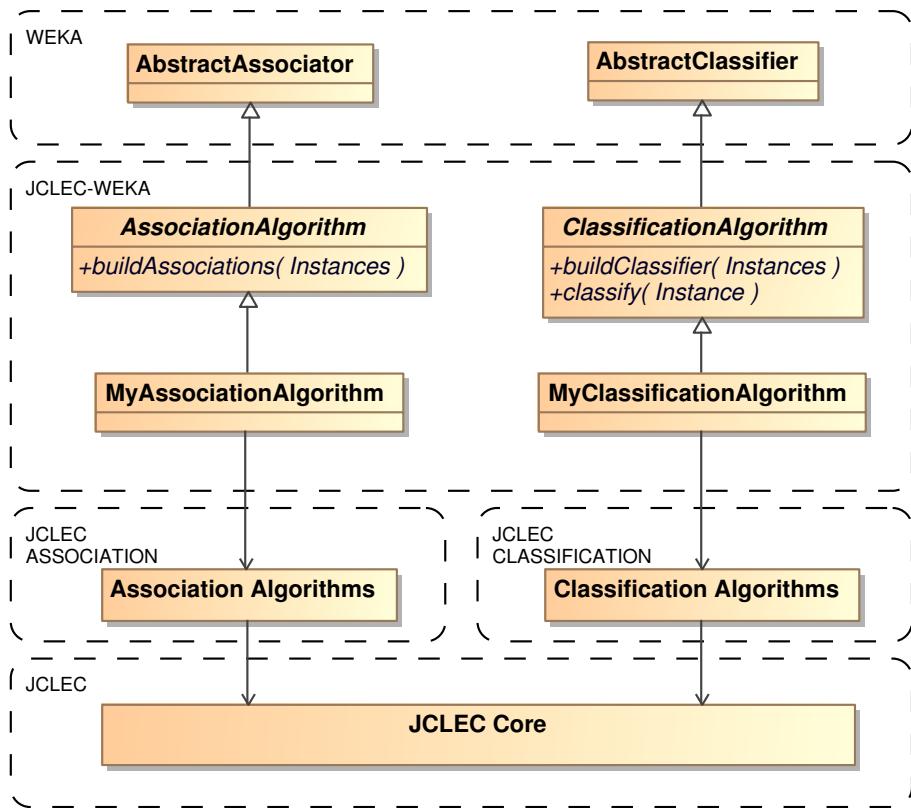


Fig. 1. Architectonical design

inherit, **AbstractAssociator** and **AbstractClassifier**. The abstract class **ClassificationAlgorithm** extends **AbstractClassifier**, and it is in charge of defining the properties and methods that any classification EA shares, e.g., population size, number of generations, crossover and mutation operators, etc. Similarly, regarding association task, the abstract class **AssociationAlgorithm** extends **AbstractAssociator**. Finally, any EA extends from one of these classes and calls the corresponding execution method of JCLEC. This way, the JCLEC algorithm is executed in WEKA tool.

Next, the methods of the intermediate layer that have to be taken into account when including any classification or association algorithm are described. In case of classification, the following two particular methods should be implemented:

- `void buildClassifier(Instances data)`. This method generates a classifier from a training set.
- `double classifyInstance(Instances instance)`. This method classifies a new instance using the classifier learned previously.

On the other hand, for association algorithms, only one particular method should be implemented:

- `void buildAssociations(Instances data)`. This method generates a set of association rules from a dataset.

Independently of whether a classification or an association algorithm is going to be included, it should implement the following methods:

- `Capabilities getCapabilities()`. This method determines if the algorithm is compatible with the type of each attribute in the dataset (e.g., numeric, nominal, etc.).
- `String globalInfo()`. Returns information about the algorithm, which will appear when selecting the *About* option in the graphical user interface.
- `TechnicalInformation getTechnicalInformation()`. Shows information about the author, year of the publication, etc.
- `void setOptions(String[] options)`. This method establishes the parameters of the algorithm, e.g., -P 50 -G 100—the former indicates the population size and the latter the number of generations.
- `String [] getOptions()`. Returns the set of parameters previously established.
- `Setters` and `getters` methods that set and get the parameter values.
- `String toString()`. This method shows the results obtained in the graphical user interface.

An additional and very useful tool contained in WEKA is the package manager. This tool allows the inclusion of any external library or code necessary to run new algorithms or features in WEKA, incorporating the files into the proper structure so that it avoids the developer to modify WEKA's source code. The directory structure of any new package has to fulfill a fixed anatomy. Hence, the JCLEC-WEKA intermediate layer is necessary in order to be able to instantiate the execution code of the algorithm, as shown in Figure 11. This way, a connection between WEKA and JCLEC is established.

3 Case Study

This section presents a sample case study that shows the functionality and process of a given classification algorithm. The considered algorihtm was presented in [11] by *Tan et al.*. This algorithm has been developed in JCLEC and it is based in a grammar guided genetic programming approach.

In order to show the advantages of incorporating JCLEC into WEKA tool, the classification algorithm presented by *Tan et al.*, has been added to a sample package¹ that can be easily included into WEKA. This package is the intermediate layer, which connects the WEKA interface with the JCLEC algorithm.

¹ This new package is also available in JCLEC site, <http://jclec.sourceforge.net>

In order to create this package, a new class with the name of the algorithm is created, which extends the `ClassificationAlgorithm` class explained in Section 2. In the `buildClassifier()` method, the algorithm is instantiated and the configuration parameters that are necessary for its execution are established. Then, the execution method of the algorithm is called and the classifier model is inferred using the training set. These steps make up the `buildClassifier()` method, as depicted in the following code:

```
public void buildClassifier(Instances instances)
{
    algorithm = new TanAlgorithm();
    configureMetadata(instances);
    algorithm.execute();
}
```

The `classifyInstance()` method receives a WEKA instance. It is necessary to turn this instance into a JCLEC instance. Then it is applied to the classifier, which returns the class predicted for this instance.

```
public double classifyInstance(Instance ins)
{
    ArffDataSet.Instance instance = dataset.new Instance();
    instance.setValues(ins.toDoubleArray());
    return algorithm.getClassifier().classify(instance);
}
```

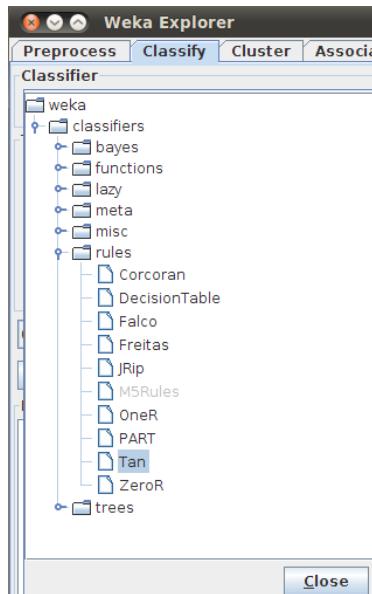


Fig. 2. WEKA's rules classifiers with EAs

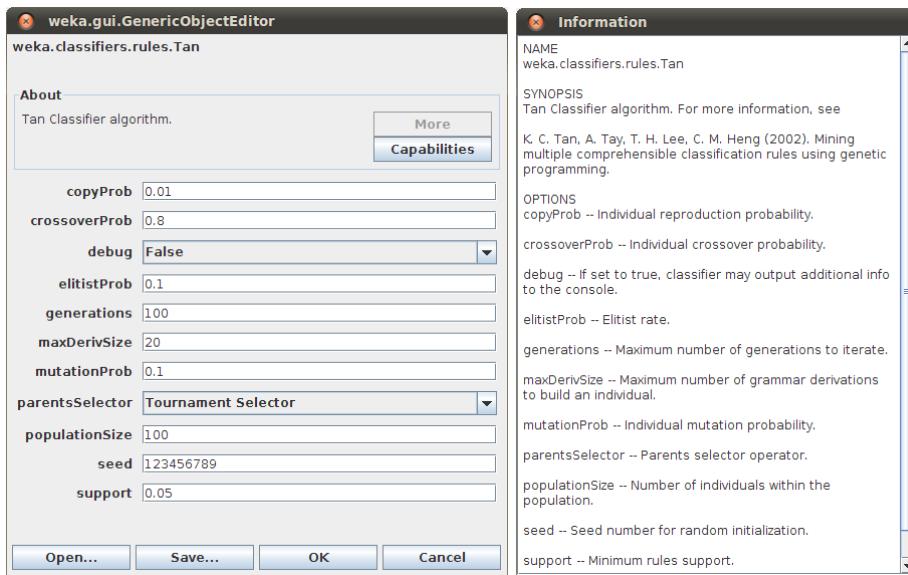


Fig. 3. Parameters configuration dialog box

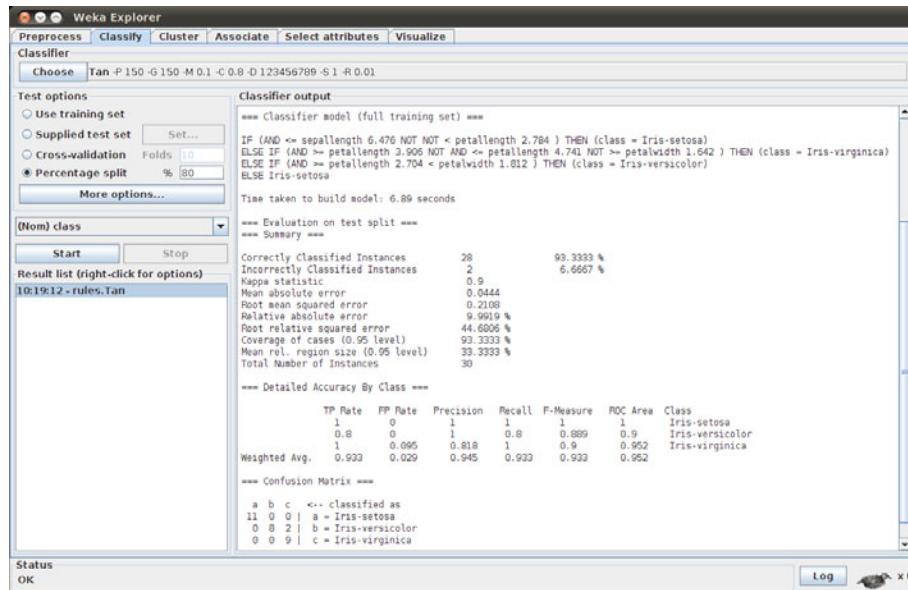


Fig. 4. Results of running *Tan et al.* EA under WEKA

The next step is to build the package using the Ant file provided by WEKA, which generates a zip file. This file has to be imported using the package manager. Once the package is charged, the classification algorithm appears beside the other

classification algorithms available in WEKA, as illustrated in Figure 2. Notice that other EAs have been included along with *Tan et al.*—*Corcoran and Sen* [4], *Falco et al.* [5] and *Freitas et al.* [6].

The next step is to specify the parameters needed to run the algorithm. Like EAs, this algorithm has a series of parameters such as population size, number of generations, crossover and mutation probability, seed, etc. All these parameters should be specified in a dialog box, as shown in Figure 3.

Finally, once the algorithm execution is carried out, the classifier obtained, computed metrics and the confusion matrix are displayed. Figure 4 shows the results of running the *Tan et al.* algorithm over the Iris dataset. This dataset along with other well-known datasets are available in the WEKA data folder.

4 Concluding Remarks and Future Work

This paper presents an intermediate layer that provides the possibility of connecting an evolutionary computing framework as JCLEC with WEKA. This synergy makes easier the final user to harness the power of EAs in the DM field, by using on the one hand WEKA’s graphical user interface and pre-processing tools, and on the other hand the power of EAs when applying them specifically to computationally expensive problems.

This work also opens up the chance of adding new features to WEKA. For instance, JCLEC has recently incorporated a new module that allows the execution of any genetic programming algorithm using graphics processing units [2]. Thus, the intermediate layer developed also offers WEKA the speed up and parallel processing capabilities inherent to graphics processing units architecture. In addition, this connection between JCLEC and WEKA permits not only the implementation of classification and association algorithms, but the implementation of EAs devoted to other DM tasks—multi-instance learning, multi-label classification, feature selection, etc.—also following binary, integer, real, expression tree and syntax tree codification schemes.

Acknowledgments

This work has been supported by the Regional Goverment of Andalucia and the Ministry of Science and Technology projects, P08-TIC-3720 and TIN2008-06681-C06-03 respectively, and FEDER funds.

References

1. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft. Computing 13, 307–318 (2008)

2. Cano, A., Zafra, A., Ventura, S.: Solving classification problems using genetic programming algorithms on gPUs. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 17–26. Springer, Heidelberg (2010)
3. Chen, M.-S.H.J.Y.P.S.: Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering* 8(6), 866–883 (1996)
4. Corcoran, A.L., Sen, S.: Using real-valued genetic algorithms to evolve rule sets for classification. In: Proceedings of 1st IEEE Conference on Evolutionary Computation, pp. 120–124 (1994)
5. De Falco, I., Della Cioppa, A., Tarantino, E.: Discovering interesting classification rules with genetic programming. *Applied Soft. Computing* 1(4), 257–269 (2001)
6. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag New York, Inc., Secaucus (2002)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *SIGKDD Explor. Newsl.* 11, 10–18 (2009)
8. Luna, J.M., Romero, J.R., Ventura, S.: Analysis of the effectiveness of G3PARM algorithm. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 27–34. Springer, Heidelberg (2010)
9. Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., Wiswedel, B.: KNIME: The Konstanz Information Miner. In: Data Analysis, Machine Learning and Applications, ch. 38, pp. 319–326. Springer, Heidelberg (2008)
10. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Ungar, L., Craven, M., Gunopulos, D., Eliassi-Rad, T. (eds.) KDD 2006: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 935–940. ACM, New York (2006)
11. Tan, K.C., Tay, A., Lee, T.H., Heng, C.M.: Mining multiple comprehensible classification rules using genetic programming. In: Proceedings of the 2002 Congress Evolutionary Computation. CEC 2002, pp. 1302–1307. IEEE Computer Society, Washington, DC, USA (2002)
12. Ventura, S., Romero, C., Zafra, A., Delgado, J.A., Hervás, C.: JCLEC: a Java framework for evolutionary computation. *Soft. Computing* 12, 381–392 (2007)

An Argumentation Framework for Supporting Agreements in Agent Societies Applied to Customer Support

Jaume Jordán, Stella Heras, Soledad Valero, and Vicente Julián

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València
Camino de Vera s/n. 46022 Valencia (Spain)
Tel.: (+34) 96 387 73 50; Fax: (+34) 96 387 73 59

Abstract. This work presents a system for customer support that integrates case-based reasoning functionalities with an argumentation framework for agent societies. This integration allows to automatically engage in agreement processes to decide the best solution to apply to solve an incidence that has been received in a call center. In this way, the quality of the response would be increased and the company running the call center can take advantage over its competitors in the market.

Keywords: Argumentation, Multi-Agent Systems, Case-Based Reasoning.

1 Introduction

Nowadays, companies have to offer a good customer support to take an advantage over their competitors. A good customer support depends, in many cases, on the experience and skills of its operators. A quick and accurate response to the customers problems ensures their satisfaction and a good reputation for the company and, therefore, it can increase its profits.

A common customer support system settled in a company consists of a network of operators that must solve the incidences (also known as *tickets*) received in a Technology Management Centre (TMC). TMCs are entities which control every process implicated in the provision of technological and customer support services to private or public organisations. In a TMC, there are a number of operators whose role is to provide the customers with technical assistance. This help is commonly offered via a call centre. The call centre operators have computers provided with a helpdesk software and phone terminals connected to a telephone switchboard that balances the calls among operators. Commonly, the staff of a call centre is divided into three levels: 1) Base operators, who receive customer queries and answer those ones from which they have background training; 2) Expert operators, who are the technicians in charge of solving new problems; 3) Managers, who are in charge of organising working groups, of assigning problems to specific operators and of creating generic solutions.

The solution applied to each problem and the information about the problem-solving process could be a suitable way to improve the customer support offered by the company. Case-Based Reasoning (CBR) systems have been widely applied to perform this task. A CBR system tries to solve a problem (case) by means of reusing the solution of an old similar case [8]. This solution is previously stored in a memory of cases (case-base) and it can either be retrieved and applied directly to the current problem, or revised and adapted to fit the new problem. The suitability of CBR systems in helpdesk applications to manage call centres has been guaranteed for the success of some of these systems from the 90s to nowadays [2][10][9].

These approaches propose systems for human-machine interaction where the CBR functionality helps the operators to solve problems more efficiently by providing them with potential solutions via the helpdesk software. In this paper, we extend a previous work that presented a CBR module that acts as a solution recommender for customer support environments [7]. The CBR module is flexible and multi-domain, in order to be easily integrable with any existing helpdesk software in a company. However, although this proposal provided successful results it also has some drawbacks. On one hand, to integrate the knowledge of all experts in a unique CBR module can be complex and costly in terms of data mining (due to extra large case-bases with possible out-of-date cases). On the other hand, to have a unique but distributed CBR could be a solution, but to assume that all operators are willing to share unselfishly their knowledge with other operators is not realistic. Also, in many companies the operators that work for a specific project can sign secrecy clauses and are not able to share certain knowledge. In this case, the modelling of the system as a Multi-Agent System (MAS) will be adequate. Finally, several experts could provide different solutions and hence, they need a mechanism to negotiate and reach an agreement about the best solution to apply.

Now, we propose to automate the system by representing the operators by means of software agents that can engage in an argumentation process to decide the best solution to apply for each new incidence.

Our approach is a hybrid system [5][1] that integrates an argumentation framework for agent societies to provide agents with argumentation capabilities and individual knowledge resources. This provides a computational framework for the design and implementation of Multi-Agent Systems (MAS) in which the participating software agents are able to manage and exchange arguments between themselves taking into account the agents' social context (e.g. their roles, dependency relations and preferences).

2 Argumentation Framework for Agent Societies

In this section we propose a computational framework for design and implementation of MAS taking into account the agents' social context. First, we propose a formal definition for an agent society. After that, we introduce the knowledge resources that agents can use to manage their positions (their solutions to the

problem at hand) and arguments. We follow a case-based approach to represent arguments. Thus, our knowledge resources are designed as cases with the common structure of cases in CBR systems.

From our point of view, an *agent society* is defined in terms of a set of *agents* that play a set of *roles*, observe a set of *norms* and a set of *dependency relations* between roles and use a *communication language* to collaborate and reach the global objectives of the *group*. This definition, based on the approach of [6] and [3], can be adapted to any open MAS where there are norms that regulate the behaviour of agents, roles that agents play, a common language that allow agents to interact defining a set of permitted locutions and a formal semantics for each of these elements.

However, we consider that the values that individual agents or groups want to promote or demote and preference orders over them have also a crucial importance in the definition of an argumentation framework for agent societies. These values could explain the reasons that an agent has to give preference to certain beliefs, objectives, actions, etc. Thus, they represent the motivation of agents to act in a specific way. For instance, an agent representing the manager of a company could prefer to promote the value of *wealth* (to increase the economic benefits of the company) over the value of *fairness* (to preserve the salaries of his employees). Also, dependency relations between roles could imply that an agent must change or violate its value preference order. For instance, a manager could impose their values to an expert or a base operator could have to adopt a certain preference order over values to be accepted in a group. Therefore, we endorse the view of [4], who stress the importance of the audience in determining whether an argument is persuasive or not for accepting or rejecting someone else's objectives. Thus, we have included in the above definition of agent society the notion of values and preference orders among them. Definition 1 provides a formal specification for our model of society:

Definition 1 (Agent Society). An Agent society in a certain time t is defined as a tuple $S_t = \langle Ag, Rl, D, G, N, V, Roles, Dependency, Group, val, Valpref_Q \rangle$ where:

- $Ag = \{ag_1, ag_2, \dots, ag_I\}$ is the set of I agents members of S_t in a certain time t .
- $Rl = \{rl_1, rl_2, \dots, rl_J\}$ is the set of J roles that have been defined in S_t .
- $D = \{d_1, d_2, \dots, d_K\}$ is the set of K possible dependency relations in S_t .
- $G = \{g_1, g_2, \dots, g_L\}$ is the set of groups that the agents of S_t form, where each $g_i = \{a_1, a_2, \dots, a_M\}, M \leq I$ consist of a set of agents $a_i \in A$ of S_t .
- N is the defined set of norms that affect the roles that the agents play in S_t .
- $V = \{v_1, v_2, \dots, v_P\}$ is the set of P values predefined in S_t .
- $Roles : Ag \rightarrow 2^{Rl}$ is a function that assigns an agent its roles in S_t .
- $Dependency_{S_t} : <_D^{S_t} \subseteq Rl \times Rl$ defines a reflexive, transitive and asymmetric partial order relation over roles.
- $Group : Ag \rightarrow 2^G$ is a function that assigns an agent its groups in S_t .
- $val : Ag \rightarrow V$ is a function that assigns an agent the set of values that it has.

- $\text{Valpref}_Q \subseteq V \times V$, where $Q = \text{Ag} \vee Q = G$, defines a irreflexive, transitive and asymmetric preference relation $<_Q^{S_t}$ over the values.

In this research, we focus on argumentation processes performed among a set of agents that belong to an agent society and must reach an agreement to solve a problem taking into account their social dependencies. Each agent builds its individual position in view of the problem (a solution for it). Thus, we assume that each agent has its individual knowledge resources to generate a potential solution. Also, agents have their own argumentation system to create arguments to support their positions and defeat the ones of other agents.

Ideally, an argumentation framework for agent societies should be easily interpreted by machines and have highly expressive formal semantics to define complex concepts and relations over them. Thus, we propose a case-based argumentation framework, which allows automatic reasoning with semantic knowledge in addition to the syntactic properties of cases. Reasoning with cases is specially suitable when there is a weak (or even unknown) domain theory, but acquiring examples encountered in practice is easy. Most argumentation systems produce arguments by applying a set of inference rules. In open MAS the domain is highly dynamic and the set of rules that model it is difficult to specify in advance. However, tracking the arguments that agents put forward in argumentation processes could be relatively simple.

In open multi-agent argumentation systems the arguments that an agent generates to support its position can conflict with arguments of other agents. These conflicts can be solved by means of argumentation dialogues between them. In our framework we propose two types of knowledge resources that the agents can use to generate, select and evaluate arguments in view of other arguments:

- Domain-cases database, with domain-cases that represent previous problems and their solutions. The structure of these cases is domain-dependent.
- Argument-cases database, with argument-cases that represent previous arguments and their final outcome.

In addition, arguments in our framework can be attacked by putting forward counter-examples, which are cases that are similar to a case (their descriptions match) but have different conclusions.

Next section shows the implementation of this argumentation framework for agent societies in the concrete domain of a helpdesk application.

3 Application Example: Customer Support

We have applied the argumentation framework described in the previous section to a customer support application domain. A prototype that provides support to the operators of a call centre has been implemented in a helpdesk application. In the prototype, the operators of a call centre are represented by agents that access to an automated helpdesk and argue to solve an incidence. Every agent has individual CBR resources and preferences over values.

Therefore, we consider a society S_t composed by call centre operators with three possible roles: base *operator*, *expert operator* and *manager*. The dependency relations of the prototype (charity, authorisation and power) are based on those proposed in [6]:

- Base operator $<_{charity}^{S_t}$ Base operator; Base operator $<_{authorisation}^{S_t}$ Expert operator; Base operator $<_{power}^{S_t}$ Manager
- Expert operator $<_{charity}^{S_t}$ Expert operator; Expert operator $<_{power}^{S_t}$ Manager
- Manager $<_{charity}^{S_t}$ Manager

which implies that in the society S_t any manager has the power to impose their position and arguments about the correct solution to apply to base operators and experts due to its higher hierarchy. Also experts are authorised to force base operators to apply their solutions for solving specific incidences because of their extended knowledge about certain types of problems.

Also, each agent can have his own values (e.g. economy, quality, solving speed), his own preferences over them and belong to different agent groups. Also, these groups can have their own social values. Following, we describe the different modules of the implemented prototype:

- **Magentix2:** to develop this prototype we have used the Magentix2 agent platform¹. Magentix2 is an agent platform that provides new services and tools that allow for the secure and optimised management of open MAS. In our system, this platform is used for the communication between the agents.
- **Domain CBR:** consists of a CBR module with data about previous problems solved in the call centre. This CBR is initialised with past tickets of the helpdesk application. To make a query, the user has to provide a ticket and a threshold of similarity. The domain CBR module searches the domain case-base and returns a list of similar domain-cases to the given ticket. The similarity algorithm used is based on the Euclidean distance between the attributes of the tickets. In addition, with every CBR cycle performed, the module adds, modifies or deletes one or more domain-cases of the case-base.
- **Argumentation CBR:** consists of a CBR module with argumentation data (previous arguments stored in the form of cases). An example of an argument case is shown in Table II. This CBR is initialised with some argument-cases generated in previous dialogues to solve past tickets. Once an agent has a list of potential solutions for a current incidence, it can generate an argument whose conclusion is the solution to apply. Then, this module is used to look for previous argumentation experiences where similar solutions where proposed and use this knowledge to select the best argument to propose in the current case (defending a particular solution), in view of the acceptance that had a similar argument in the past. Thus, argument-cases store information related to the domain and the social context where previous arguments (and their associated solution) were used. The information about

¹ <http://users.dsic.upv.es/grupos/ia/sma/tools/magentix2/index.php>

Table 1. Structure of an Argument-Case

	Domain Context	Premises*
PROBLEM	Social Context	ID
		Role
		Value Preference Relation
		ID
		Role
		Value Preference Relation
SOLUTION		ID
		Role
		Value Preference Relation
		Dependency Relation
	Solution Applied	
	Value Promoted	
	Acceptability State	
	Received Attacks	[Counter Examples]*

the domain consists of a set of features to compare cases (e.g. the type of incidence or the affected equipment) and information about the social context where the solution was applied (the agents that participated in the dialogue, their roles or their value preferences). The latter information can determine if certain arguments are more persuasive than others for particular social contexts (their acceptability state is set to *accepted*) and hence, agents can select the best solution to propose and an argument to support it.

- **Ontology parsers:** the case-bases of the domain CBR and the argumentation CBR are stored as OWL ² data of defined ontologies. In this way, heterogeneous agents can use them as common language to interchange solutions and arguments generated from the case-bases of the argumentation framework. Thus, the main advantage of using ontologies is that the structures and features of the cases are well specified and agents can easily understand them. The ontology parsers developed provide an API to read and write data in the case-bases of the system.
- **Argumentation agent:** it is an agent with a domain CBR and an argumentation CBR capable to engage in an argumentation dialogue to solve an incidence. These agents learn about the domain problem and the argumentation dialogue adding and updating cases into the domain and argumentation case-bases with each CBR run. Moreover, the agent can play any role defined before. In our prototype, this agent is a extension of Magentix2 Base-Agent³.
- **Commitment Store:** it is a resource of the argumentation framework that stores all the information about the agents participating in the problem-solving process, argumentation dialogues between them, positions and arguments. By making queries to this resource, every agent can read the information of the dialogues that it is involved in. It has been implemented as a Magentix2 Base-Agent to allow a good communication with the agents.

In order to show how the prototype works, the data-flow for the problem-solving process to solve each ticket is shown in Figure II and described below:

1. Some argumentation agents run in the platform and represent the operators of the call centre. The manager of the group acts as the *initiator* of the

² <http://www.w3.org/TR/owl2-overview/>

³ http://users.dsic.upv.es/grupos/ia/sma/tools/magentix2/archivos/javadoc/es/upv/dsic/gti_ia/core/BaseAgent.html

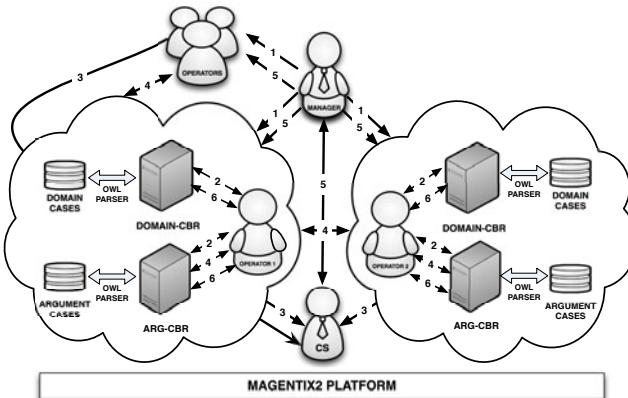


Fig. 1. Data-flow for the argumentation process of the helpdesk application

dialogue. This agent has a special behaviour to receive tickets to solve and create a new dialogue with the agents of his group. The process begins when a ticket that represents an incidence to solve is received by the initiator agent. Then, this agent sends the ticket to the agents of their group.

2. Each agent evaluates if it can engage in the dialogue offering a solution. To do that, the agent makes a query to its domain CBR to obtain potential solutions to the ticket based on solutions applied to similar tickets. If one or more valid solutions are retrieved, the agent will be able to defend a position in the dialogue. A valid solution is any domain case from the domain CBR with one or more solutions and with a similarity degree greater than a given threshold. Moreover, the agent makes a query to its argumentation CBR for each possible position to defend. With these queries a *suitability degree* of the positions is obtained. This degree represents if a position will be easy to defend based on past similar argumentation experiences. Then, all positions to defend are ordered and proposed from more to less suitability degree.
3. When agents have a position to defend (a proposed solution), these positions are stored by the commitment store agent. Thus, other agents can check the positions of all dialogue participants. Every agent tries to attack the positions that are different from its position.
4. The argumentation process consists on a series of steps by which agents try to defend its positions by generating counter-examples for the positions and arguments of other agents. A counter-example for a case is generated by retrieving from the domain case-base other case that matches the features of the former, but has a different conclusion. If different counter-examples can be generated, agents select the best attack to rebut the position of other agent by making a query to their arguments case-base, extending each case with the current social context. In this way, agents can gain knowledge about how each potential counter-example worked to attack the position of an agent in a past argumentation experience with a similar social context.

5. The dialogue finishes when it is stopped by more than a specified time, without new positions or arguments. The initiator agent makes queries to the commitment store agent to determine if the dialogue must finish. Then, this agent retrieves the active positions of the participating agents and the most frequent will be the solution (or a random choice in case of draw). The initiator agent communicates the solution to the participating agents.
6. Finally, each agent updates his argumentation CBR with the new arguments produced in dialogue and his domain CBR with the solution applied.

4 Conclusions

This paper proposes the extension of a CBR module for customer support with an argumentation module that allows agents to engage in a dialogue to decide the best solution for an incidence received in a call center. The main advantage of this approach over similar proposals is that agents can argue to reach an agreement about the best solution taking into account their social context.

Acknowledgement

This work is supported by the Spanish government grants CONSOLIDER INGENIO 2010 CSD2007-00022, TIN2008-04446 and TIN2009-13839-C03-01 and by the GVA project PROMETEO 2008/051.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72(13-15), 2729–2730 (2009)
2. Acorn, T., Walden, S.: Smart: Support management automated reasoning technology for compaq customer service. vol. 4, pp. 3–18 (1992)
3. Artikis, A., Sergot, M., Pitt, J.: Specifying norm-governed computational societies. *ACM Transactions on Computational Logic* 10(1) (2009)
4. Bench-Capon, T., Atkinson, K.: Abstract Argumentation and Values. In: *Argumentation in AI*, pp. 45–64 (2009)
5. Corchado, E., Abraham, A., Carvalho, A.C.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)
6. Dignum, V.: PhD Dissertation: A model for organizational interaction: based on agents, founded in logic. Ph.D. thesis (2003)
7. Heras, S., García-Pardo, J.A., Ramos-Garijo, R., Palomares, A., Botti, V., Rebollo, M., Julian, V.: Multi-domain case-based module for customer support. *Expert Systems with Applications* 36(3), 6866–6873 (2009)
8. Kolodner, J.: Case-based Reasoning (1993)
9. Roth-Berghofer, T.R.: Learning from homer, a case-based help-desk support system, pp. 88–97 (2004)
10. Watson, I.: Applying case-based reasoning. Techniques for enterprise systems. Morgan Kaufmann Publishers, Inc., San Francisco (1997)

Finger Vein Pattern Extraction Algorithm

Michał Waluś, Jan Kosmala, and Khalid Saeed

AGH University of Science and Technology,
Faculty of Physics and Applied Computer Science,
ul. Mickiewicza 30, 30-059 Kraków, Poland
michal.walus@gmail.com, jan.kosmala@gmail.com,
saeed@agh.edu.pl

Abstract. In this paper the outline of complete vein pattern recognition system is introduced. An innovative method for finger vein image binarization is given in detail. The main purposes in the developed method are: lower time consuming comparing to other methods, simplicity and promising results achievement in spite of working with low contrast images. Both the algorithm and the experimental results of the pattern extraction are presented in detail. The achieved results are compared with other binarization techniques to show the superiority of the worked out algorithm over the others. The algorithm has proved to be of less computational complexity and easier implementation.

Keywords: Finger-vein images, vein, position-gray-profile, binarization, vein-extract algorithm, thresholding.

1 Introduction

There can be seen a significant increase of interest in biometric systems and their applications in human verification and identification. Vein pattern is count as a physiological stable biometric feature. The idea of using vein patterns as a form of biometric technology was first proposed in 1992 [1]. Because of the low costs of creating the sample recorder and easiness and acceptability in their acquisition, researchers are increasingly dealing with this issue developing more and more new algorithms in this area of biometric systems.

Vein patterns strongly differ among individuals. With high probability it is true that vein patterns are unique to every individual, even among twins. This follows from the fact that the circulatory system arises in the process of embryo development and together with human development, this basic feature remains unchanged. Hand vein patterns are usually the most exposed part of veins in human body, so it can be detected by low-resolution cameras without any other devices.

In image analysis, binarization is one of the important steps in the image pre-processing stage. Despite the high dynamics of new methods development in most cases the simplest methods are selected. The ‘simplest’ means in particular ‘easy to implement’. A thorough survey over automated image thresholding selection can be found in [2].

In the case of binarization of non-typical images, where general methods do not work correctly, we are looking for less popular or special designed solutions. This is when image after binarization does not include information about the elements necessary for further analysis. In this paper the algorithm of finger vein pattern extraction is given in detail with a general reference to the whole designed biometric system.

2 How Others See the Finger-Vein Pattern Extraction Problem

In [3] the authors propose a method based on the position-gray-profile curve. Using local minimal gray-profile-curves across the image for a given line, the local minima are located line by line. Unfortunately, in some cases veins cannot be detected. In low contrast images, veins are not visible in the histogram as peaks. For this reason the algorithm uses six profiles to extract exactly the vein positions associated with high computational complexity.

Another method with satisfactory results is presented in [4]. This algorithm selects a random point on the image and tries to move in valleys, minimal values in the histogram and its surroundings, pixel by pixel for each image line. To receive good effects tracking has to run multiple times. By random operations and multiple tracking repetitions, the computing complexity is high. Also the time complexity is not satisfactory. Despite these drawbacks [5], [6], this algorithm is still used. Some experiments on vein image extraction are based on the non-adaptive technique of curvlets for multi-scale object representation [7], [8]. In [9] authors use dyadic wavelet to transform image into wavelet coefficients domain. Researchers propose different thresholding methods. One of them (local adaptive threshold) is described and used in [10] and [11].

Appropriate thresholding algorithm selection for this kind of images is difficult although it has significant influence on the pattern extraction. In this paper the authors present an innovative approach to solve the problem of vein extraction from the inside part of hand fingers.

3 System Description

The designed biometric system in this work is to for human recognition on the basis of vascular pattern of his finger. The general recognition system block diagram is presented in Fig.1 with its application to the biometric system.

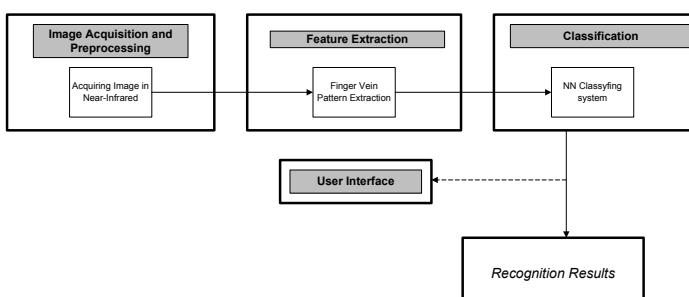


Fig. 1. Block diagram of recognition system and its relation to the biometrics

The complete system contains a laboratory-designed device (built by the second author) to collect vein images, database to store patterns and recognition software layer. To capture an image, a simple internet camera was used after removing its near-infrared filter. The light source of the device was built using self-constructed LED matrix (wavelength 850 nm). The basic electronic circuit is presented in Fig. 2.

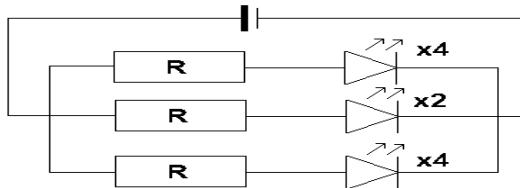


Fig. 2. Basic circuit of the illumination device in the laboratory system

The classification stage of the system involves implementing a neural network algorithm to classify the resulting images. The network is built with 3 layers while the learning is according to the delta method. The sigmoid function $f(x)$ as given in eq. (1) is the neuron activation function in our model, with x to be the value of the sum of multiplied weights and inputs. The value of β is considered to be 1.

$$f(x) = \frac{1}{1 + e^{-\beta x}}. \quad (1)$$

Recognition module takes at its input an 8-bit gray scale image and extracts the pattern. The luminosity value of each pixel from the image is normalized to [0 - 1]. The created array is the input to the neural network. The maximal value is taken from NN output vector as the user identifier. If the value is greater than ϵ parameter stored in database, the user is rejected. The parameter ϵ is the maximal value of the difference between the stored value and the value for a person under recognition.

We collected 348 images from 29 persons. From each person 6 fingers from both hands were scanned twice. The collected data set is divided between training and testing assemblies. This helps receive reliable results (the values of the FAR and FRR coefficients).

4 Finger Vein Pattern Extraction Algorithm Description

The proposed algorithm is a dedicated solution used for separating the vein pattern (object) from human hand fingers (background). Pseudocode is listened below.

```

in_img = ReadInputImage()
for pixels from in_img
    p0,p1,p2,p3,p4,p5,p6,p7 //neighbourhood of pixel px
  
```

```

val = max[ (abs(p0-px) , abs(p2-px) ,
            abs(p4-px) , abs(p6-px)) ]/sqrt(2)
val=[abs(p7-px) , abs(p3-px) , abs(p5-px) ,
      abs(p1-px),val]

grad_img(pixel)=max(val);

end

grad_img = Threshold(grad_img,threshold)
grad_img = MorphologicalOpen(grad_img)
finger_img = ExtractFingerArea(in_img,grad_img)

for lines from finger_img
    [WK_Line_L, WK_Line_R] = ComputeWKFunctions(line)
    out_img(line) = ExtractVein(WK_Line_L, WK_Line_R)
end

out_img = MedianFilter(out_img) //optional

```

The first step involves separating the area occupied by the finger from the input image. This is done by determining the gradient magnitude. Consider the pixel intensities with spatial positions:

$a(i-1,j-1)$	$b(i-1,j)$	$c(i-1,j+1)$
$d(i,j-1)$	$e(i,j)$	$f(i,j+1)$
$g(i+1,j-1)$	$h(i+1,j)$	$l(i+1,j+1)$

Fig. 3. Kernel of the mask

A fast estimate of the gradient magnitude at (i,j) position of the image [12] is given in eq. (2).

$$Grad_{img}(i, j) = \max(|d - e|, |f - e|, |h - e|, |b - e|, val). \quad (2)$$

with

$$val = \frac{\max(|a - e|, |c - e|, |l - e|, |g - e|)}{\sqrt{2}}. \quad (3)$$

This is based on the fact that for an analytic function the gradient magnitude is invariant to rotation and it is therefore equal to the maximum magnitude of the directional derivatives [12].

After this operation, the gradient image is obtained to be binarized using static pre-specified threshold value. This value depends on the image quality. Then the next stage morphological operations (dilate and erosion) are performed. This is done in

purpose of noise reduction without affecting the characteristics of the region of interest [13] for visual effect improvement. In the process of separating the area occupied by the finger both the original and gradient images are needed.

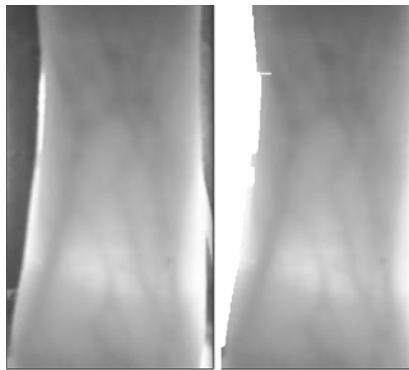


Fig. 4. Image area of interest before and after separating the area occupied by the finger

The next stage relies on the position-gray-profile curve [3] examination for each line of the image.

Define functions WK_{LineL} and WK_{LineR} :

$$WK_{LineL(i)} = \frac{n \left[\sum_{k=i}^{i+val-1} Line(k) \right] + \left[\sum_{j=i-1}^{i+val-2} Line(j) \right]}{n+1} \quad (4)$$

where $i \in [2, \text{size}(Line) - val - 1]$, $n, val \in N^+$

$$WK_{LineR(i)} = \frac{n \left[\sum_{k=i-val+1}^i Line(k) \right] + \left[\sum_{j=i-val}^{i-1} Line(j) \right]}{n+1} \quad (5)$$

where $i \in [\text{size}(Line) - 1, val]$, $n, val \in N^+$

Position-gray-profile example curve of the image is presented in Fig. 5 (the line is taken from the cut border image). The solid line indicates the original intensity levels in a given image line, the dotted line indicates the resulting graph of the function

WK_{LineL} and the third line (dash-dotted) indicates the resulting graph of the function WK_{LineR} .

After determining the values of the functions WK_{LineR} and WK_{LineL} the next stage is to determine local minima for those functions. The shift corresponding to the local extremes of WK_{LineR} and WK_{LineL} functions can be seen.

Local minima in a given line of the image correspond to the area occupied by the veins. The main task of the defined functions is to extract from lines the areas they correspond to. Those functions also have the task of values averaging in order to eliminate cases where small differences in the intensity of each pixel may generate a vein location where actually they do not exist.

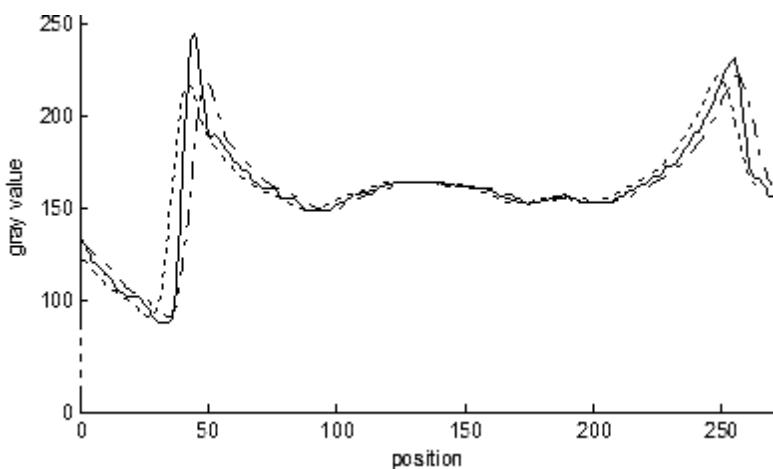


Fig. 5. Position-gray-profile example curve

The last stage is to extract the vein pattern using the information prepared in previous stages. For each line of the image the values of the corresponding local minima are taken from WK functions. In addition, the image is one more time binarized. For this purpose a static threshold value is used. Median filtering is applied to reduce the existing noise and smooth the edges of the vein pattern.

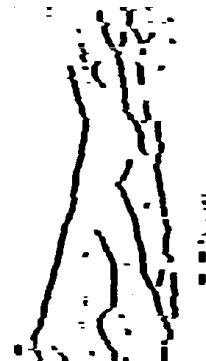
5 Results and Conclusions

A sample vein image together with the result of the proposed algorithm performance are presented in Fig. 6. Figure 7 (a - e) shows comparison with other techniques, mainly dynamic global and local thresholding. It can easily be seen that the authors' algorithm generates significantly better results.

The conditions associated with time, computational complexity and easiness of implementation have been fulfilled. For those reasons, the authors have used the proposed method as a part of the vein-based human recognition system.



(a) Original image



(b) Proposed method

Fig. 6. A sample vein image: (a) original image, (b) the resulting image after applying the authors' algorithm



(a)



(b)



(c)



(d)



(e)

Fig. 7. (a) Original image, (b) Proposed method, (c) Global dynamic threshold, (d) Local dynamic threshold with 14×14 mask, (e) Local dynamic threshold with 20×20 mask

Acknowledgments

The authors are indebted to Dr. Krzysztof Malarz, the Deputy Dean of the Faculty of Physics and Applied Computer Science at AGH University of Science and Technology, for his support to do this research.

References

1. Kono, M., Ueki, H., Umemura, S.: A New Method for the Identification of Individuals by Using of Vein Pattern Matching of a Finger. In: Fifth Symposium on Pattern Measurement, Yamaguchi, Japan (2000)
2. Sankur, B., Sezgin, M.: A Survey Over Image Thresholding Techniques And Quantitative Performance Evaluation. *Journal of Electronic Imaging* 13(1), 146–165 (2004)
3. Hong, J., Shuxu, G., Xueyan, L., Xiaohua, Q.: Vein Pattern Extraction Based on the Position-Gray-Profile Curve. *Image and Signal Processing, CISP* (2009)
4. Miura, N., Nagasaka, A., Miyatake, T.: Feature extraction of finger-vein patterns based on repeated Line cracking and its application to personal identification. *Machine Vision and Applications* (15), 194–203 (2004)
5. Tang, D., Huang, B., Li, R., Li, W.: A Person Retrieval Solution Using Finger Vein Patterns. In: 2010 20th International Conference on Pattern Recognition (ICPR), August 23–26, pp. 1269–1272 (2010)
6. Huang, B., Dai, Y., Li, R., Tang, D., Li, W.: Finger-vein Authentication Based on Wide Line Detector and Pattern Normalization. In: 20th International Conference on Pattern Recognition (ICPR), August 23–26, pp. 1269–1272 (2010)
7. Zhang, Z., Ma, S., Han, X.: Multiscale Feature Extraction of Finger-Vein Patterns Based on Curvlets and Local Interconnection Structure Neural Network. In: The 18th International Conference on Pattern Recognition ICPR (2006)
8. Qian, X., Guo, S., Li, X., Zhong, F., Shao, X.: Finger-vein Recognition based on the Score Level Moment Invariants Fusion. In: International Conference on Computational Intelligence and Software Engineering. CiSE 2009, December 11-13, pp. 1–4 (2009)
9. Xueyan, L., Shuxu, G., Fengli, G., Ye, L.: Vein Pattern Recognitions by Moment Invariants. In: The 1st International Conference on Bioinformatics and Biomedical Engineering. ICBBE 2007, July 6–8, pp. 612–615 (2007)
10. Mulyono, D., Shi Jinn, H.: A Study of Finger Vein Biometric for Personal Identification. In: International Symposium on Biometrics and Security Technologies -ISBAS (April 2008)
11. Ding, Y., Zhuang, D., Wang, K.: A Study of Hand Vein Recognition Method. In: 2005 IEEE International Conference on Mechatronics and Automation, July 29–August 1, vol. 4, pp. 2106–2110 (2005)
12. http://www.thbcomponents.com/image/Sobel_operator_implementation_b9b9300e_0e92_44d4_a6a5_7cbd76414e.html (11.01.2011)
13. Tadeusiewicz, R., Korohoda, P.: Komputerowa analiza i przetwarzanie obrazów, Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków (1997) (in Polish)
14. Wozniak, M., Zmyslony, M.: Designing fusers on the basis of discriminants – evolutionary and neural methods of training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 590–597. Springer, Heidelberg (2010)
15. Derrac, J., García, S., Herrera, F.: A First Study on the Use of Coevolutionary Algorithms for Instance and Feature Selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)

An Exploratory Research on Text-Independent Speaker Recognition

Mohammad Kheir Nammous¹, Adam Szczepański², and Khalid Saeed²

¹ Autonomous University of Barcelona

Plaça Cívica 08193 Bellaterra, Barcelona, Spain

m.k.nammous@gmail.com

² AGH University of Science and Technology

Faculty of Physics and Applied Computer Science

Al. Mickiewicza 30, PL-30059 Cracow, Poland

{aszczepa, saeed}@agh.edu.pl

Abstract. This work will handle an exploratory method for text-independent speaker identification. The purpose of the work is to apply a new home-made voice database to the modified algorithm to increase its efficiency for further development and using. Some particular methods of text-independent speaker recognition are described. The proposed techniques do not need signal pre-processing and hence they are time saving easy to implement. The used database obtained from news in a broadcasting channel with varieties of speech tone, emotions, accents in different recording conditions. The feature extraction stage of the proposed approaches is based on Burg's method of linear prediction and the minimal eigenvalues of Toeplitz matrices. The classification part is based on Neural Network - probabilistic and radial types.

Keywords: text-independent speaker recognition, Toeplitz matrices, Burg estimation, Probabilistic NN, Radial basis function NN.

1 Introduction

Speech analysis is one of the most widely considered research fields of behavioral biometrics. The voice contains many useful parts of information about the speaker, including age, gender, emotions or level of stress and tiredness. The most important idea is the possibility of identification of particular person using his voice samples. The voice characteristics of people are unique enough to make an attempt of identification. There are two main ways of speaker identification:

For most of the applications the text-dependent techniques are sufficient, it relies on features extracted from keywords speakers are obligated to say. Such algorithms are easier to implement, usually fast, efficient, and fulfill most of the security and interface control needs. On the other hand there are situations where the person has to be identified using random voice samples, for example eavesdropping on criminals. Also in systems where identification is based both on password and speaker recognition, the text-independent speaker recognition does not allow to re-teach the system when the password changes.

Early approaches to text-independent speaker recognition started in 1960s [1]. The research was done on the basis of pattern matching of particular common word-samples spoken by known persons on a telephone conversation. The results were quite good, acquiring 78% of correct recognition with the time-energy analysis and 98% of correct recognition in the energy-frequency domain on 100 samples. The method was not yet fully text independent but could identify the speaker without the need of any keywords and without his knowledge about being identified.

One of the most interesting early approaches was the experiments with text-independent speaker identification and recognition during long distance phone calls undertaken in the mid 1980s. The tests on 40 samples generated by 10 speakers gave the results of 68% to 76% of correct recognition depending on the used method [2].

The developed and widely used methods during 1990s are described in detail in many comparative papers published in mid 1990s, for example [3], [4]. One of the them was the Gaussian mixture speaker model which gave very good results of recognition [5]. More recent approach to this model is described in [6], [7]. This method is often used as the reference method to most modern approaches [8]. More and more approaches rely on wavelet transform and Mel-Frequency Cepstral Coefficients. It is based on decomposing signal using Discrete Wavelet Transform (DWT) into approximations and details coefficients and then using them to compute MFCCs. The efficiency of this method varies from 86.77% to 96.25% of correct recognition for over 2000 samples [9]. Other models based on using features extracted from Nyquist filter bank acquiring up to 91.06% of correct recognition [10]. The most advanced systems are capable of acquiring up to 200 voice features for text-independent speaker recognition with almost 100% accuracy and up to 15 000 features with over 80% accuracy [11].

The authors have also experience in the field of speech and speaker recognition. One of the earlier approaches of the authors was an image-based algorithm for spoken Arabic-digit identification. The algorithm was tested with different types of neural networks (Probabilistic Neural Networks and Radial basis function NNs) as classifiers [12]. This approach was also implemented again and slightly modified to test its effectiveness on spoken Polish digits [13]. The most recent approaches were the revisions and modifications of this algorithm to work as a text-independent speaker recognizing system [14].

2 Implemented Algorithms

The process of speaker identification methods, generally, is divided into three stages: preprocessing, feature extraction and classification. The algorithms presented in this work are based on the algorithms used in [14]. The main and essential change is the partial elimination of the first stage and the only remaining related steps are resampling to 10K and white fragments elimination (where no voice content).

For feature extraction, Burg estimation and minimal eigenvalues of Toeplitz matrices are followed. The signal is divided into equal parts with 2 sec. length, for which

the features are extracted and classified. When using Burg's method of estimation, however, we need to specify the prediction order (P), and the FFT size, called the length of FFT ($NFFT$).

In the classification stage we use two kinds of neural networks: the probabilistic and the Radial basis function NN. The inputs to the NN are the feature vectors while the output is the classified speaker. All of the algorithms were implemented in MATLAB environment.

2.1 Description of Samples

Taking into consideration the purpose of this work, we had many difficulties in preparing the database of samples. It had to be large enough, systematic and the most difficult issue was to be realistic. The samples had to be taken from regular everyday dialogs or speeches. The choice of this kind of samples guarantees the accurate and reliable measurement of the effectiveness of the algorithm in contrast to the experiments conducted on the samples recorded in artificial environment (for example recorded studio) or with artificial noise. A description of a selected group of samples is presented in Table 1 using initials instead of the real names of the speakers.

Table 1. A selected group of voice samples

Speaker	Duration	Training	Testing
G. O.	0:08:55	0:02:00	0:06:55
H. J.	0:11:01	0:02:00	0:09:01
M. A.	0:12:58	0:02:00	0:10:58
A. S. N.	0:13:12	0:02:00	0:11:12
H. A.	0:16:48	0:02:00	0:14:48
W. A.	0:06:19	0:02:00	0:04:19
M. R.	0:06:56	0:02:00	0:04:56
R. H.	0:07:30	0:02:00	0:05:30
A. M.	0:07:30	0:02:00	0:05:30
F. K.	0:08:09	0:02:00	0:06:09
J. A.	0:11:41	0:02:00	0:09:41
A. A. A.	0:12:54	0:02:00	0:10:54
Jo. N.	0:13:32	0:02:00	0:11:32
Ja. N.	0:16:34	0:02:00	0:14:34
L. A.	0:19:32	0:02:00	0:17:32
Total	2:53:31	0:30:00	2:23:31
Time Percentage	100%	17.29%	82.71%

The speech samples were recorded from the known Aljazeera news broadcasting. In order to obtain a regular database we first worked on the recorded video. The voice was then extracted, collected and separated for each speaker. This allowed to collect speeches in different sessions and situations, and of course only those samples which satisfied the assigned conditions for our system. Many recorded samples were rejected or removed from the base because they did not satisfy the conditions, or simply they were not within the time limit for the given speaker. The high number of recoded

video hours was reduced to about three hours of concentrated samples of 15 different speakers, often one speech belonged to more than one real TV session or program. The recorded samples also varied in quality from broadcasting of news studio to transmission from reporters via cell phone. In all these situations the selected fragment had to be clear from any additional effects in the background. All voices were resampled to 10 kHz in order to reduce the amount of data for further analysis without any information loss. Moreover, fragments with no voice data were eliminated, as mentioned above. For every person the first 2 minutes were taken for training purposes and the rest was used for testing.

2.2 Feature Extraction

The first of the algorithms presented in this paper uses the characteristic points acquired from Burg's method of linear prediction which smoothes the signal transformed by Fast Fourier Transform from time-amplitude into amplitude-frequency domain. Figure 1 shows the plot of frequency spectral estimation as suggested by Burg when performing in MATLAB. The details of Burg method of estimation is given in [15].

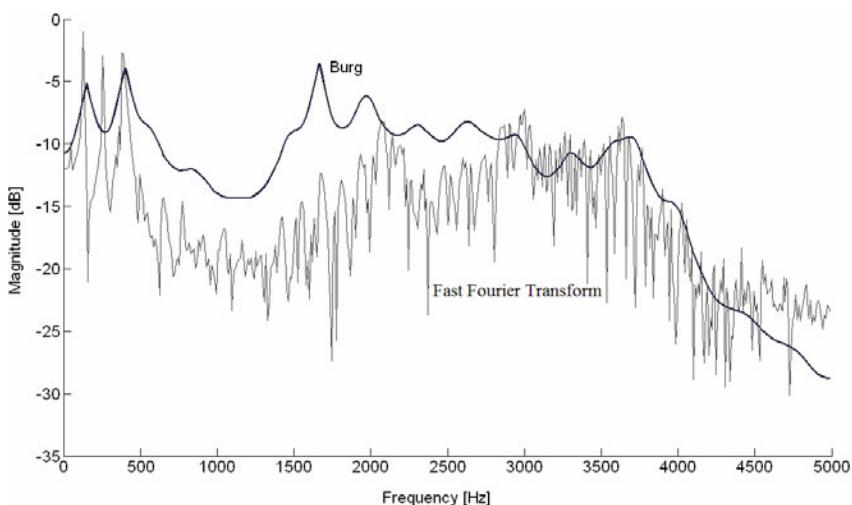


Fig. 1. Frequency spectral estimation - Burg's model compared with FFT. Computations are in MATLAB [15].

The feature points from Burg plot form the input to the second employed algorithm, namely Toeplitz matrix approach. Toeplitz matrices are formed from these points to calculate their minimal eigenvalues, which produce the unique feature vector for each sample. The input to Toeplitz matrix model is the computed distances W_i for $i=1, \dots, n$ where n is the number of the considered points taken from the plot as shown in Fig. 2.

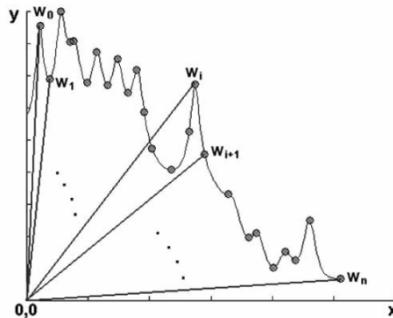


Fig. 2. Measuring the distances W_i from the origin to the feature points [15]

2.3 Classification

For classification purposes, two neural networks were built and trained in order to perform the experiments for this paper: Probabilistic Neural Network (PNN) and Radial Basis Function Neural Network (RBF). The structures of these NNs are widely described in [14], [16], [17] and [18].

Probabilistic Neural Network is one of the Bayesian statistics models. For classification, this NN bases on learning to estimate probability density functions of all of the possible classes. Then it compares the probabilities of the various classes in order to select the most appropriate one [18].

Radial Basis Function Neural Network on the other hand has a hidden layer of radial units which represent nonlinear functions. This means RBFs are efficient and easy for training and can model every shape with only one layer [18]. The results of applying RBF with Burg's method in the authors' model proved the efficiency of this type of neural networks. The success rate was increased significantly as shown in the following section.

3 Experiments and Results

The experiments were conducted using combinations of methods:

1. Burg's method of linear prediction for feature extraction and PNN for classification (Table 2).
2. Burg's method of linear prediction for feature extraction and RBF for classification (Table 3).
3. Burg's method of linear prediction followed by the calculation of minimal eigenvalues of Toeplitz matrices (TM) for feature extraction and RBF for classification (Table 4).

Burg's method was tested with many possible values for the estimation parameters prediction order (P), and the FFT size, called the length of FFT ($NFFT$) and during all of the tests the aforementioned split of the samples was used - 2 minutes of each sample for training and the rest for testing. The overall success rate is calculated as the mean of individual success rate.

Table 2. Burg with Probabilistic NN. $NFFT$ and P are parameters of Burg's method of linear prediction

$NFFT$	P	Result	$NFFT$	P	Result	$NFFT$	P	Result
64	16	69.47%	128	16	69.07%	256	16	68.50%
	20	69.60%		20	70.05%		20	69.48%
	24	70.35%		24	70.67%		24	70.43%
	28	71.72%		28	71.37%		28	71.39%
	32	72.72%		32	72.38%		32	71.39%
512	16	68.46%	1024	16	68.35%	256	24	70.43%
	20	69.21%		20	69.10%		28	71.39%
	24	70.46%		24	70.36%		32	71.39%
	28	71.15%		28	71.29%			
	32	71.75%		32	71.60%			

Table 3. Burg with Radial Basis Function NN

$NFFT$	P	Result	$NFFT$	P	Result	$NFFT$	P	Result
64	16	62.89%	128	16	64.71%	256	16	70.08%
	20	67.65%		20	70.36%		20	69.95%
	24	66.87%		24	69.98%		24	73.78%
	28	65.75%		28	52.14%		28	74.53%
	32	42.93%		32	72.43%		32	72.28%
512	16	70.76%	1024	16	67.04%	256	24	69.50%
	20	70.70%		20	65.78%		28	68.73%
	24	74.97%		24	69.50%		32	71.77%
	28	76.29%		28	68.73%			
	32	70.61%		32	71.77%			

Table 4. TM with Radial Basis Function NN

$NFFT$	P	Result	$NFFT$	P	Result	$NFFT$	P	Result
64	16	54.24%	128	16	40.72%	256	16	53.99%
	20	48.74%		20	56.73%		20	59.40%
	24	45.18%		24	62.97%		24	59.41%
	28	47.00%		28	63.52%		28	65.14%
	32	51.00%		32	58.91%		32	63.95%
512	16	52.72%	1024	16	56.84%	256	24	62.31%
	20	58.10%		20	57.83%		28	63.38%
	24	56.92%		24	62.31%		32	60.73%
	28	63.40%		28	63.38%			
	32	59.35%		32	60.73%			

4 Conclusions and Future Work

The first step of experiments was to find out whether RBF or PNN would be more efficient with Burg prediction. As the results show, the RBF is more efficient (almost 4%) in proper classification than PNN approach. Applying the sequence of

eigenvalues of Toeplitz matrices after Burg prediction without modification, however, decreased the success rate by more than 11% although in earlier works [14] it did improve the results. The difference was the larger number of samples and the nature of database. In this work, the goal of our research was to develop the system that will work well in real life environment (not in the laboratory). Thus, making tests on the samples recorded in acoustically perfect environment or with artificial noise would not meet the main aim of the research. This is because it is not difficult then to acquire over 90% proper recognition ratio with such kind of samples as authors proved with their earlier works on a laboratory-prepared database.

The main observation from the results is that for different methods and kinds of NN, different parameters of *NFFT* and *P* should be used to obtain the best recognition rate. The conclusion that may be drawn is that more experiments with the wider range of parameters may lead to an increase in the recognition rate. However, the complex results show that the recognition rate has the tendency to reach the peak with certain *NFFT* and *P* parameters and it decreases with further increase of these parameters. The results are stable and promising and lead to a conclusion that further development of the proposed algorithms may be productive and guide to a development of efficient and more accurate text-independent speaker identification system. The proposed improvements would cover, in particular, the increase in the database size and the modification of Toeplitz matrix approach to adopt the new changes in Burg-NN algorithm when considering higher number of classes.

References

1. Pruzansky, S.: Pattern-Matching Procedure for Automatic Talker Recognition. *Journal of the Acoustical Society of America* 35(3), 354–358 (1963)
2. Gish, H., Karnofsky, K., Krasner, M., Roucos, S., Schwartz, R., Wolf, J.: Investigation of text-independent speaker identification over telephone channels. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP 1985*, vol. 10, pp. 379–382 (April 1985)
3. van Vuuren, S.: Comparison of text-independent speaker recognition methods on telephone speech with acoustic mismatch. In: *Proceedings of Fourth International Conference on Spoken Language ICSLP 1996*, October 3-6, vol. 3, pp. 1788–1791 (1996)
4. Matsui, T., Furui, S.: Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMM's. *IEEE Transactions on Speech and Audio Processing* 2(3), 456–459 (1994)
5. Reynolds, D., Rose, R.: Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing* 3(1), 72–83 (1995)
6. Hou, F., Wang, B.: Text-independent speaker recognition using probabilistic SVM with GMM adjustment. In: *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering*, October 26-29, pp. 305–308 (2003)
7. Nakagawa, S., Zhang, W., Takahashi, M.: Text-independent speaker recognition by combining speaker-specific GMM with speaker adapted syllable-based HMM. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP 2004*, May 17-21, vol. 1, pp. 81–84 (2004)
8. Sant'Ana, R., Coelho, R., Alcaim, A.: Text-independent speaker recognition based on the Hurst parameter and the multidimensional fractional Brownian motion model. *IEEE Transactions on Audio, Speech, and Language Processing* 14(3), 931–940 (2006)

9. Malik, S., Afsar, F.: Wavelet transform based automatic speaker recognition. In: Proceedings of IEEE 13th International Multitopic Conference INMIC 2009, December 14-15, pp. 1–4 (2009)
10. Sen, N., Basu, T., Patil, H.: Significant improvement in the closed set text-independent speaker identification using features extracted from Nyquist filter bank. In: Proceedings of International Conference on Industrial and Information Systems ICIIS 2010, July 29-August 1, pp. 303–308 (2010)
11. Wolf, M., Park, W., Oh, J., Blowers, M.: Toward Open-Set Text-Independent Speaker Identification in Tactical Communications. In: Proceedings of IEEE Symposium on Computational Intelligence in Security and Defense Applications CISDA 2007, April 1-5, pp. 7–14 (2007)
12. Saeed, K., Nammous, M.: Heuristic Method of Arabic Speech Recognition. In: Proceedings of 7th International Conference on Digital Signal Processing and its Applications IEEE-DSPA 2005, Moscow, vol. 2, pp. 528–530 (2005)
13. Saeed, K., Szczepański, A.: A Study on Noisy Speech Recognition. In: Proceedings of International Conference on Biometrics and Kansei Engineering ICBAKE 2009, June 25-28, pp. 142–147 (2009)
14. Saeed, K., Nammous, M.: A Speech-and-Speaker Identification System: Feature Extraction, Description, and Classification of Speech-Signal Image. *IEEE Transactions on Industrial Electronics* 54(2), 887–897 (2007)
15. Saeed, K.: Toeplitz-Based Voice Verification for Human Identification - Extensive Review. *Recent Patents on Electrical Engineering* 1(3), 238–243 (2008)
16. Hill, T., Lewicki, P.: Statistics Methods and Applications. StatSoft Inc (2006)
17. Mak, M., Allen, W., Sexton, G.: Speaker identification using radial basis functions. In: Proceedings of Third International Conference on Artificial Neural Networks, May 25-27, pp. 138–142 (1993)
18. Electronic Statistics Textbook. StatSoft Inc. (accessed 2010),
<http://www.statsoft.com/textbook/>

Towards Automatic Image Annotation Supporting Document Understanding

Urszula Markowska-Kaczmar, Paweł Minda, Krzysztof Ociepa,
Dariusz Olszowy, and Roman Pawlikowski

Wroclaw University of Technology, Wroclaw, Poland

urszula.markowska-kaczmar@pwr.wroc.pl

<http://www.iis.pwr.wroc.pl/~kaczmar>

Abstract. The paper describes our research concerning image classification of types of graphics like plots, flow charts, illustrations and photos. Illustrations and photos are also classified into one of the following semantic classes: buildings, people, nature landscape, and interior. On this basis each image is annotated by its type and class. The key elements of the system – feature extraction and classification methods – are described in detail. A new classifier based on fuzzy logic was proposed. Moreover, we developed the Multi-Classifier, a hierarchical architecture encouraging the creation of hybrid classifiers tailored to the problem being solved. Experimental results of classification efficiency show that our approach is definitely worth further development.

Keywords: Image annotation, image classification, text understanding, feature extraction, machine learning, SVM, k-NN, fuzzy logic.

1 Introduction

Systems based on natural language processing become more and more popular. It is possible thanks to the rapid development of the Internet (WordNet) and computer resources. The key aspects of systems carrying a dialog in natural language are based on recognition of semantic elements, using carefully annotated attestations taken from large electronic corpora. Much research has been made concerning deep and shallow text parsing, and ontology development that provides means to select key sense elements expressing the focus of interest of a user. These techniques are very useful in text understanding, classification of text collections, and document retrieval based on its content.

However, a great deal of information about the text can be acquired from the graphical structures included in documents. They can support similar documents retrieval or, in general, particular information retrieval. Once we recognize the type of graphics in the document (for instance image, illustration, schema, diagram, graph or chart) we can reason about the content of the document. Going further, recognition of objects existing in a picture or class of pictures, like, for example, inside or outside, offers new prospects for text understanding. This is

the idea behind our project. During our research we focused first on the automatic recognition of graphics types. We then enhanced the level of specificity, considering two types of graphics: illustrations and images. These graphics are classified into classes chosen by us arbitrarily: buildings, nature landscape, people, interior. Accomplishing this task will support in our case text understanding, but it would be also helpful in similar images retrieval.

Application of image annotation is not a new approach. It forms one of the core methods of image retrieval systems, but the idea lying behind its use is in our case novel. Systems based on annotated images typically focus on searching for images in a database whose description matches a user query. These queries are usually formulated so that they contain keywords corresponding to the most important graphical or semantic aspects that the image should possess. The advantages of automating the annotation process are obvious – manual annotation is time-consuming and expensive. Much research has been done in this field (8,9).

The goal of this study was to classify images according to their type and class (we use the term category to denote either type or class of an image). To put it across, the main difficulty was to define a set of features that discriminate these types and classes of graphics. That is why we have proposed a simple system architecture for performing feature extraction and image classification.

The paper is organized as follows. In section 2, a general overview of the method is given. Section 3 contains a more detailed description of all system components, focusing on those used during experiments. In section 4 we discuss the experiments performed and the results obtained. Finally, conclusions and future research directions can be found in section 5.

2 Overview of Our Approach

As we have stated earlier, in this study our goal was to annotate graphics with one of the following types: plot, flow chart, illustration, photo. Among the last two types of images we further distinguish four classes, namely: buildings, nature landscape, people, interior. Examples of graphic types considered in our research are shown in Fig. 1. Sample images of particular classes are presented in Fig. 2 and Fig. 3 for illustrations and photos respectively.

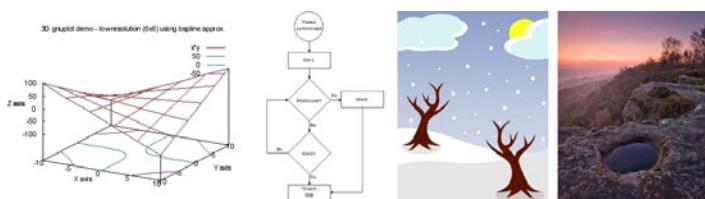


Fig. 1. Examples of image types: plot, flow chart, illustration, photo



Fig. 2. Examples of illustrations for classes: buildings, nature landscape, people, interior



Fig. 3. Examples of photos for classes: buildings, nature landscape, people, interior

The choice of the above mentioned types and classes might seem rather arbitrary, since many other can easily be conceived. We chose types of images that we believed most people would consider easily distinguishable from one another. The same applies to the classes chosen. It is important to note that in our corpus of images we collected graphics that really belong to one (and only one) of the predefined types or classes.

To achieve our goal, two elements are essential – representation of the image in form of a vector $\mathbf{v} = [v_1, v_2, \dots, v_k]$ containing values of features f_i where $i = 1, \dots, k$, and a classification method that, after system training phase, is able to assign a query image to a particular image type or class. This is reflected in the system architecture composed of two modules: the feature extraction module and the classification module.

It was difficult to know in advance which features will be responsible for discrimination of types and classes of images, and to check whether the result is independent of the classifier. The architecture of the experimental framework was designed so as to provide flexibility for feature extraction, especially the choice of the features considered in particular experiments. The feature extraction module takes as input an image. The module consists of submodules, each one responsible for extracting from the image a specific single feature f_i (e.g. the number of straight lines in the image found using the Hough transform) or related group of features (such as various histogram measures). Submodules work independently from one another, enabling a choice of a custom feature set. We expected to be able to extract global as well as local image features that would result in overall high classification rates. It should be noted though, that the categories share some graphical similarities (e.g. plots and flow charts are figures

which usually have a constant background, images in the interior and buildings classes will contain many straight lines) which makes the classification process challenging.

We have used three different classifiers during our research: k-nearest neighbours, SVM and our own method based on fuzzy approach. They will be shortly characterized in the next section. The classification module can make use of any classifier.

The modular architecture of the proposed system makes it scalable in that different or more categories than the ones proposed may be analysed. In that case, modifying the features and classifiers used should be considered as well. The possibility of changing categories, features extracted, and classifiers makes the system a truly hybrid solution suitable for research purposes.

3 Method Details

In this section we will describe the main components of the method. Because the classifiers need as input a feature vector, we will start with description of features considered that play an essential role in discrimination of classes and types of images.

3.1 Image Features

Both local and global features were extracted from images. Some of them, such as face detection and diagram detection, were 'human-inspired' – they reflect what people might seek in the images in order to classify them correctly. Others were more technical. Some were used exclusively during either type or 'class' classification. A list of feature extraction methods along with short characteristics and our observations is given below.

GLCM Texture. It makes use of the Grey-Level Co-ocurance Matrix (), a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image. It distinguishes well photos from plots.

Frame color. It calculates the number of times the color of pixels in the 1-pixel frame of the image changes. We observed that most plots and flow charts have a constant background resulting in few changes of color in the frame section. This number increases in the case of illustrations, and is usually very high for photos. It is used only during type classification.

Diagram detector. It computes measures which might be useful in detecting diagrams in flow charts: the number of standalone rectangles in the image, and the ratio of horizontal to vertical lines found in the image. It is used only during type classification.

Histogram. It calculates histogram features such as: average and median values of the histogram, level of histogram 'smoothness', number of separate color areas found by the histogram, and existence of a probable background color.

Corner detector. It works on a binarized image, finding points resembling corners. Only corners formed from crossing horizontal and vertical lines are analyzed. These are often present in plots and flow charts. It is used only during type classification.

Face detector. It uses the face detection tools provided by the OpenCV library [7]. It is used only during 'class' classification.

Dominating channel. For every pixel in the image it finds the RGB channel that has the highest value, and increments a global counter for this channel. It returns the accumulated results for each channel. It works well when separating images of interior from buildings.

Straight lines detector. It detects straight lines using the probabilistic Hough transform [1]. At preprocessing stage it converts the image to grayscale, and performs edge detection (we used the Canny edge detector [2]). It is tuned so as to find lines especially in images with buildings. It is applied only during 'class' classification.

3.2 Classifiers

The key role in the system is played by the classification module. We have tested three different base classifiers, described below. Two of them are well known, so in these cases we refer readers to the appropriate literature. The last one was designed for the purpose of image classification, and will be presented in detail.

SVM. The Support Vector Machine [11] method is one of the most popular classifiers. An implementation from [10] was used in the experiments.

k-NN. The simple k-nearest neighbors algorithm [12] with $k = 5$ was the second classifier used in the experiments. We used our own implementation.

Fuzzy Classifier. A new classifier based on fuzzy logic was developed for our research purposes. The idea behind this classifier is shown in Fig. 4 and is described below.

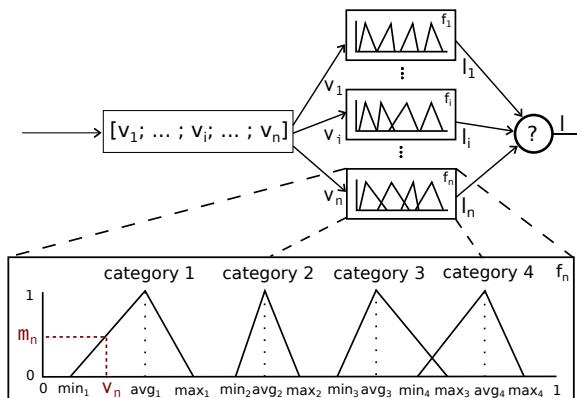


Fig. 4. The Fuzzy Classifier

In our classifier each feature value v_i ($i = 1, \dots, n$) from the feature vector is processed separately. The training phase of the classifier consists of calculating for each feature f_i characteristic points for each fuzzy set (which refer to particular categories). We assumed triangular fuzzy sets (see Fig. 4), formed by the minimal (\min), average (avg), and maximal (\max) values of a particular feature.

During classification, the degree of membership value m_c to category c of feature value v_i is calculated as follows:

$$m_c = \begin{cases} \frac{v_i - \min_c}{\text{avg}_c - \min_c} & \text{if } \min_c \leq v_i < \text{avg}_c , \\ \frac{\max_c - v_i}{\max_c - \text{avg}_c} & \text{if } \text{avg}_c \leq v_i \leq \max_c , \\ 0 & \text{otherwise .} \end{cases}$$

The highest value of m_c for v_i defines the label l_i , holding the image category of the fuzzy set, assigned to the analyzed feature f_i . Once the membership values are calculated for every feature, the feature with the highest membership value is selected, and the label assigned to this feature defines the final category that the image belongs to.

The Fuzzy Classifier is especially suited to problems in which some features have highly discriminative values for two different classes. In this case, such a feature is given precedence over all others classifying with 'less certainty'. This way the Fuzzy Classifier avoids the problem of losing classification precision due to 'overconsidering' features which are not important. Unfortunately it is not always possible to find features which separate classes so well from one another.

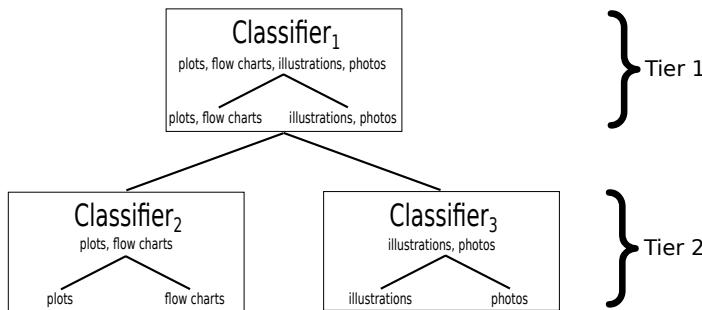


Fig. 5. The Multi-Classifier

The last classifier we have developed and tested is the Multi-Classifier (Fig. 5), a hybrid solution composed of various classifiers. The basic notion stems from the simple idea, that discriminating between two classes is easier than among four. This fits well with the fact, that our image types can be easily divided into two groups – one containing plots and flowcharts, the other illustrations and photos. Thus, the Multi-Classifier is a hierarchical 'classifier of classifiers', containing three classifiers at two different tiers. At tier one, the first classifier divides the images into two groups as described above. The second tier consists of two classifiers, each

applied to one of the groups separated in the previous step. Such an architecture allows us to use any classifier we wish at three different points, which greatly encourages the development of hybrid classifiers. This is an important factor, since certain classifiers might perform poorly in the general case of discerning among many classes, but much better when faced with a smaller set of them. We have tested the Multi-Classifier using SVM, k-NN and Fuzzy classifiers.

4 Experimental Results

The goal of the experimental study was to evaluate the efficiency of our approach. The F-score measure was chosen as the base performance metrics. Since it is based on two other measures, precision and recall, it is more informative than either of the two. We also noted the accuracy measure for reference purposes.

The image corpus used in our research was composed from images from various freely available sources on the internet (among others [3], [4]), manually annotated by us. It contains 1360 graphics: 239 plots, 240 flow charts, 441 illustrations, and 440 photos. We tried to collect graphics which were not too similar, yet retained the crucial aspects of the types and classes analyzed.

Table 1 shows the results of type-classification obtained by Multi-Classifiers consisting solely of one of the three base classifiers. **Table 2** contains results of both type and 'class' classification for base classifiers. In both cases the tables contain average values computed from a ten-fold crossvalidation. The best results were obtained by the base k-NN classifier, closely followed by SVM. Although their Multi counterparts performed worse, the differences were not extreme.

Table 1. Multi-Classifier type-classification results

Classifier	F-score		Accuracy	
	Train data	Test data	Train data	Test data
Multi FFC	63	61	83	91
Multi k-NN	83	78	93	95
Multi SVM	77	77	90	95

Table 2. Classification results for types, and classes of illustrations and photos

Classifier	Kind	F-score		Accuracy	
		Train data	Test data	Train data	Test data
FFC	Types	46	44	73	86
	Illustrations	36	46	68	83
	Photos	35	43	68	83
k-NN	Types	85	80	93	95
	Illustrations	72	68	86	90
	Photos	74	71	86	91
SVM	Types	76	78	90	95
	Illustrations	55	66	77	89
	Photos	60	70	79	90

The Fuzzy Classifier fared considerably better when coupled with the Multi-Classifier architecture than when used separately (a 15% difference in F-score measure). The experiments have shown though, that the Fuzzy Classifier would need further improvement in order to be useful in our classification problems.

5 Conclusion and Future Research

Although the initial experiments evaluating the method are promising, to fully evaluate its efficiency a greater corpus is needed. It seems that the feature extractors developed need more tuning and testing. We also want to compare results obtained by our Fuzzy Classifier with other methods, such as fuzzy k-NN ([13]).

In our opinion the Multi-Classifier holds a hidden potential for raising the overall classification rates when used with appropriate inner classifiers and feature vectors.

The experiments conducted have shown that it is possible to classify images according to their type with considerable success. The on average 80% correctly classified images using k-NN algorithm can be considered a good starting point for future development.

Acknowledgement

This work was supported by the Innovative Economy Programme project POIG.01.01.02-14-013/09-00.

References

1. Duda, R.O., Hart, P.E.: Use of Hough transform to detect lines and curves in pictures. *Commun. ACM* 15, 11–15 (1972)
2. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Match. Intell.* 8(6), 679–698 (1986)
3. MIT Indoor Scene Recognition Database, <http://web.mit.edu>
4. ETH Zurich Computer Vision Laboratory, <http://www.vision.ee.ethz.ch>
5. Haralick, R.M., Dinstein, I., Shanmugam, K.: Textural features for image classification. *IEEE Trans. Syst. Man Cybernet.* 3, 610–621 (1973)
6. Lu, X., Kataria, S., Brouwer, W.J., Wang, J.Z., Prasenjit, C., Giles, L.: Automated analysis of images in documents for intelligent document search. *IJDAR* 12(2) (2009)
7. OpenCV library Wiki, <http://opencv.willowgarage.com/wiki>Welcome>
8. ALIPR - Automatic Photo Tagging and Visual Image Search, <http://alipr.com>
9. Malathi, G., Shanthi, V.: Histogram Based Classification of Ultrasound Images of Placenta. *IJCA* 1(16), 0975-8887 (2010)
10. LIBSVM - A Library for SVM, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
11. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2(2), 121–167 (1998)
12. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1), 21–27 (1967)
13. Chen, L., Yap, K.-H.: A fuzzy K-nearest-neighbor algorithm to blind image deconvolution. *IEEE Trans. Syst. Man Cybernet* 3, 2049–2054 (2003)

A Computational Assessment of a Blood Vessel's Compliance: A Procedure Based on Computed Tomography Coronary Angiography

Piotr Porwik¹, Maciej Sosnowski²,
Tomasz Wesolowski¹, and Krzysztof Wrobel¹

¹ University of Silesia, Institute of Computer Science,
ul. Bedzinska 39, 41-200 Sosnowiec

{piotr.porwik,tomasz.wesolowski,krzysztof.wrobel}@us.edu.pl

<http://zsk.tech.us.edu.pl>

² Medical University of Silesia, Unit of the Noninvasive Cardiovascular Diagnostics;
3rd Division of Cardiology, Upper Silesian Heart Centre, Katowice, Poland

Abstract. Cardiovascular mortality remains a leading health and social problem in many countries throughout the world. Therefore, it is obvious that current preventive measures include early detection of atherosclerosis process. Computed tomography (CT) is one of imaging modalities allowing for noninvasive detection of atherosclerotic lesion within coronary arteries in subjects with accumulation of risk factors (smoking, high lipids, hypertension, etc.). Functional analysis of coronary arteries is usually approached by means of invasive procedures. We aimed at finding solution for evaluation of another kind of functional analysis of coronary arteries, namely vessel's wall compliance by means of CT coronary angiography. On the basis of the CT images of the vessels over entire cardiac cycles, the internal area of the blood vessel is measured and its changes during various phases of heartbeat (systole, diastole) are calculated. If the vessel wall has been changed by atherosclerotic plaque, either calcified or non-calcified, then its compliance will be reduced due to its stiffness. Calculation of coronary artery compliance requires a series of measurements, which is unreliable and impractical for doing manually. The overall methodology proposed in this paper assists in the preparation of a medical diagnosis.

Keywords: Coronary artery disease, Arteriosclerosis, Image processing.

1 Introduction

Hybrid intelligent systems are today very popular due to their capabilities in handling many real world complex problems. They have the opportunity to use both, knowledge and raw data to solve problems in many multidisciplinary researches - for example in analyzing of medical images, and so on [2]. It is also an area of the artificial intelligence, therefore we have an opportunity to present and discuss the theoretical advances and real-world applications - computed tomography imaging. Heart attack and stroke are two main reasons of cardiovascular

mortality - an epidemic in many countries all over the world. The common reason of these two severe complications is atherosclerosis. The process of atherosclerosis is related with development of so-called plaques within arteries' walls through accumulation of lipids, blood cells, calcium, vascular smooth muscle and fibrous cells. It leads to remodeling of vessels which is characterized by thickening and stiffening of their walls. If the process of atherosclerosis is located within coronary arteries then a disease known as coronary artery disease (CAD) develops.

Recent advances in non-invasive imaging of coronary arteries using multi-detector computed tomography (MDCT) allow for early detection of atherosclerotic plaques, either by means of coronary artery calcium scoring (CACS) or by means of CT coronary angiography (CTCA) [6,7]. The CACS is the simplest *in vivo* method in which a series of non-contrast (native) scans is analyzed semi-automatically. Detection of calcium lesions provides an unambiguous proof for the presence of coronary atherosclerosis. If the CACS is high, the probability of the presence of significant coronary lumen stenosis is also high. Additionally, it is possible to evaluate extension of calcified lesions, as CACS can be determined separately for each main coronary artery as well as for their specific parts (segments). Detection of CACS within all main arteries indicates more advanced CAD. The CTCA requires that similar scans are obtained after injection of contrast agents. Cardiac images ranging from aortic root to the cardiac apex are acquired during a single breath-hold of 10 second. The CTCA enables for both coronary artery lumen and walls to be visualized. A lumen stenosis that limits coronary blood flow can be quantified and atherosclerotic plaques, both calcified and non-calcified, can be detected and characterized.

At early stages of atherosclerosis, vascular lesions, being small and non-calcified, cannot be easily detected by means of CTCA, taking into account its limited resolution. However, these small atherosclerotic deposits influence upon arterial wall function through its thickening and vessel enlargement (so-called positive remodeling). As arterial lumen, estimated on its cross-sections, is changeable over cardiac cycle due to variable blood flow within coronaries, measurements of vessel's cross-sections might help to evidence an abnormal vessel's wall function, namely a decreased compliance [1,5]. A procedure based on this data and approach will be presented in this paper.

2 Data Acquisition

During CT procedures only raw-data are recorded. It is from this raw-data that images, for display on a work station, are constructed. By using specialized graphics programs, a radiologist can display images built from this data on their monitors in any of a range of scales, in different views, three-dimensional projections and reconstructions. These projections produce the "tomogram": a two-dimensional image of a section, or slice, through a three-dimensional object.

The cardiac CT scans present images similar to Fig. 1. The radiologist indicates the areas and sections of blood vessels where measurement should be conducted. Fig. 1a presents a CT image upon which a blood vessel examination

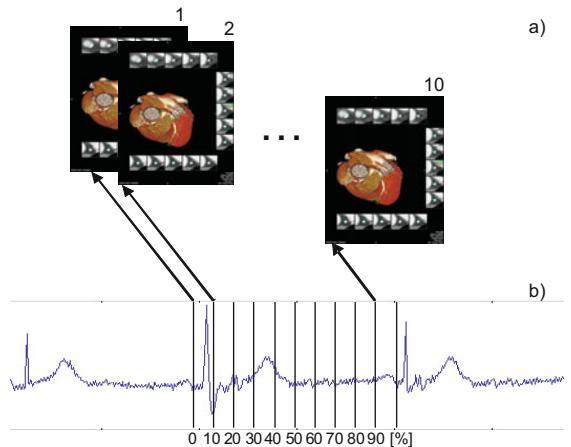


Fig. 1. CT imaging (a), initialized by the ECG strobing signal (b)

has been performed. In this picture, different cross-sections along the vessel can be studied. Each cross-section is surrounded by a frame, which has been automatically added by the work-station software. Inside the frame, the shape of the cross-section is then estimated. This examination was performed by a Toshiba Aquilion 64 CT scanner and by the Vitrea2 software (Vital Images Inc.) application at the Unit of the Noninvasive Cardiovascular Diagnostic at the Medical University of Silesia. If the reference points for the measurements have been chosen by the physician, then evaluations will be automatically conducted at these points and at points adjacent to each point, both in front of the point and behind it. Here seven locations before and seven locations after the reference point are studied: fifteen blood vessel cross-sections will be investigated. The cross-sections can be separated by a constant distance; here the step size is 0.5 mm. Fig. 1b shows the patient's ECG pulse as recorded during the examination. The appropriate phases of the patient's pulse trigger the CT imaging. In practice, ten phases of the ECG, separated by a 10% step, are used to construct the images. Hence, the 10 images I_1, I_2, \dots, I_{10} can be successively displayed, as depicted in Fig. 1a. As was mentioned, inside every individual image I_i , fifteen sub-images are included: $I_i \supset \{I_i^1, I_i^2, \dots, I_i^{15}\}$; each sub-image $I_i^j, i = 1, \dots, 10, j = 1, \dots, 15$ includes one cross-section of the analyzed vessel. In total there are 150 sub-images. Thus, one cross-section can be observed over ten heartbeat phases. For this paper, due to technical restrictions, the images and the places on these images at which measurements are conducted are too small to be properly displayed. Therefore, in Fig. 2, an exemplary coronary vessel model has been used. In this model a fragment of the vessel displaying pathological changes (having an inner wall layer with calcification) is included, together with a normal fragment, illustrating the positions at which contractibility studies of the vessel are performed. In the same figure (Fig. 2) the reference point (indicator 0) is presented, along with fourteen additional points $-1, \dots, -7, +1, \dots, +7$ at each of which the vessel's cross-section will be displayed and studied.

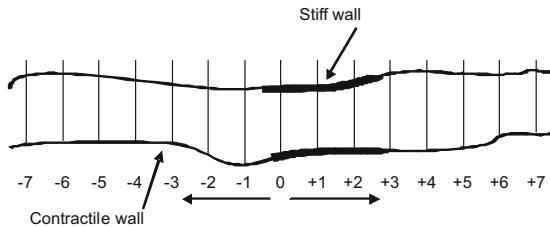


Fig. 2. Exemplary model of a coronary vessel, with calcified pathological changes (stiff vessel walls), and a normal, contractile vessel's wall

3 Image Processing

During the first stage, the appropriate image is constructed, as presented in Fig. 3a. This is the image I_i . On the left side (Fig. 3a), the original CT image is presented, using which, the appropriate coronary vessel, as indicated inside the white circle, will be examined by a radiologist. On the same CT image, the 15 frame-sub-images I_i^j are shown. Each sub-image I_i^j displays a cross-section of the examined vessel. These cross-sections are automatically selected by the tomograph's controlling software, are displayed on the radiologist's computer screen and can then be written to any of a range of graphic formats such as JPG, PNG and many others. These files can each be separately processed by image processing techniques. In the next stage, the source image is converted into a binary representation [3], as depicted in Fig. 3b. The CT image is a color image, with specific set of colors; image I is presented in an RGB color space. The binary representation is established as indicated in the following pseudocode. Let the current position of the pixel on the image will be denoted by $I(x, y)$ then:

- binary rule for frame detection:

```

if (R=178,G=178,B=178) or (R=100,G=149,B=237) or
    (R=0,G=205,B=0) or (R=240,G=128,B=128)
then
    the pixel I(x,y) is changed to black (R=G=B=0)
otherwise
    the pixel I(x,y) is changed to white (R=G=B=255)
endif

```

- binary rule for cross-section detection:

```

if (R>90) or (G<140 or G>240) or (B<80 or B>240)
then
    the pixel I(x,y) is changed to white (R=G=B=255)
otherwise
    the pixel I(x,y) is changed to black (R=G=B=0)
endif

```

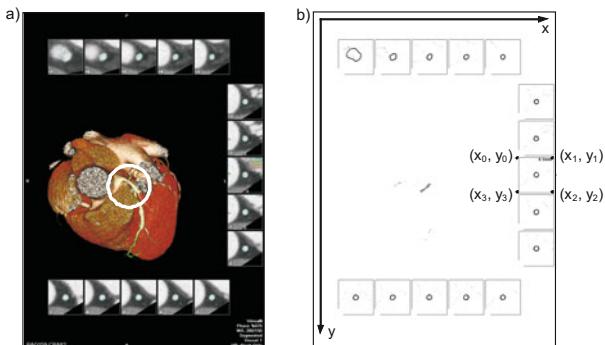


Fig. 3. The source CT image (a), and the same image after conversion to binary (b)

The process of conversion into a binary image permits the extraction of both the frame that surrounds each sub-image, as well as the shapes of the cross-sections inside each frame. After conversion into a binary representation, in the central region of the image (Fig. 3b), some artifacts can be observed. Similar noise inside the frames is also present. It should be noted that in the cross-sections, especially the shapes of these cross-sections, are clearly visible and properly detected, so that these artifacts of the image can be correctly extracted. In the subsequent step, the coordinates of the four vertexes of every frame are established. The vertex coordinates are calculated by the Hough Transform (HT).

In this paper the Hough Transform for straight lines was applied [4].

If each frame's coordinates are numbered in a clockwise fashion, then the first frame's centre of gravity will have the coordinates $(x'_1, y'_1) = (\frac{x_1-x_0}{2}, \frac{y_3-y_0}{2})$, and similarly for the points of each following frame. Thus the (x'_j, y'_j) , $j = 1, \dots, 15$ points are fixed. Now each cross-section is centered to its point (x'_j, y'_j) . The reference frame, selected by the physician, includes a cross-sectional area (in mm^2), automatically computed by the tomograph's controlling software. Obviously, the CT images are taken with a uniform resolution. Thus there are a fixed number of pixels inside each cross-section's perimeter. In each successive step, the number of pixels inside the remaining cross-sections is assigned. Hence, by a simple computation the remaining cross-sectional areas, mm^2 , can be determined. Currently, a diagnosis of atherosclerosis disease can be made when 1mm^2 or more of the vessel's cross-sectional area is occupied by atherosclerotic plaque.

4 The Results Obtained

The determination of the contractility of coronary vessels has been practically demonstrated for sets of images derived from three patients. Each image consisted of a set of 15 sub-images, as shown in Fig. 3a. These images were processed by the image processing procedures described in the previous paragraphs. The two outermost cross-sections (+7, +6 and -7, -6) were not analyzed because, during the observation, the physician could not accurately identify the reference

Table 1. Vessel cross-sectional changes across the ten phases of the cardiac cycle; one patient

T[%]	The vessel's cross-sections [mm ²]										
	5	4	3	2	1	0	-1	-2	-3	-4	-5
0	7,93	8,06	7,90	8,43	8,50	8,40	8,94	8,94	9,17	9,17	10,08
10	8,63	9,39	10,71	12,21	14,62	15,90	17,69	16,81	13,78	8,99	8,08
20	9,90	9,34	9,55	9,21	9,17	9,90	10,63	10,49	10,70	10,46	12,12
30	8,93	8,79	9,55	9,17	9,31	10,00	10,83	10,38	11,17	11,14	12,07
40	6,72	6,93	7,28	8,34	9,97	11,00	12,14	12,48	13,59	14,59	14,93
50	7,31	6,89	5,88	6,48	6,71	6,80	6,58	7,18	7,78	7,84	9,90
60	16,57	12,23	9,83	8,17	7,70	7,50	7,73	7,87	8,07	8,30	8,90
70	10,87	8,84	7,65	7,62	8,09	8,50	8,74	9,08	8,60	9,18	9,01
80	7,57	7,11	6,87	7,57	8,20	8,60	8,77	8,97	8,90	9,30	9,36
90	7,19	7,16	7,09	8,10	8,46	9,00	8,80	9,03	9,03	9,33	9,27
Variance	7,59	2,46	2,22	2,07	4,15	5,81	8,73	6,93	4,24	3,31	3,86

cross-section, the 0 point. Additionally, the extreme points can be closer to vessel branching points. Hence, only eleven of the vessel cross-sections have been examined. The vessel cross-section changes across the ten phases (T%) of the cardiac cycle have been listed (see Table 1). Table 1 includes the complete results from the measurements made on one patient. The vessel's contractility was measured by calculating the variance of these measurements. This measure allows for an evaluation of the dynamics of the changes in the vessel's wall. The last row of Table 1 lists the variance in the vessel's cross-sectional areas. In the table it can be seen that the greatest contractility of the vessel is in cross-section number -1, and the least contractility is in cross-section number 2. Where the dynamics of the vessel wall are small pathological changes can occur, especially changes that anticipate atherosclerotic plaque. The results gathered in Table 1 can be presented more conveniently, that is, graphically. See Fig. 4. The similar cross-section graphic projection over the cardiac cycle can be also simply displayed. From the data displayed in Fig. 4 it can be noticed that the vessel has good contractility during all phases of the cardiac cycle so, with high degree

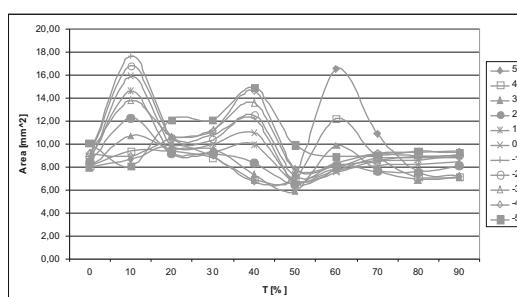
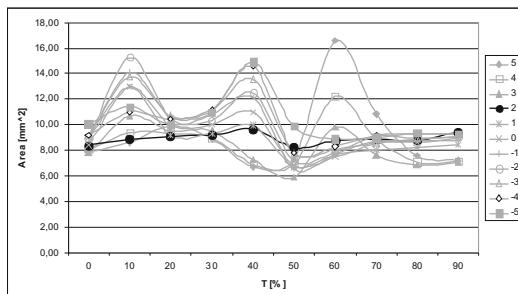
**Fig. 4.** Vessel cross-sectional areas, by phase of the cardiac cycle

Table 2. Vessel cross-sectional changes across the ten phases of the cardiac cycle; one patient

Number of cross-section											
5	4	3	2	1	0	-1	-2	-3	-4	-5	
Variance	7,59	2,48	2,22	0,18	2,57	2,89	4,45	5,08	4,24	3,34	2,90

of probability, atherosclerosis disease is not present in the vessel examined. For comparison, in Table 2, data is presented from a patient that has been diagnosed as having atherosclerotic plaque. From Table 2 it can be clearly seen that cross-section number 2 has a very small contractility, indicating a risk of disease. As with the previous example, the full data can be presented graphically (Fig. 5). In Fig. 5 it can be clearly seen that the coronary vessel's cross-section number 2 is almost constant throughout the cardiac cycle. This location is, probably, already calcified as the vessel wall has become stiffened. If inner layer of the artery is calcified, but pathological changes are still small, then these pathologies can not be imaged by normal CT procedures. However, pathologies of this type can be observed by the method described in this paper. This approach can also be used for automatic cross-sectional area computation in cases when pathological changes inside vessel are already well visible on CT images.

**Fig. 5.** Vessel cross-sectional areas across the cardiac cycle, from a patient with atherosclerotic plaque

5 Conclusions

The method proposed in this paper allows for data to be automatically read from CT images. Some additional information can also be extracted from these images. This additional information has been credibly and correctly interpreted, and the interpretation given matched the diagnosis made by a physician during the patient's examination. The CT dataset derive from the Medical University of Silesia archives and from current medical examinations. All these images are owned by the Medical Center. During investigations 30 CT images have been transformed and then processed, according to the presented in this paper method.

This procedure as our software is used by physicians during CT image analysis. The procedure detailed herein was accomplished with a graphical C# program tool that runs on all tomography workstations. Thus, using this new technique, CT images can be also studied at medical centers at which computed tomography is not present. This is possible as, after their examination, the patient can obtain their data and medical documentation on CD media. It should be emphasized here that this method permits the identification of pathological changes when these changes are not yet visible on CT images.

References

1. Chien, C., Feng, Y.F., Arora, R.R.: Advances in computed tomography-based evaluation of coronary arteries: a review of coronary artery imaging with multidetector spiral computed tomography. *Rev. Cardiovasc Med.* 8, 53–60 (2007)
2. Derrac, J., Garca, S., Herrera, F.: A first study on the use of coevolutionary algorithms for instance and feature selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
3. Dougherty, G.: Digital Image Processing for Medical Applications. Cambridge University Press, Cambridge (2009)
4. Duda, R.O., Hart, P.E.: Use of Hough transformation to detect lines and curve in pictures. *Comm. ACM* 15, 11–15 (1972)
5. Gola, A., Pysz, P., Szymanski, L., Sosnowski, M.: 64-slice MDCT coronary artery functional assessment in zero calcium score patients a novel parameter emerging. *Eur. Radiol.* 18(supl. 3)(C25), 86 (2008)
6. Schroeder, S., Kopp, A.F., Baumbach, A., et al.: Noninvasive detection and evaluation of atherosclerotic coronary plaques with multislice computed tomography. *J. Am. Coll. Cardiol.* 37, 1430–1435 (2001)
7. Walecki, J., Sosnowski, M., Zawadzki, M.: Tomografia komputerowa serca. In: Szczeklik, A., Tendera, M. (eds.) Medycyna Praktyczna, pp. 130–142. Kardiologia (2009) (in polish)

Visual System for Drivers' Eye Recognition

Bogusław Cyganek and Sławomir Gruszczyński

AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@agh.edu.pl

Abstract. The paper presents a visual system for human eye recognition for control of driver's fatigue. Modules of the system are organized into a cascade. Processing starts with skin segmentation, followed by the compact skin regions selection obtained with the adaptive window growing technique. Next step consists in selection of candidate regions based on the proposed eye model. The final eye/non-eye classification is done with a classifier build upon the higher order singular value decomposition of the deformable pattern tensor.

1 Introduction

Car drivers' fatigue, sleepiness or inattention are cause of many traffic accidents. Thanks to recent developments in computer hardware as well as in hybrid artificial intelligence methods [1][3][7][23], computers can be easily used to monitor drivers' conditions with potential of alerting if dangerous situations are detected [17]. The key module of such systems is reliable detection of driver's eyes. However, the existing solutions still lack sufficient accuracy or speed to be mounted in moving vehicles.

Detection of human faces and eyes belongs to the most fundamental tasks of computer vision [8][21]. Briefly, eye detection can be divided into two groups of active and passive methods [22]. In the first one a special lighting, such as near IR, is assumed [24]. However, this requires custom hardware setups. On the other hand, passive methods operate in the same way as the human visual system, i.e. only the natural illumination spectrum is assumed. These can be further partitioned into the methods which use color or monochrome images, respectively [2]. Reported eye recognition methods rely on detection of some characteristic eye features. These, in turn, can be obtained with template matching [10], projections [25], Hough transform [17][25], gradients [12] or wavelets [16], to name a few. Further information and references on the subject can be found e.g. in the papers [22] [24].

In this paper a visual system for eyes recognition in moving vehicles is proposed, which constitutes the front-end of a driver monitoring system. The main assumptions are high accuracy, fast operation, as well as simple implementation and low cost. The system is organized as a cascade of classifiers. Processing starts with the skin segmentation. This stage is followed by the skin region detection obtained with the adaptively window growing method (AWG) [4]. Then, eye candidate regions are selected thanks to the proposed eye model. Finally, eyes are verified by the classifier operating with the higher order singular value decomposition (HO-SVD) of the tensor of geometrically deformed images of real eye prototypes [15].

It is also assumed that the system can operate in day or night conditions utilizing daylight or near IR lighting illumination, respectively. However, in this paper we address only the first approach. For this purpose the RGB color images are considered as an input to the presented system. This allows efficient skin segmentation in daylight conditions. Such approach increases robustness of the system and greatly reduces costs. Nevertheless, all modules - except the skin segmentation - can operate regardless of the used spectra. In further work we plan to join solutions pertaining the two spectral domains, i.e. daylight and near IR.

2 System Overview

Fig. 1 depicts architecture of the proposed system. It is organized as a cascade of specialized modules, each refining output of its predecessor. Such architecture showed robustness in many classification tasks [13][11][19].

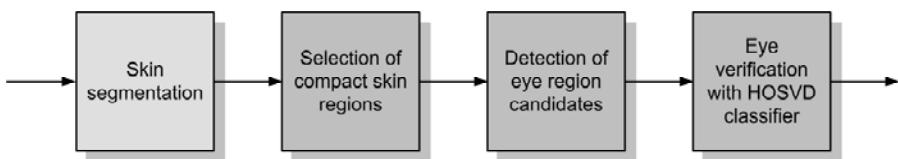


Fig. 1. Cascade structure of the eye recognition system

Skin segmentation follows the method proposed by Peer *et al.* [18]. However, an original crisp functions were converted into the following fuzzy *IF...THEN* rules to increase robustness. The advantage of this approach is high recall factor and very fast operation. On its output a binary map of skin/non-skin labeled pixels is obtained.

Fuzzy rules for sun lighting conditions

“Range of skin color components in daily conditions found in experiments”

R₁: **IF** $R > 95$ AND $G > 40$ AND $B > 20$ **THEN** $T_0 = \text{high}$;

“Sufficient separation of the RGB components; Elimination of grey areas”

R₂: **IF** $\max(R, G, B) - \min(R, G, B) > 15$ **THEN** $T_1 = \text{high}$;

“ R, G should not be close together”

R₂: **IF** $|R-G| > 15$ **THEN** $T_2 = \text{high}$;

“ R must be the greatest component”

R₃: **IF** $R > G$ AND $R > B$ **THEN** $T_3 = \text{high}$;

Fuzzy rules for artificial lighting conditions

“Skin color values for illumination”

R₄: **IF** $R > 220$ AND $G > 210$ AND $B > 170$ **THEN** $T_4 = \text{high}$;

“ R and G components should be close enough”

R₅: **IF** $|R-G| \leq 15$ **THEN** $T_5 = \text{high}$;

“ B component has to be the smallest one”

R₆: **IF** $B < R$ AND $B < G$ **THEN** $T_6 = \text{high}$;

The combined (aggregated) fuzzy rule for human skin detection

“Fuzzy rule for human skin color detection in RGB”

R_{HS}: **IF** T₀₋₃ are high **OR** T₄₋₆ are high **THEN** H=high; (1)

The other advantage of the fuzzy formulation (1) over its crisp version given by Peer *et al.* is that the influence of each particular rule can be controlled separately. Result of each relation is a fuzzy measure. These are then combined with the Mamdani rule. In result, the labeled skin segmentation map is obtained.

Next stage does grouping of pixels from the skin segmentation map into compact regions. This is done with the AWG method, details of which are presented in [4].

Then candidates for eye regions are selected from the compact skin areas based on the proposed eye model, discussed in the next section. Finally, this way refined eye region candidates are fed to the classifier operating with the HO-SVD of the tensor formed from the geometrically deformed eye prototypes. The prototypes for this classifier were gathered from real human face examples.

3 Detection of Eye Region Candidates

The AWG algorithm starts from a single skin point. Then, around such a point a rectangular window is extended in all eight directions, as long as there are sufficient number of skin pixels acquired in each expansion step [4].

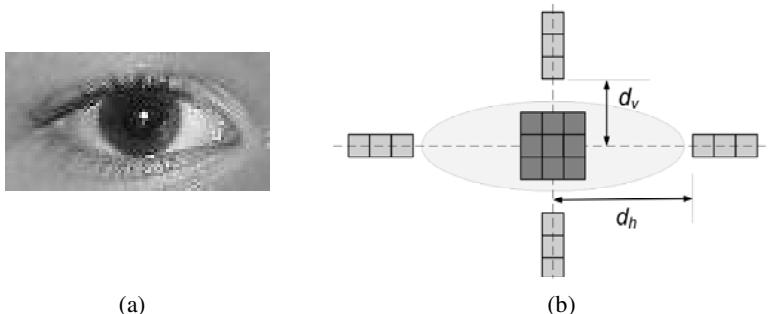


Fig. 2. An example of a human eye (a). Eye model for detection of eye region candidates in the skin segmented image (b).

Candidates for eye regions are searched in the compact skin regions obtained with the AWG method. For this purpose a model has been created (Fig. 2b) based on observations of human eyes (Fig. 2a). The central part of the model corresponds to an image of an iris of the eye. An assumption is that in real conditions an iris is enclosed by a larger skin region. However, this can be positioned in some distance d_H and d_V , due to images of white of the eye, as well as to the eyebrow. Because of this and to avoid time consuming template matching, only four lines of pixels are sampled and matched against skin color. Each match is assessed based on fuzzy rules which allow proper answer on different face positions and lighting conditions. For example, it can

happen that due to the viewpoint, an eye is seen only at the border of the face. In such a situation left or right horizontal line in the model (Fig. 2b), which corresponds to the skin region, will not match. Such situation are dealt well by the fuzzy rules. This way found eye candidate regions are finally fed to the HO-SVD classifier, described next.

4 Eye Recognition in the Deformable Pattern Space

Final eye verification is done with the classifier build upon decomposition of a tensor containing the prototype eye prototype patterns as well as their geometrically deformed versions. Such a classifier showed high accuracy in the system for road signs recognition [5]. In the presented system the patterns are real face regions, such as eyes, cheeks, etc. The prototype tensor is then decomposed with the HO-SVD decomposition [14][15], which allows a P -dimensional tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \dots \times N_m \times \dots \times N_n \times \dots \times N_p}$ to be represented in the following form

$$\mathcal{T} = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \dots \times_p \mathbf{S}_p . \quad (2)$$

In the above, \mathbf{S}_k denote unitary *mode matrices*, each of dimension $N_k \times N_k$. $\mathcal{Z} \in \Re^{N_1 \times N_2 \times \dots \times N_m \times \dots \times N_n \times \dots \times N_p}$ is a *core tensor* which is of the same dimensions as \mathcal{T} .

Now, for each mode matrix \mathbf{S}_i and using the commutative properties of the k -mode tensor-matrix multiplication, equation (2) can be rewritten as

$$\mathcal{T} = \sum_{h=1}^{N_p} \mathcal{T}_h \times_p \mathbf{s}_P^h , \quad (3)$$

where

$$\mathcal{T}_h = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \dots \times_{p-1} \mathbf{S}_{p-1} \quad (4)$$

denotes the basis tensors, while \mathbf{s}_P^h are columns of the unitary matrix \mathbf{S}_P . If \mathcal{T} is three-dimensional, then \mathcal{T}_h in (3) becomes two-dimensional (i.e. a matrix). Moreover, thanks to the orthogonality properties of the core tensor \mathcal{Z} in (4), \mathcal{T}_h are also orthogonal. Thus, \mathcal{T}_h can form the base of the new pattern space. In this framework, pattern recognition can be stated as computation of a distance of a given test pattern \mathbf{P}_x to its projections in the space spanned by the set of the base tensors \mathcal{T}_h in (3). This can be expressed as the following minimization problem

$$\min_{i, c_h^i} \left\| \underbrace{\mathbf{P}_x - \sum_{h=1}^N c_h^i \mathcal{T}_h^i}_{Q_i} \right\|^2 . \quad (5)$$

In the above equation c_h^i are unknown coordinates of \mathbf{P}_x in the space spanned by \mathcal{T}_h^i , and $N \leq N_p$ denotes a number of chosen dominating components.

For each pattern i the optimization problem (5) can be solved by first expressing the squared norm in terms of the inner product, and then computing the set of derivatives in respect to each c_h . In effect, for each pattern i , and for *a priori* chosen number of bases $N \leq N_p$, the following residual value ρ_i denotes a degree of similarity of a pattern \mathbf{P}_x to its projection into the space spanned by \mathcal{T}_h^i

$$\rho_i = \left\| \mathbf{P}_x - \sum_{h=1}^N \frac{\langle \mathcal{T}_h^i, \mathbf{P}_x \rangle}{\langle \mathcal{T}_h^i, \mathcal{T}_h^i \rangle} \mathcal{T}_h^i \right\|^2. \quad (6)$$

Finally, without loss of generality it can be assumed that \mathcal{T}_h^i and \mathbf{P}_x are normalized (hat notation). Then, to minimize (5) the following value has to be maximized

$$\hat{\rho}_i = \sum_{h=1}^H \langle \hat{\mathcal{T}}_h^i, \hat{\mathbf{P}}_x \rangle^2. \quad (7)$$

In operation mode the system returns a class i for which the corresponding $\hat{\rho}_i$ in (7) is the largest. To cope with the problem of outliers, i.e. patterns for which the system was not trained for, a minimal acceptance threshold was set on $\hat{\rho}_i$ (in our experiments it is 0.8). The number of components N in (7) was chosen from 5 to 25. Higher N allows better accuracy, at an expense of computation time. However, N cannot exceed N_p , which equals number of deformed patterns in tensor \mathcal{T} [5].

5 Experimental Results

The system was implemented in C++ and run on the computer with the 8 GB RAM and Pentium Core Q 820 @ 1.73GHz. The only one classifier that needs to be trained is the HO-SVD classifier. For this purpose a data base was created (Fig. 3) with examples of some common face parts, such as eyes and cheeks, obtained in different expressions and lighting conditions. These are then geometrically transformed, as described in [5]. Such tensor space is decomposed with the HO-SVD. In result, the base tensors \mathcal{T}_h^i are computed, which are required during system operation.



Fig. 3. A training database with the eye and not-eye patterns expected in the input images

Fig. 4 depicts first four tensors \mathcal{T}_h and the corresponding core tensors \mathcal{Z}_n for an eye training pattern from the data base in Fig. 3. It is visible that the "energy", defined as a norm of subtensors \mathcal{Z}_n , spreads out from the left top corner of the images [14].

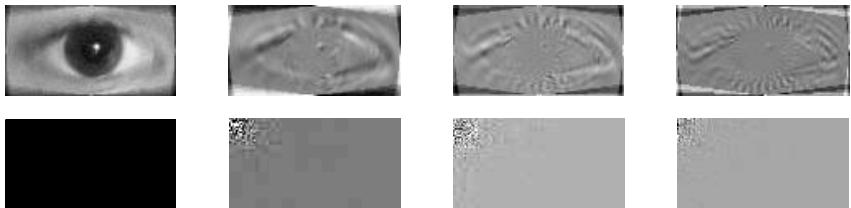


Fig. 4. First four tensors \mathcal{T}_h of an eye (first row). Corresponding core tensors \mathcal{Z}_n (second)

During operation the HO-SVD classifier tells eye regions from all the other, such as cheeks, mouths, etc. Fig. 5 depicts results of operation of our method on three selected frames showing a driver (the first column). The second column of Fig. 5 contains the skin segmented areas (in white). Third column shows compact skin regions detected with the AWG method. Finally, fourth column of Fig. 5 depicts identified eye regions. Eyes visible in the two first frames in Fig. 5 were correctly identified and placed. On the other hand, for the last frame (a third row in Fig. 5) system correctly recognized and reported closed eyes, i.e. not visible irides.

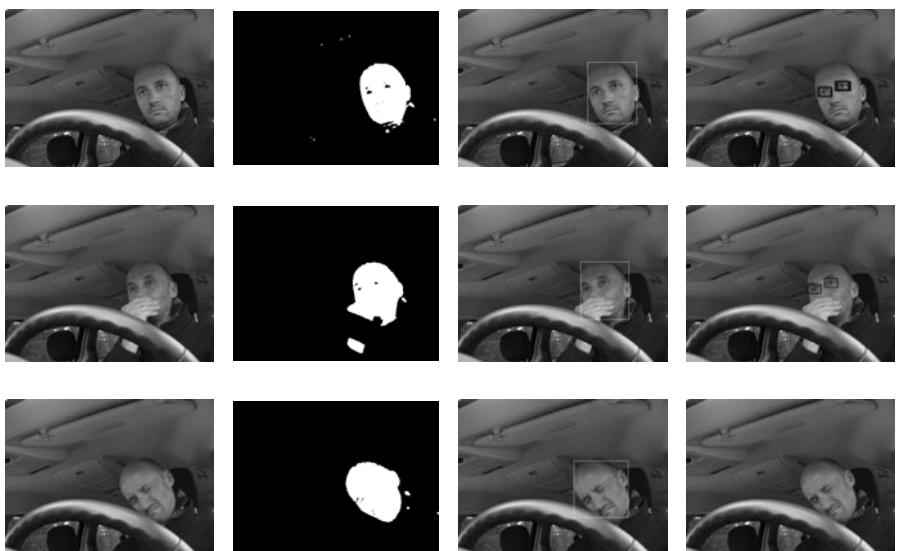


Fig. 5. Results of eye recognition for three frames (top-down) taken with the Olympus camera mounted in a car. From left to right: 640x480 RGB frames, skin binarization, human skin regions after the adaptive window growing, recognized eyes from the eye candidates.

Experiments were conducted on a database of selected test images containing persons with well visible faces in sufficient lighting conditions (i.e. daily or artificial light). To test accuracy of the method it was compared with answers of a human operator. The resulting average recall parameter is 95% and precision 98%. This compares favorably with the results reported e.g. in [24][2]. Examination of the misclassified cases reveals that problems are usually due to wrong initial skin segmentation. There are other methods which offer more precise segmentation, although at a cost of higher computational complexity [6][9][26].

The system offers an average execution time in order of 100-120 ms for a 640x480 RGB image, which to the best of our knowledge outperforms other software implementations. Finally, the other advantage of the proposed system comes from the relatively simple algorithms. These can be easily ported to hardware platforms, such as smart cameras.

6 Conclusions

In this paper a system for drivers' eye recognition from color RGB images is proposed. The method operates as a cascade of classifiers, starting from skin segmentation, followed by the compact skin region extraction with an adaptively window growing method. Then the eye region candidates are selected based on the proposed eye model. Finally, eye validation is done with a classifier based on the HO-SVD decomposition of the tensor of geometrically deformed eye prototypes. The latter were obtained from real face examples. The method shows high accuracy and speed of operation in software implementation. Moreover, the used algorithms can be easily ported to hardware platforms. For further work we plan to make the system operating in night conditions with the near IR lighting, as well as to augment it with a module analyzing time intervals of opened/closed eyes to control state of a driver. Finally, an embedding into a smart camera module is envisioned.

Acknowledgement

The work was financially supported by the National Center for Research and Development under Lider Program, contract no. LIDER/16/66/L-1/09/NCBiR/2010.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. *Neurocomputing* 72(13-15), 2729–2730 (2009)
2. Chiang, C.-C., Tai, W.-K., Yang, M.-T., Huang, Y.-T., Huang, C.-J.: A novel method for detecting lips, eyes and faces in real time. *Real-Time Imaging* 9, 277–287 (2003)
3. Corchado, E., Abraham, A., Ponce, A.C., de Carvalho, L.F.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)
4. Cyganek, B.: Circular Road Signs Recognition with Soft Classifiers. *Integrated Computer-Aided Engineering* 14(4), 323–343 (2007)

5. Cyganek, B.: An Analysis of the Road Signs Classification Based on the Higher-Order Singular Value Decomposition of the Deformable Pattern Tensors. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2010, Part II. LNCS, vol. 6475, pp. 191–202. Springer, Heidelberg (2010)
6. Cyganek, B.: Image Segmentation with a Hybrid Ensemble of One-Class Support Vector Machines. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS (LNAI), vol. 6076, pp. 256–263. Springer, Heidelberg (2010)
7. Derrac, J., García, S., Herrera, F.: A First Study on the Use of Coevolutionary Algorithms for Instance and Feature Selection. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
8. Hsu, R.-L., Abdel-Mottaleb, M., Jain, A.K.: Face Detection in Color Images. IEEE PAMI 24(5), 696–707 (2002)
9. Jones, M.J., Rehg, J.M.: Statistical Color Models with Application to Pixel-Level Human Skin Detection. IEEE Int. Conf. Pattern Recognition 1, 1056–1059 (2000)
10. Kawaguchi, T., Hidaka, D., Rizon, M.: Detection of eyes from human faces by Hough transform and separability filter. In: Int. Conf. Image Processing, vol. 1, pp. 49–52 (2000)
11. Kittler, J., Hatef, M., Duing, R.P.W., Matas, J.: On Combining Classifiers. IEEE PAMI 20(3), 226–239 (1998)
12. Kothari, R., Mitchell, J.L.: Detection of eye locations in unconstrained visual images. Int. Conf. Image Processing 3, 519–522 (1996)
13. Kuncheva, L.I.: Combining Pattern Classifiers. Wiley, Chichester (2004)
14. de Lathauwer, L.: Signal Processing Based on Multilinear Algebra. PhD dissertation, Katholieke Universiteit Leuven (1997)
15. de Lathauwer, L., de Moor, B., Vandewalle, J.: A Multilinear Singular Value Decomposition. SIAM J. Matrix Analysis and App. 21(4), 1253–1278 (2000)
16. Ma, Y., Ding, X., Wang, Z., Wang, N.: Robust precise eye location under probabilistic framework. In: IEEE Int. Conf. on Automatic Face and Gesture Recogn., pp. 339–344 (2004)
17. D’Orazio, T., Leo, M., Guaragnella, C., Distante, A.: A visual approach for driver inattention detection. Pattern Recognition 40, 2341–2355 (2007)
18. Peer, P., Kovac, J., Solina, F.: Human skin colour clustering for face detection. In: EUROCON 2003 – International Conference on Computer as a Tool (2003)
19. Polikar, R.: Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine, 21–45 (2006)
20. Savas, B., Eldén, L.: Handwritten digit classification using higher order singular value decomposition. Pattern Recognition 40, 993–1003 (2007)
21. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensor-Faces. In: European Conference on Computer Vision, Denmark, pp. 447–460 (2002)
22. Wang, P., Green, M., Ji, Q., Wayman, J.: Automatic Eye Detection and Its Validation. In: CVPR 2005 Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 03, pp. 164–171 (2003)
23. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants – Evolutionary and Neural Methods of Training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS (LNAI), vol. 6076, pp. 590–597. Springer, Heidelberg (2010)
24. Zhu, Z., Jib, Q.: Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. Computer Vision and Image Understanding 98, 124–154 (2005)
25. Zhou, Z.H., Geng, X.: Projection functions for eye detection. Pattern Recognition 37(5), 1049–1056 (2004)
26. Phung, S.L., Bouzerdoum, A., Chai, D.: Skin Segmentation Using Color Pixel Classification: Analysis and Comparison. IEEE PAMI 27(1), 148–154 (2005)

A Hybrid Context-Aware Wearable System with Evolutionary Optimization and Selective Inference of Dynamic Bayesian Networks

Jun-Ki Min and Sung-Bae Cho

Department of Computer Science, Yonsei University
262 Seongsanno, Seodaemun-gu, Seoul 120-749, Korea
loomlike@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. Multiple sensor-based context inference systems can perceive users' tasks in detail while it requires complicated recognition models with larger resources. Such limitations make the systems difficult to be used for the mobile environment where the context-awareness would be most needed. In order to design and operate the complex models efficiently, this paper proposes an evolutionary process for generating the context models and a selective inference method. Dynamic Bayesian networks are employed as the context models to cope with the uncertain and noisy time-series sensor data, where the operations are managed by using the semantic network which describes the hierarchical and semantic relations of the contexts. The proposed method was validated on a wearable system with variable sensors including accelerometers, gyroscopes, physiological sensors, and data gloves.

Keywords: Context-awareness, dynamic Bayesian networks, evolutionary algorithm, selective inference.

1 Introduction

With the growth of the mobile sensor technologies, there are many attempts to recognize user contexts and to provide situation-appropriate services by using the wearable sensors [1]. These systems have combined various types of sensors in order to perceive the high-level contexts more accurately [2]. For example, vision, illumination, and noise detection sensors are employed to understand environmental situation, while wearable sensors including accelerometers and gyroscopes are used for capturing human activities.

As a large number of sensors are used for the systems, the training and operating of the inference model becomes difficult because of the highly complicated causal relationships among sensing information and target contexts. A substantial burden would be placed on the power consumption, bandwidth of the system, and computational complexity as well [3, 4]. Several modularization methods, such as grouping correlated sensors with contexts [5] and decomposing the complex problem into sub-problems [6], have been considered to resolve the problems, yet the model optimization is still a difficult issue. Moreover, how to combine the multiple models is the

key to conduct the systems effectively. Since the systems do not know which contexts have to be recognized prior to the others and which contexts do not need to be identified, they should operate all the models and choose the context of the highest confidence as the final inference result. In this case, the essential problems of the large scale context-awareness still remain.

In this paper, an effective context-aware method based on evolutionary optimization and selective inference is proposed, which is capable of recognizing complex contexts under a computationally restricted environment. The optimal one context-versus-all (OVA) dynamic Bayesian networks (DBNs), which can handle the uncertain, incomplete, and time-series sensory observations, are constructed by using a genetic algorithm (GA). The processing cost of the system is then reduced by selecting the models to be evaluated based on the semantic network (SN) where the relations among the sensing values and contexts are hierarchically represented.

2 Background

Recently, integrating various sensors has been investigated to recognize diverse contexts such as eating, typing, shaking hands, clapping hands, driving, brushing teeth, and washing the dishes [7]. Oliver et al. [8] proposed the use of layered probabilistic representations for modeling office activities such as phone conversation and presentation. They collected real-time streams of evidence from video, audio, and computer interactions. Ermes et al. [9] recognized daily activities and sports with wearable sensors of 3D accelerometers and a GPS receiver, in which the activities included walking, running, cycling with an exercise bike, playing football, etc.

Incorporating a large number of sensors implies many difficulties as described in Section 1. Schmidt et al. [10] presented a layered architecture for context-aware adaptation based on collections of multimodal sensors (including a photodiode, two accelerometers, a passive IR, a temperature/pressure sensor, and a CO gas sensor) in the TEA project. Their approach first analyzes the situation cues by using a self-organizing map, and recognizes the corresponding contexts with rule-based models. As more recent studies have increasingly considered embedding of direct awareness in mobile devices, minimization of the overall system cost is getting important. Wang et al. [3] proposed a hierarchical sensor management scheme based on the state transition rules which turn on and off the sensors. Krause et al. [4] suggested a collection of selective sampling strategies in order to reduce the number of required sensor readings and the computation duty cycles even further.

3 Context-Model Optimization and Selective Inference

Fig. 1 shows the overall process of the proposed method. For the incoming data, the activation values (degrees of occurrence) are propagated from low-level sensor nodes to high-level contexts nodes on the semantic network. The activated contexts are then recognized by using the OVA DBNs trained by using GA.

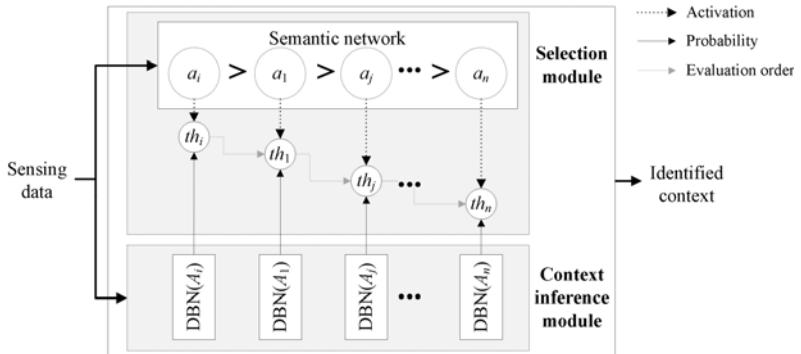


Fig. 1. Overall process of the proposed method

3.1 Evolutionary Optimization

A Bayesian network is a powerful tool for representing a large number of joint probability among the variables, and it effectively handles the uncertainty as well [6]. In order to construct a DBN, a model structure has to be identified first, and conditional probability tables (CPTs) are then estimated. We defined four kinds of nodes as shown in Fig. 2(a). A_c is the context node to be recognized in the current time, and A_p is the past state of the context. E_c and E_p are sensing data (evidences) of the current and past time, respectively. According to the common cause-effect relationships, A_c generates E_c where A_c is caused by A_p and E_p . The probability of the context $P(A_c = \text{yes} | A_p, E_c, E_p)$ is calculated under Bayes theorem with Markov chain by:

$$P(A_c = \text{yes} | A_p, E_c, E_p) = \frac{P(A_c = \text{yes}, A_p, E_c, E_p)}{\sum_{A_c=\text{yes,no}} P(A_c, A_p, E_c, E_p)}, \text{ where} \quad (1)$$

$$P(A_c, A_p, E_c, E_p) = P(A_p)P(E_p)P(A_c | A_p, E_p)P(E_c | A_c).$$

An expert DBN computes the observation probability of the corresponding activity state using a number of continuing sensory observations.

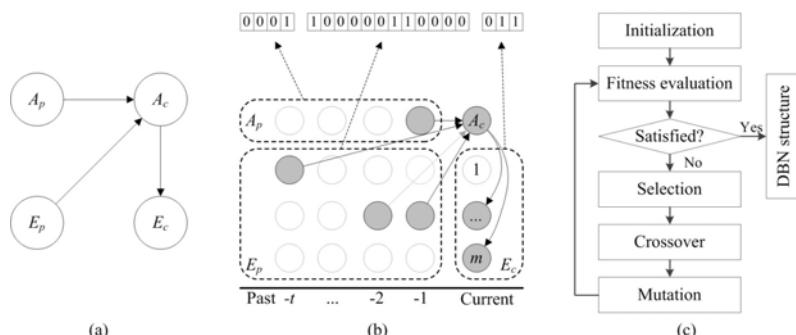


Fig. 2. (a) Constraint of the link directions for the context inference DBNs where A_p = past context, A_c = current context, E_p = past evidence, and E_c = current evidence. (b) Examples of a DBN. (c) The overall process of the genetic algorithm used in this paper.

Since time-series variables often lead to complexity problems, the proposed method applies GA for finding optimal structure among the large number of variables, while it trains the CPTs by using an expectation-maximization (EM) algorithm. GA is a search technique based on the evolution of biological system [11]. According to a predefined evaluation criterion called fitness (use the recognition rate for the training dataset in this paper), fitter individuals survive into the next generations after some modifications using crossover and mutation. First, construct a random initial population where the structure of DBN is represented as a chromosome (see Fig. 2b). The chromosome is composed of a fixed-length binary string to determine the edges between A_c and corresponding variables. In the selection phase, a roulette wheel method is adopted to select chromosomes to be crossed over and mutated. The crossover process recombines the genetic material of two parent chromosomes by exchanging pairs of their genes. Here, two points crossover is used where the first point is chosen from A_p and E_p , while the second point is picked from E_c . We used elitism which reserves two slots in the next generation for the highest scoring chromosome of the current generation, without allowing crossover and mutation. Fig. 2(c) shows the overall GA process.

3.2 Selective Inference

A SN is a directed graph which represents semantic relations (links) among concepts or common sense knowledge represented by nodes [12]. In this paper, a SN is constructed for modeling relations among sensing information, low-level contexts, and high-level contexts. Here, the activation values (the degree of probable occurrence for a context) are propagated to other context nodes with different weights according to the directions and the types of the links.

Six types of links are used for the proposed SN as {PartOf, Cause, Prevent, FollowedBy, HardToFollow, Prerequisite}, where the first three of them are used for defining relations between low-level and high-level contexts, and the rests are for the high-level to high-level connections. The activation value of a high-level context node a_h is estimated as follows:

$$a_h = a_{hl} + a_{hh} + a_{hr} . \quad (2)$$

Here, a_{hl} and a_{hh} are the activation values propagated from the low-level context nodes (the first-level propagation) and other high-level context nodes (the second-level propagation), respectively, while a_{hr} is estimated from the previous recognition state of the corresponding context (activation from goal).

For the first-level propagation, a_{hl} is calculated based on the three types of relations (PartOf, Cause, and Prevent) with a different weight w by:

$$a_{hl} = \sum w_R a_l m_R, \text{ where the relation } R \in \{\text{PartOf, Cause, Prevent}\} . \quad (3)$$

In the case of Cause and Prevent relations, momentum variable m can be used to reduce the drastic changes of the activation values. The activation value for the low-level context node a_l is estimated by using simple rules with feature vectors.

The previous recognition state of the high-level context can be one of three types: not executed (inactivated), rejected (the output probability of the inference model is

less than a threshold), and accepted. Let $P(t)$ be the output value of the context inference model at time t . Here, a_{hr} is calculated as follows:

$$a_{hr} = \begin{cases} m_{\text{not_executed}} P(t') & \text{not executed,} \\ - \sum_{t=1}^n (1 - (1/(1 + e^{P(t)})))/n & \text{where the context is rejected,} \\ \sum_{t=1}^n (1/(1 + e^{P(t)}))/n & \text{accepted.} \end{cases} \quad (4)$$

If the context model was not executed at the previous time t , $P(t')$ the output value at the latest execution time t' is used with the momentum variable $m_{\text{not_executed}}$. If the recognition engine rejected the context, the activation value is decreased by the inverse sigmoid function. Otherwise (accepted state), the sigmoid function is used to increase the activation value.

At the second-level propagation, Prerequisite, FollowedBy, and HardToFollow relations among the high-level contexts are considered as follows:

$$a_{hh} = \sum w_R a'_h m_R, \quad (5)$$

where the relation $R \in \{\text{Prerequisite}, \text{FollowedBy}, \text{HardToFollow}\}$.

We used $a'_h = a_{hl} + a_{hr}$ (differ from Eq.2) in order to avoid the recursive reference of a_{hh} . In this paper, the weight values were defined as $\{w_{\text{PartOf}}, w_{\text{Cause}}, w_{\text{Prevent}}, w_{\text{Prerequisite}}, w_{\text{FollowedBy}}, w_{\text{HardToFollow}}\} = \{0.5, 0.6, -0.2, 0.7, 0.3, -0.5\}$, while all the momentum variables were set as one. Fig. 3 shows the overall process of the activation propagation.

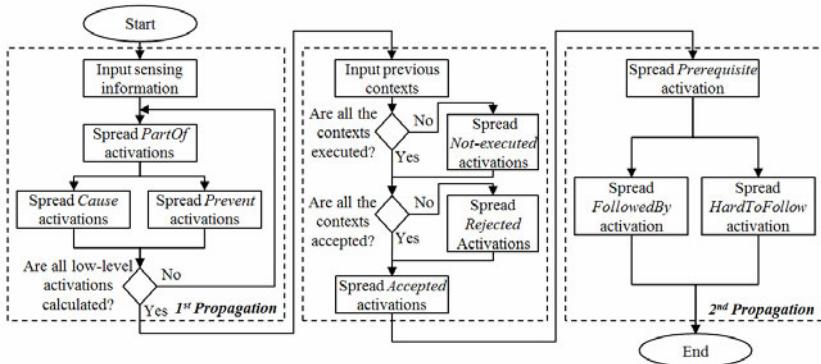


Fig. 3. The overall process diagram of the activation propagation

4 Experimental Results

4.1 The Sensor Platform and Scenario Analysis

A sensor platform used in this paper integrates two data gloves; an armband on a right arm; five sets of accelerometers and gyroscopes on a head and both arms and wrists; a GPS receiver; and a notebook that collecting sensor data and processing the proposed

system as shown in Fig. 4(a). Data gloves (5DT Inc.) sense the degree of bending fingers from five channels of the left hand and 14 channels of the right hand. Bodymedia's armband captures the user's physiological signals, related to emotion and stress, including a skin temperature, heat flux, and Galvanic skin response. In order to track human motion, XSens XBus Kit is used which includes a 3-axis accelerometer and a 3-axis gyroscope for each sensor node. The system collects the activity information by using the sensor modules periodically, where the smoothing, normalization, and quantization are conducted as the preprocessing. Sliding window-based averaging and K -means clustering algorithm are used for the smoothing and vector quantization, respectively.

Time	Scenario
6:00	Wake up in the morning
8:00	Drive a car to go to work
9:00	Work with a computer at the office
10:00	Meet colleagues for a business conference
11:00	Meeting (missed call occurred)
12:00	After the meeting, check the missed call
12:00	Go to lunch
13:00	Go back to the office and take a rest
14:00	Work with a computer at the office
16:00	Take a coffee and go to the rest room
16:00	Go back to work
18:00	Leave the office and go to gym

Fig. 4. (a) The multimodal sensor platform for the proposed context-awareness system and (b) the simulation scenario

Table 1. Cause-effect relations among the contexts where row activities are followed by the column ones (\blacktriangle : Cause, \blacktriangledown : Prevent, and -: Unrelated)

	D	HC	HS	TB	WD	KT	MS	C	S	ME	W	M	U	E	R
Driving (D)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HandClapping (HC)	-	\blacktriangle	\blacktriangle	-	-	-	-	-	-	-	-	\blacktriangle	-	-	-
HandShaking (HS)	-	\blacktriangle	\blacktriangle	-	-	-	-	-	-	-	-	\blacktriangle	-	-	-
TeethBrushing (TB)	-	-	-	\blacktriangledown	-	-	-	\blacktriangle	\blacktriangledown	-	-	\blacktriangle	-	-	-
WaterDrinking (WD)	-	-	-	-	\blacktriangledown	-	-	-	\blacktriangle	-	-	\blacktriangle	-	-	-
KeyboardTyping (KT)	-	-	-	-	-	\blacktriangle	\blacktriangle	-	-	\blacktriangle	-	-	-	-	-
MouseScrolling (MS)	-	-	-	-	-	\blacktriangle	\blacktriangle	-	-	\blacktriangle	-	-	-	-	-
Calling (C)	-	-	-	-	-	-	-	-	\blacktriangle	-	\blacktriangle	-	-	-	-
Sleeping (S)	-	-	-	\blacktriangle	-	-	-	\blacktriangledown	-	-	\blacktriangle	-	\blacktriangledown	-	-
MealEating (ME)	-	-	-	\blacktriangle	\blacktriangle	-	-	\blacktriangle	\blacktriangledown	-	\blacktriangle	-	\blacktriangle	-	\blacktriangle
Working (W)	-	-	-	-	-	\blacktriangle	\blacktriangle	-	\blacktriangledown	-	-	-	-	-	\blacktriangle
Meeting (M)	-	\blacktriangle	\blacktriangle	-	\blacktriangle	-	-	\blacktriangle	-	-	-	-	-	-	-
Urinating (U)	-	-	-	\blacktriangle	-	-	-	-	-	-	-	\blacktriangledown	-	-	-
Exercising (E)	-	-	-	-	\blacktriangle	-	-	-	-	-	-	-	\blacktriangledown	-	\blacktriangle
Resting (R)	-	-	-	-	-	-	-	\blacktriangle	-	\blacktriangle	-	-	-	-	-

In order to verify the proposed method, we considered one day scenario of the office environment, as described in Fig. 4(b), which includes 15 activities {Sleeping, MealEating, TeethBrushing, Driving, Working, KeyboardTyping, MouseScrolling, HandShaking, HandClapping, Meeting, Calling, WaterDrinking, Urinating, Exercising, Resting}. For a training set of the DBNs, a subject performed the activities 10 times over a period of two minutes. According to the domain knowledge and some constraints, the cause-effect relations among the contexts can be represented as shown in Table 1. Based on the definitions of the variables and relations, we designed the SN which consists of 34 nodes (15 high-level contexts and 19 low-level contexts) and 42 relations (13 PartOf, 9 Cause, 2 Prevent, 7 FollowedBy, 7 HardToFollow, and 2 Prerequisite).

4.2 Context Inference Results

For recognizing the contexts, 15 OVA DBNs were generated where the GA parameters were selected as follows: #population = 50, selection rate = 0.8, crossover rate = 0.8, mutation rate = 0.1, and #generation = 50. Fig. 5(a) illustrates an example of the evolution result for the Urinating context. Although the structure of each model was optimized based on the GA, applying the multiple models to the mobile environment is still difficult because of their high complexities. There are 35 past evidence nodes E_p and seven current evidence nodes E_C for each model in average. The average number of conditional probabilities ($S_{E_p}^{|E_p|} \times S_A + (S_A \times S_{E_C}) \times |E_C|$) was then 4,584 where S_{E_p} , S_A , and S_{E_C} denote the number of states for the past evidence node, the current evidence node, and the current context node, respectively. The selective inference approach of the proposed method can effectively handle this problem.

Fig. 5(b) shows the time cost of the proposed method. When the selective inference approach is not used, all of the 15 models have to be evaluated for every time-slice in order to recognize a context. The proposed method, however, exploited 1.55 models (66.4ms in a Pentium4 PC) per context in average. The overall inference accuracy was 91.5%.

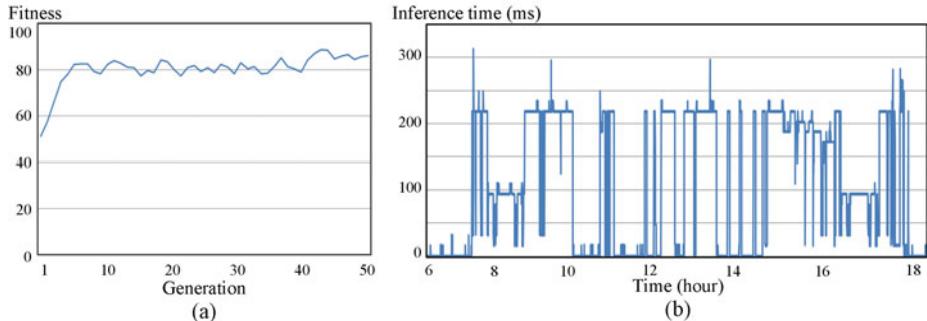


Fig. 5. (a) The evolution result for the Urinating context and (b) the inference cost of the proposed method

5 Concluding Remarks

For the mobile context-aware systems with multiple wearable sensors, the efficiency of the system is essential. In this paper, an evolutionary process for generating the

context models and a selective inference method were presented. A genetic algorithm was used to optimize the structures of the probabilistic time series models. Meantime, the relationships between the contexts and sensing variables were analyzed in order to construct the semantic network which selects the appropriate models dynamically. The proposed method was implemented with multiple wearable sensors, and experimental results showed that it effectively reduces the complexity of the context-aware systems. Since users have different behavioral patterns for each other, we will consider their variations to build the realistic semantic models as the future work.

Acknowledgement. This research was supported by the Converging Research Center Program through the Converging Research Headquarter for Human, Cognition and Environment funded by the Ministry of Education, Science and Technology (2010K001173).

References

1. Roy, N., Gu, T., Das, S.K.: Supporting Pervasive Computing Applications with Active Context Fusion and Semantic Context Delivery. *Perv. and Mobile Comp.* 6, 21–42 (2010)
2. Laerhoven, K.V., Aidoo, K.A., Lowette, S.: Real-time Analysis of Data from Many Sensors with Neural Networks. In: Proc. of the 5th Int. Symposium on Wearable Computers, pp. 115–123 (2001)
3. Wang, Y., Lin, J., Annavaram, M.: A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition. In: Proc. of MobiSys., pp. 179–192 (2009)
4. Krause, A., Ihmiq, M., Rankin, E., Leong, D., Smriti, G., Siewiorek, D., Smailaqic, A., Deisher, M., Senqupta, U.: Trading off Prediction Accuracy and Power Consumption for Context-aware Wearable Computing. In: ISWC 2005, pp. 20–26 (2005)
5. Hwang, K.-S., Cho, S.-B.: Landmark Detection from Mobile Life Log using a Modular Bayesian Network Model. *Expert Syst. Appl.* 36, 12065–12076 (2009)
6. Hong, J.-H., Yang, S.-I., Cho, S.-B.: ConaMSN: A Context-aware Messenger using Dynamic Bayesian Networks with Wearable Sensors. *Expert Syst. Appl.* 37, 4680–4686 (2010)
7. Blum, M., Pentland, A., Troster, G.: InSense: Interest-based Life Logging. *IEEE Multimedia* 13, 40–48 (2006)
8. Oliver, N., Garg, A., Horvitz, E.: Layered Representations for Learning and Inferring Office Activity from Multiple Sensory Channels. *Comput. Vis. Image Understand.* 96, 163–180 (2004)
9. Ermes, M., Parkka, J., Mantyjarvi, J., Korhonen, I.: Detecting of Daily Activities and Sports with Wearable Sensors in Controlled and Uncontrolled Conditions. *IEEE T. Inf. Technol. B.* 12, 20–26 (2008)
10. Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Laerhoven, K.V., Velde, W.V.D.: Advanced Interaction in Context. In: 1st Int. Symposium on Handheld and Ubiquitous Computing, pp. 89–101 (1999)
11. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
12. Marinov, M., Zheliazkova, I.: An Interactive Tool based on Priority Semantic Networks. *Knowledge-Based Systems* 18, 71–77 (2005)

Global/Local Hybrid Learning of Mixture-of-Experts from Labeled and Unlabeled Data

Jong-Won Yoon and Sung-Bae Cho

Dept. of Computer Science, Yonsei University,
134 Shinchon-dong, Seodaemoon-gu, Seoul, Korea
jwyoohn@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. The mixture-of-experts (ME) models can be useful to solve complicated classification problems in real world. However, in order to train the ME model with not only labeled data but also unlabeled data which are easier to come, a new learning algorithm that considers characteristics of the ME model is required. We proposed global-local co-training (GLCT), the hybrid training method of the ME model training method for supervised learning (SL) and the co-training, which trains the ME model in semi-supervised learning (SSL) manner. GLCT uses a global model and a local model together since using the local model only shows low accuracy due to lack of labeled training data. The models enlarge the labeled data set from the unlabeled one and are trained from it by supplementing each other. To evaluate the method, we performed experiments using benchmark data sets from UCI machine learning repository. As the result, GLCT confirmed the feasibility of itself. Moreover, a comparison experiments to show the excellences of GLCT showed better performance than the other alternative method.

Keywords: mixture of experts, co-training, semi-supervised learning.

1 Introduction

Many classification problems in real world usually have complicated problem spaces, and only one model for classification which covers entire problem spaces may not produce accurate results. One of ways to solve this is to use several local experts such as the mixture-of-experts (ME) [1]. Since the model divides the problem into smaller sub-problems, its complexity can be reduced and it turns to be easier.

Prior to apply the ME model to the problems, it should be trained first with training data instances. There are already several learning techniques for the ME model [1] which uses labeled data only. However, in many machine learning settings, unlabeled examples are significantly easier to come by than labeled ones [2]. Moreover, unlabeled data can be used to augment labeled data. In these reasons, semi-supervised learning (SSL) approach which uses both data has received attention recently. There are several SSL techniques for general models. But in order to train the ME model in semi-supervised manner, the techniques should consider characteristics of the model.

In this paper, we present global-local co-training (GLCT), the hybrid method of the conventional ME training method and the co-training scheme [3]. It uses global and local views. The global view is represented by the global model, and the ME model what we want to train corresponds to the local view. Each model has its own weaknesses if it is used solely, nevertheless, we take only strengths of them by using both model together.

The role of the global model is to generalize entire problem space. Even if there are only small amount of labeled data, it less tends to be biased than local experts because it is less sensitive to the size of the data than the local one. On the other hand, if the local model has the expert which is enough specialized to specific region which the global one has not been learned delicately about, it can help to improve the opponent.

We also conduct experiments to evaluate our proposed method via experiments with data sets from UCI machine learning data repository.

2 Backgrounds

2.1 Co-training

Co-training is the multi-view SSL approach proposed by Blum and Mitchell [4]. The algorithm works well if the feature set division of dataset satisfies two assumptions:

- Each set of features is sufficient for classification
- The two feature sets of each instance are conditionally independent given the class

Co-training uses two learnable models with different feature sets. The models are trained independently and each model enlarges the opponent's training data set by predicting unlabeled instances in it iteratively. Co-training firstly applied to classify web pages [4] and has been widely used for various classification problems in semi-supervised approach [5, 6].

Co-training has been frequently compared with the Expectation-Maximization (EM) algorithm [7], another representative SSL technique. Recently, co-training has been more preferred than the EM because of the model assumption problem of the EM [8], also it can applied to train the ME in SSL manner.

Despite the original co-training requires the conditional independencies between the two feature sets as views, it is not easy to separate feature sets nicely in real-world problems. To overcome this problem, Goldman and Zhou discovered that two models trained by different learning methods can be regarded as two different views also [9]. With considering this, the models built by different learning methods were used as different views in order to adopt the co-training to train the ME.

2.2 Mixture-of-Experts in Semi-supervised Learning

ME models [1] consist of a set of local experts and a gating network which combines the decisions from experts. One motivation for the ME model is based in the divide-and-conquer principle. According to the principle, the model should work well for problems that are composed of smaller unconnected ideas. The model has been applied to traditional complicated recognition problems [10,11].

In supervised learning (SL), the ME model can be trained easily by using conventional machine learning techniques, for example, MLP [12] or support vector machine (SVM) [13]. However, in SSL, some additional techniques are required to deal with unlabeled data.

One simple way of the techniques is to apply previous SSL techniques such as self-training or co-training directly to the model. However, it is known that using only the local model can be biased or variances can be increased when there are only a few amounts of training data [14]. Moreover, EM for the ME model [15] also has a limitation due to the model assumption that already mentioned in previous section.

In this paper, we proposed a SSL method for ME by hybridizing the training method for the ME in SL, and the co-training scheme. It has its own contribution that it is a novel approach that applied the co-training scheme to train the ME model.

3 Global-Local Co-training

Figure 1 shows the procedure of the proposed global-local co-training (GLCT). The main idea of the method is to use *the global view* and *the local view*. Instead of dividing the feature set according to the views, we chose two different learning algorithms which generate both models.

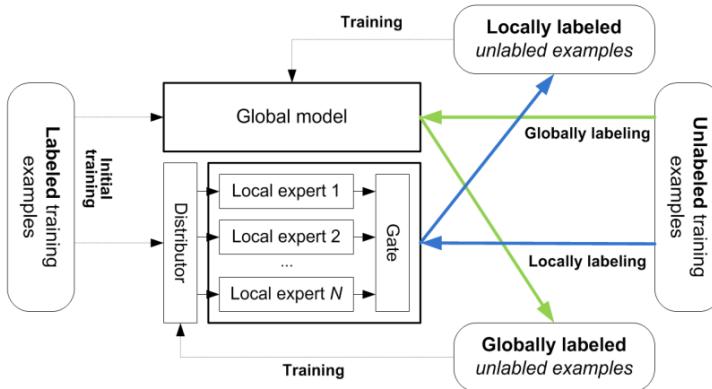


Fig. 1. The overall procedure of global-local co-training method

3.1 Algorithm

Figure 2 shows the algorithm of GLCT. Given a set L of labeled training instances, we can obtain the global model H_G and the ME model H_L . H_L consists of N experts $\mu_1, \mu_2, \dots, \mu_N$ and the gating network g through the training.

In each iteration, the algorithm chooses the instances which are going to be labeled based on two criterions, *preventing degradations* and *expecting improvements*.

Inputs:

Labeled training data L and Unlabeled training data U

Global model H_G and Mixture-of-expert model $H_L = (\mu_1, \mu_2, \dots, \mu_N, g)$

Initialization:

Create training set $L_G \leftarrow L$ and $L_L \leftarrow L$

Repeat until L_G and L_L do not change:

Build classifier H_G and H_L using L_G and L_L , respectively using existing training method.
Use H_G and H_L to U .

Pick instances which H_G is more confident than H_L and add it to U'_L

Pick instances which H_G is less confident than H_L and add it to U'_G

Sort U'_L and U'_G by confidence in descending order

For each class C

Pick N instances gradually from U'_L with higher confidence than threshold e , add it to L_L with label C

Pick N Instances gradually from U'_G with higher confidence than threshold e , add it to L_G with label C

Leave out the picked instances from U , U_L , and U_G

Output:

The predicted labels for unlabeled examples

Global model H_G and Mixture-of-expert model H_L

Fig. 2. Global-local co-training algorithm

Since the models can be imperfect and show poorer performance than the opponent in some conditions, the algorithm compares the confidences from H_G and H_L to prevent degradations due to the imperfections. The models can label instances for the opponents if only each model is more confident than it. The instances passed the first criteria are added to candidate sets U'_L or U'_G .

After obtain the candidate sets, they are sorted with considering confidence degrees in descending order. At the next step, at most N instances for each class with higher confidence degree than the certain threshold e are chosen gradually from the top of the set to be labeled. By choosing instances with sufficient confidence, it is expected that the models would be improved at the next iteration.

The chosen instances are added to the training data set for each model, L_G and L_L . The models are trained with newly extended labeled data set. If L_G and L_L do not change, the training ends.

3.2 Measuring Confidence Degrees

In order to obtain confidence degrees of examples, we designed the measure for them. Let $conf(w, x)$ is the original confidence degree for input instance x defined by the corresponding model w . $conf(w, x)$ can be changed by the type model used to build the global or the local model. In this work, we implemented both the global and the local model by using MLP (see 4.1). The confidence degree can be estimated as the output of MLP, and defined as below:

$$conf(w, x) = w(x) \quad (w(x) \text{ is output of } w \text{ for } x). \quad (1)$$

Since the original problem space was divided into several sub-spaces, the number of training data for each expert is rather smaller than the case of the global model. This raises chances that training data for certain classes are missing, and experts may generate unexpected outputs if they have not been learned about those classes. To prevent this problem, the existence of training instances of classes should be checked prior to compute confidence degrees. Eq (2), (3) and (4) shows the modified confidence measures for class C of the global model and the ME model, respectively.

$$\text{conf}(H_G, x, C) = \begin{cases} H_{G,C}(x), & \text{if } N_{HG,C} > 0 \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

$$\text{conf}(H_L, x, C) = \sum_{i=1}^N g_i \text{conf}(\mu_i, x, C). \quad (3)$$

$$\text{conf}(\mu_i, x, C) = \begin{cases} \mu_{i,C}(x), & \text{if } N_{\mu_i,C} > 0 \\ 0, & \text{otherwise} \end{cases}. \quad (4)$$

where μ_i is i th local expert, and g is the gating network. $H_{G,C}(x)$ and $\mu_{i,C}(x)$ represent the C th output of H_G and μ_i respectively. g_i is the gating output for i th local expert and $N_{HG,C}$ and $N_{\mu_i,C}$ are the numbers of training instances for class C used to train H_G and μ_i , respectively.

4 Experiments

4.1 Model Implementation

For the global model, general MLP was used. It also applied to the ME model, including experts and the gating network. The model designed to include total three local experts. Figure 3 shows the structure of the used ME model. As the training method for the ME model with labeled data, learning technique for MLP-based ME [15] was used. The number of hidden nodes for each network was fixed to ten.

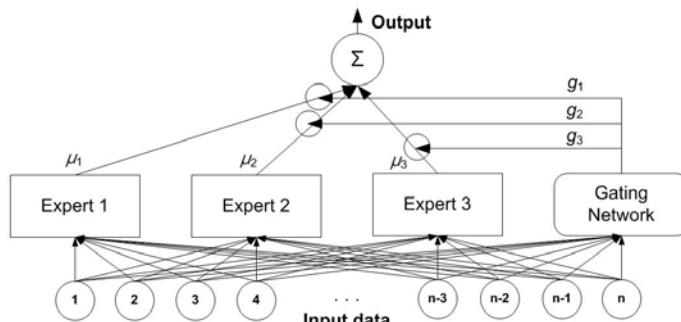


Fig. 3. Used mixture-of-experts model

4.2 Data Sets

We used three of the UCI benchmark data sets for the experiments; Wine, Breast Cancer, and SPECT Heart. Table 1 shows a summary of used data sets. Data instances were partially labeled depending on the ratio of labeled instances for all data sets. Labeled data instances were chosen randomly without considering class distributions.

Table 1. A summary of used data sets

Name	# of Attributes	# of Classes	# of Instances
Wine	13	3	178
Breast Cancer	32	8	569
SPECT Heart	22	2	267

4.3 Experimental Results

The performances of the ME model trained only with labeled data (ME_L) and the model with both labeled and unlabeled data (ME_UL) was measured by five-folds cross validations. Moreover, the average labeling accuracy (LA) was also obtained. All experiments were performed five times and results were averaged.

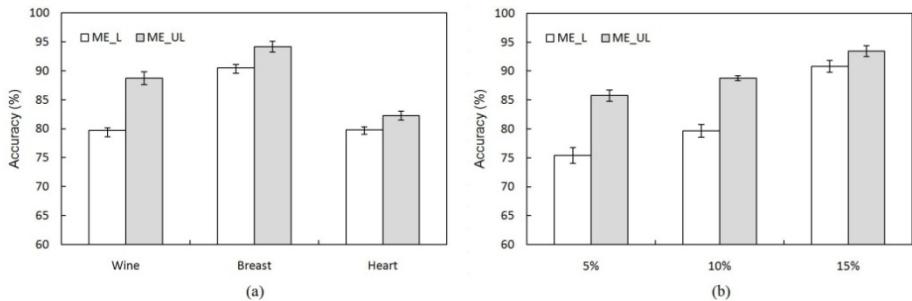


Fig. 4. The ME model performance comparison result (a: according to data sets, b: according to the ratio of labeled instances)

First of all, we performed learning process for all data sets with 10% the ratio of labeled instances to show the feasibility of the proposed GLCT. As shown in Figure 4(a), the ME model which trained with both labeled and unlabeled data showed better performance than the model with only labeled data. This results implied that the GLCT can train the successfully ME model in semi-supervised approach.

Moreover, in order to show that GLCT performs well regardless of the ratio of initial labeled data, we changed it from 5% to 15%. In this experiment, only the Wine data set was used. Figure 4(b) show the result of the experiment. The performance of the model was improved in every case from 4% to 8%. With only 5% initial labeled data, the model showed the most significant improvement than other cases even though it has the lowest performance. However, with 15% initial labeled data, the results showed that it showed the highest performance but slight enhancement.

Finally, we compared GLCT with the alternative way to train the ME model to show the excellences of it. In this experiment, the self-training algorithm [8] was chosen as the alternative. Figure 5 shows the algorithm of the self-training.

Inputs:

Labeled training data L and Unlabeled training data U

Classifier A

Repeat until L_G and L_L do not change:

Build classifier A using L .

Use A to U , pick instances with higher confidence degrees, and add them into L

Output:

The predicted labels for unlabeled examples

One trained classifier A

Fig. 5. Self-training algorithm for the comparison experiment

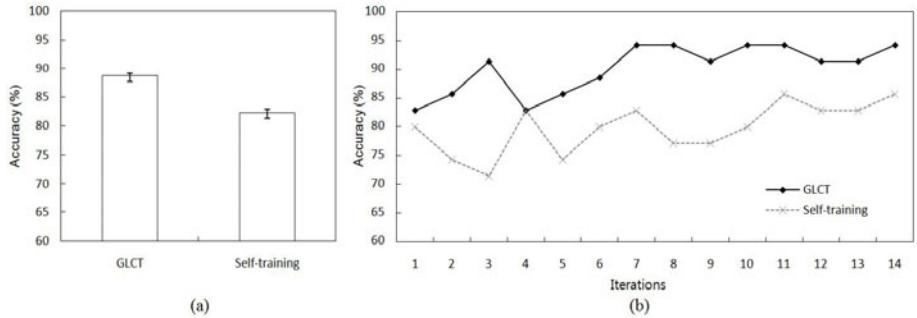


Fig. 6. Results of comparison experiments (a: Performances comparison, b: An example of accuracies according to iterations from one trial)

For the experiment, the Wine data set with 10% labeled instances was used. Figure 6(a) shows results of performance comparisons. GLCT shows 6.44% improved performance than the self-training algorithm. As shown in Figure 6(b), initial accuracies of two methods were rather similar. However, as iterations increased, the performance of GLCT was improved faster than the case of the self-training.

5 Concluding Remarks

In this paper, we proposed the global-local co-training (GLCT) methods to train the mixture-of-experts (ME) model with both labeled and unlabeled data. The method is another variation of co-training, however, is modified to be specialized to train the ME model by hybridizing the existing ME training method and the co-training. In order to overcome the limitation of the local experts with small amount of training data, we employed the global model to support the experts during the training.

We conducted experiments using benchmark data sets from UCI machine learning data repository. The results presented that GLCT can train the ME model well by predicting unlabeled data accurately. Moreover, the method which uses the self-training algorithm directly for the training was compared, and GLCT showed better performance.

In the future, we are planning to proof theoretically that the ME model is learnable via GLCT. Additionally, other data sets which are more complicated or contain real world problems should be tested to show the usefulness of the proposed method.

Acknowledgement. This research was supported by the Converging Research Center Program through the Converging Research Headquarter for Human, Cognition and Environment funded by the Ministry of Education, Science and Technology (2010K001173).

References

1. Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptive mixture local experts. *Neural Computation* 3(4), 79–87 (1991)
2. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. Wiley, Chichester (1973)
3. Wang, W., Zhou, Z.-H.: Analyzing co-training style algorithms. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 454–465. Springer, Heidelberg (2007)
4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of the 11th Annual Conf. on Computational Learning Theory, pp. 92–100 (1998)
5. Kiritchenko, S., Matwin, S.: Email classification with co-training. In: Proc. of the 2001 Conf. of the Centre for Advanced Studies on Collaborative Research, pp. 1–10 (2001)
6. Fang, Y., Cheng, J., Wang, J., Wang, K., Liu, J., Lu, H.: Hand posture recognition with co-training. In: Proc. of Int'l Conf. on Pattern Recognition, p. 104 (2008)
7. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39(2–3), 103–134 (2000)
8. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Proc. of the 9th Int'l Conf. on Information and Knowledge Management, pp. 86–93 (2000)
9. Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proc. of the 17th Int'l Conf. on Machine Learning, pp. 327–334 (2000)
10. Ubeyli, E.D.: Wavelet/mixture of experts network structure for EEG signals classification. *Expert Systems with Applications* 34(3), 1954–1962 (2008)
11. Ebrahimpour, R., Kabir, E., Esteky, H., Yousefi, M.R.: View-independent face recognition with mixture of experts. *Neurocomputing* 71(4–6), 1103–1107 (2008)
12. Ebrahimpour, R., Kabir, E., Yousefi, M.R.: Face detection using mixture of MLP experts. *Neural Processing Letters* 26(1), 69–82 (2007)
13. Lima, C.A.M., Coelho, A.L.V., Zuben, F.J.V.: Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification. *Information Sciences* 177(10), 2049–2074 (2007)
14. Ko, A.H.R., Sabourin, R., Britto Jr., A.S.: From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition* 41(5), 1718–1731 (2008)
15. Karakoulas, G., Salakhutdinov, R.: Semi-supervised mixture-of-experts classification. In: Proc. of 4th IEEE Int'l Conf. on Data Mining, pp. 138–145 (2004)

Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer

Young-Seol Lee and Sung-Bae Cho

Dept. of Computer Science, Yonsei University
262 Seongsanno, Seodaemun-gu, Seoul 120-749, Korea
tiras@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. As smartphone users have been increased, studies using mobile sensors on smartphone have been investigated in recent years. Activity recognition is one of the active research topics, which can be used for providing users the adaptive services with mobile devices. In this paper, an activity recognition system on a smartphone is proposed where the uncertain time-series acceleration signal is analyzed by using hierarchical hidden Markov models. In order to address the limitations on the memory storage and computational power of the mobile devices, the recognition models are designed hierarchy as actions and activities. We implemented the real-time activity recognition application on a smartphone with the Google android platform, and conducted experiments as well. Experimental results showed the feasibility of the proposed method.

Keywords: Hierarchical hidden Markov models, 3-axis accelerometer, activity recognition, Google android phone.

1 Introduction

As many types of smartphones such as Google Android phone and Apple iPhone are released and the use of smartphones increases, services using built-in sensors on a smartphone is also increasing. Accordingly, a lot of research is processing data collected from multiple sensors on a smartphone. Especially, some sensors like accelerometers have been used to recognize a person's activities in mobile devices before smartphones appear. Various studies on the user activity have been investigated to provide adaptive services on the mobile devices like healthcare systems for disabled or elderly people.

Some systems are developed to provide context aware services using a smartphone user's behaviors. Böhmer et al. developed a system to select suitable applications from a user's behavior on the Android platform [1]. Bellotti et al. developed a system named Maggit which inferred a user's activities (Eating, Shopping, etc.) and recommended some content on Windows Mobile platform [2]. Probabilistic models are appropriate for dealing with vagueness and uncertainty in real life for context-aware services. However, it is difficult to apply them to mobile devices because it requires a lot of memory and CPU time. Hybridization of intelligent techniques is the combination of different computational intelligence techniques. The hybrid intelligent systems become

popular to deal with many pattern recognition tasks [3][4]. In this paper, we proposed a hierarchical probabilistic model based approach to recognize a user's activities. It is applied to acceleration data gathered from an Android smartphone. Especially, it consists of two different kinds of probabilistic models which are continuous HMM and discrete HMM. Also, the feasibility of the method is shown by experiments.

2 Related Works

2.1 Activity Recognition with Accelerometers

There are many attempts to recognize a user's actions and behaviors with accelerometers. Table 1 summarizes some studies.

Table 1. Activity recognition using accelerometers

Author	Classifier	Contribution	Sensor	Year
Kwapisz et al. [5]	ANN, J48	Need for a little data size for recognition, fast recognition	Accelerometer	2010
Longstaff et al. [6]	NB, DT	Use of co-learning, high accuracy	GPS receiver, Accelerometer	2010
Maguire et al. [7]	kNN, J48	Fast recognition	Accelerometer, Heart-bit rate	2009
Gyorbiro et al. [8]	C4.5, NN	Real time recognition, efficient battery usage	Accelerometer	2009
Song et al. [9]	ANNs	Implementation on mobile environment	Accelerometer	2008
Zappi et al. [10]	HMM	Dynamic sensor selection	Accelerometer	2008
Yang et al. [11]	Neuro-fuzzy	Application of Neuro-fuzzy	Accelerometer	2007
Ganti et al. [12]	HMM	Implementation of GPS receiver, middleware	Accelerometer	2006

Many studies have typically been implemented in a mobile device as one of main contributions, and they also tried to recognize actions in real time with fast calculation speed. In this paper, we propose a novel method to use acceleration data for five seconds to recognize activities in real time.

2.2 Hierarchical Probabilistic Models

Hierarchical modeling approach is often applied to some probabilistic models such as Bayesian network [13], dynamic Bayesian network, hidden Markov model [14], etc. The approach is useful in most cases that patterns to recognize can be divided into smaller units with a hierarchical structure. Well designed hierarchical models improves accuracy and speed of recognition.

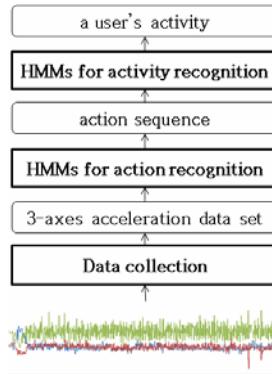
Table 2 summarizes some cases of hierarchical probabilistic models. In many cases, hierarchical model is used to recognize an activity that consists of some actions, or can be inferred from integration of each module which represents a part of a body. The approach is suitable to recognize a specific pattern from time-series patterns.

Table 2. Related works using hierarchical model

Author	Classifier	Contribution	Sensor	Year
Yang et al. [15]	HDBN	Activity recognition with multi-modal sensors	Accelerometer, wearable sensors	2009
Park et al. [16]	HDBN	Modularization using HDBN	Video camera	2004
Mengistu et al. [17]	HHMM	Understanding underlying semantics of words and sentences	Text data	2008
Wang et al. [18]	HDBN	Gesture recognition using HDBN	Video camera	2007
Du et al. [19]	HDBN	Hierarchical durational-state DBN	Video camera	2006

3 Proposed Method

The proposed structure consists of two steps of HMMs to analyze acceleration data and recognize a user's behavior. First, after acceleration data collected from a three-axis accelerometer on a smartphone, the acceleration data is transferred to low-level HMM to classify a user's actions. In the second phase, high-level HMM is used to recognize a user's activities from the set of actions. Figure 1 shows the process of the entire system.

**Fig. 1.** The flow of the whole process

3.1 HMMs for Action Recognition

In general, there are two methods, which are quantization and continuous HMM, to process continuous data. In this paper, continuous HMMs with Gaussian distribution are used to recognize a user's actions from the acceleration data.

Mostly continuous HMM training requires more time than discrete HMM learning, and it may burden real time pattern recognition with additional computation. Here, complexity and time required for the computation is not too large because we acceleration data for short time are used. A set of actions recognized by low-level HMMs are four kinds of actions such as stand, walk, run, stair up/down. Each HMM for each action has five hidden states. The frequency to collect acceleration data is 12hz, and data length for action recognition is five seconds. The data window size for action is 60.

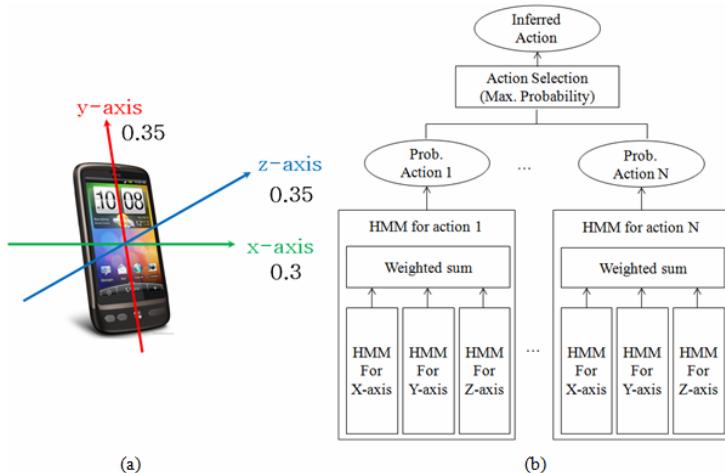


Fig. 2. (a) Directions and weights of X, Y, Z axes (b) HMM structure for action recognition

Acceleration data include x, y, z axis values as shown in Figure 2(a). In order to handle three-dimensional acceleration data, we train a HMM for each axis and integrate probability values of three axes with a weighted sum approach. Weight for each axis is shown as Figure 2(a). Figure 2(b) shows the structure of low level HMM for action recognition.

3.2 HMMs for Activity Recognition

In order to recognize a user's activities, previously recognized actions in step 1 are used as input data. In general, an activity consists of a number of actions. It is possible to recognize the user's activities if and only if the acceleration data should be collected over an enough period of time. The data window size for activity is 3600.

If a single-layered HMM is used to recognize the user's activities, the increase of the length of the acceleration data accompanies the increase of the number of state transitions. Many state transitions make the probability very small and require additional

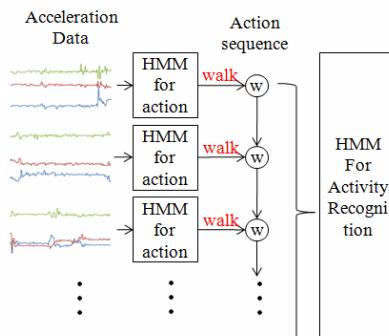


Fig. 3. HMM structure for activity recognition

calculations such as log function. However, a sequence of actions are used as input for the HMM for real-time activity recognition on a mobile device without using the acceleration data directly. It can reduce the required time for calculation and can enhance the precision. Figure 3 shows the basic structure for the activity recognition.

4 Experimental Results

4.1 Data Collection

Android OS based HTC Desire was used as a platform for data collection. Participants were three graduate students between 20-30 years old. They grasped the smartphone by hand for data collection. The amount of collected data for training action/activity recognition is shown in Table 3.

Table 3. Training HMMs for data size

Type	Class	Data size
Action	Stand	315 seconds
	Walk	463 seconds
	Stair Up/Down	1,178 seconds
	Run	213 seconds
Activity	Shopping	1,435 seconds
	Taking Bus	5,478 seconds
	Moving (by walk)	11,264 seconds

4.2 Evaluation of HMMs for Action Recognition

Figure 4(a) showed the performance of HMMs for action recognition. The experiment was done with 4-fold cross validation. Recognition for 'stand', 'run', 'walk' actions show good performance except for 'stair up/down' action. It is difficult to classify 'stair up/down' and 'walk' actions sometimes. There were various types of stairs, and some stairs have similar properties with undulating ways. Classification between 'stair up' and 'stair down' was conducted for a more detailed analysis of the precision evaluation.

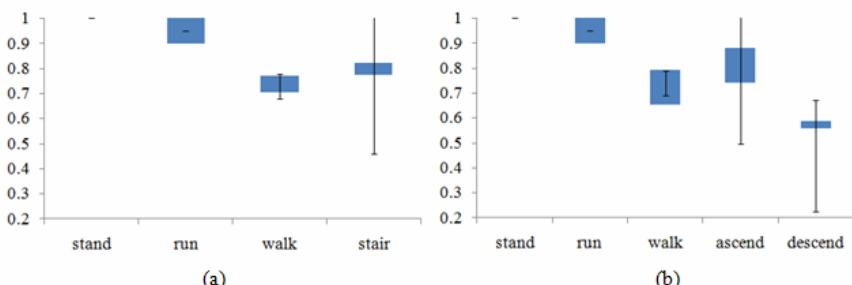


Fig. 4. (a) Precision of action recognition (b) Precision of action recognition (*stair up/down*)

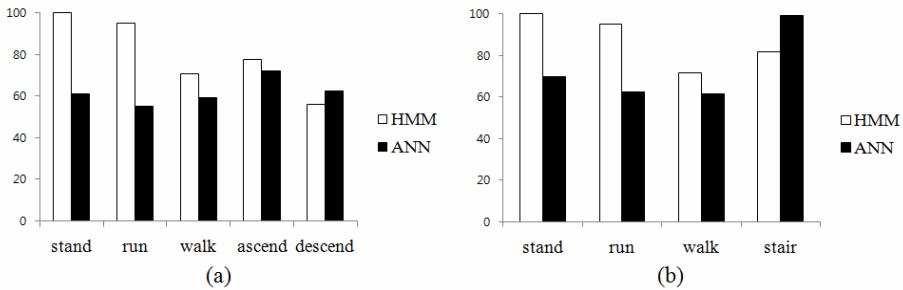


Fig. 5. Comparison of precision (*HMM and ANN*)

Figure 4(b) shows the result of the evaluation for more detailed actions including 'stair up' and 'stair down.' The result showed decrease of precision of 'walk' action. It implied that walk and stair up are confusing. Also, it meant that classification between 'stair up' and 'stair down' is very difficult by using a built-in accelerometer on a mobile phone. Also, Figure 5(a) and (b) shows the result of comparison with ANN(artificial neural network).

4.3 Evaluation of HMMs for Activity Recognition

We attempt to recognize three kinds of activities which can be determined by acceleration data. They are shopping, taking bus, and moving by walk. It is assumed that there are some acceleration patterns for all activities. For instance, when a user is shopping in a department store, he or she is going to repeat walking and standing regularly. If a user takes a bus, the acceleration pattern may be similar to 'standing' action for sitting in a seat. When a user moves for a long time, the user's acceleration pattern may show 'walking' and 'stair up/down'. Figure 6 shows the comparison among hierarchical HMM, HMM, and ANN for activity recognition. The two step hierarchical HMM shows better average performance than original HMM and ANN.

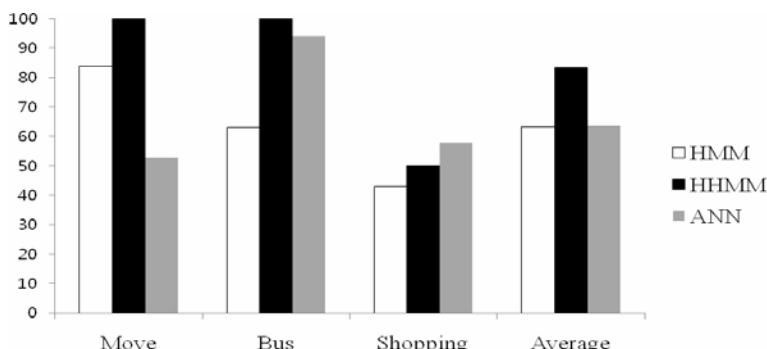


Fig. 6. Comparison of precision (*HHMM , HMM and ANN*)

5 Summary and Future work

In this paper, we attempt to recognize actions in real time on the Android platform, and recognize the user's activities from a recognized set of actions. The two step HMM structure is appropriate for mobile environment to reduce computational complexity. Looking at the results of action recognition, there is confusion between 'stair up/down' and 'walk' actions. Careful data selection and training will improve the performance of recognition. Also, 'shopping' and 'taking a bus' activities have some difficult patterns to classify, and it is necessary to use more sensors such as a GPS receiver, Wi-Fi, etc.

There still are many problems to be solved on the mobile activity recognition such as integration of multi-modal sensor data, and modeling user's variations. Moreover, the comparison with other methods such as DTW, and BN is also a very important issue to be considered as a future work.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0083838).

References

1. Corchado, E., Abraham, A., Carvalho, F.L.P.C.A.: Hybrid intelligent algorithms and applications. *Information Science* 180(14), 2633–2634 (2010)
2. Abraham, A., Corchado, E., Corchado, M.J.: Hybrid learning machines. *Neurocomputing* 72(13-15), 2729–2730 (2009)
3. Böhmer, M., Bauer, G., Krüger, A.: Exploring the design space of context-aware recommender systems that suggest mobile applications. In: 2nd Workshop on Context-Aware Recommender Systems (2010)
4. Bellotti, V., Begole, B., Chi, H.E., Ducheneaut, N., Fang, J., Isaacs, E., King, T., Newman, W.M., Partridge, K., Price, B., Rasmussen, P., Roberts, M., Schiano, J.D., Walendowski, A.: Activity-based serendipitous recommendations with the Magitti mobile leisure guide. In: CHI 2008 Proceedings, pp. 1157–1166 (2008)
5. Kwapisz, R.J., Weiss, M.G., Moore, A.S.: Activity recognition using cell phone accelerometers. In: 4th ACM SIGKDD International Workshop on Knowledge Discovery from Sensor Data (2010)
6. Longstaff, B., Reddy, S., Estrin, D.: Improving activity classification for health applications on mobile devices using active and semi-supervised learning. *Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pp. 1–7 (2010)
7. Maguire, D., Frisby, R.: Comparison of feature classification algorithm for activity recognition based on accelerometer and heart rate data. In: 9th. IT & T Conference (2009)
8. Györbíró, N., Fábián, Á., Homány, G.: An activity recognition system for mobile phone. *Mobile Netw. Appl.* 14, 82–91 (2008)
9. Song, K.S., Jang, J., Park, S.: A phone for human activity recognition using triaxial acceleration sensor. In: Digest of Technical Papers - IEEE International Conference on Consumer Electronics (2008)

10. Zappi, P., Lombriser, C., Stiefmeier, T., Farella, E., Roggen, D., Benini, L., Tröster, G.: Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection. In: Verdone, R. (ed.) EWSN 2008. LNCS, vol. 4913, pp. 17–43. Springer, Heidelberg (2008)
11. Yang, J.-Y., Chen, Y.-P., Lee, G.-Y., Liou, S.-N., Wang, J.-S.: Activity recognition using one triaxial accelerometer: A neuro-fuzzy classifier with feature reduction. In: Ma, L., Rauterberg, M., Nakatsu, R. (eds.) ICEC 2007. LNCS, vol. 4740, pp. 395–400. Springer, Heidelberg (2007)
12. Ganti, K.R., Jayachandran, P., Abdelzaher, F.T., Stankovic, A.J.: Satire: a software architecture for smart attire. In: Proceedings of the 4th International Conference on Mobile Systems, Applications and Services, pp. 110–123 (2006)
13. Korb, B.K.: A Bayesian Artificial Intelligence. Chapman & Hall/CRC Computer Science & Data Analysis, Boca Raton (2004)
14. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
15. Yang, S.-I., Hong, J.-H., Cho, S.-B.: Activity recognition based on multi-modal sensors using dynamic Bayesian networks. J. KIISE: Computing Practices and Letters 15(1), 72–76 (2009)
16. Park, S., Aggarwal, K.J.: A hierarchical Bayesian network for event recognition of human actions and interactions. Multimedia Systems 10(9), 164–179 (2004)
17. Mengistu, T.K., Hannemann, M., Baum, T., Wendemuth, A.: Hierarchical HMM-based semantic concept labeling model, pp. 57–60. IEEE Computer Society Press, Los Alamitos (2008)
18. Wang, A.W.-H., Tung, C.-L.: Dynamic gesture recognition based on dynamic Bayesian networks. WSEAS Transactions on Business and Economics 4(11), 168–173 (2008)
19. Du, Y., Chen, F., Xu, W., Zhang, W.: Interacting activity recognition using hierarchical durational-state dynamic Bayesian network. In: Zhuang, Y.-t., Yang, S.-Q., Rui, Y., He, Q. (eds.) PCM 2006. LNCS, vol. 4261, pp. 185–192. Springer, Heidelberg (2006)

Author Index

- Aamodt, Agnar II-197
Abracham, Ajith I-30
Abraham, Ajith I-372
Aguilar-Ruiz, Jesús S. II-279, II-303
Alonso, J.B. I-75
Alvarez-Hernández, Victor M. II-26
Amoza, Franco Robledo II-42
Arabas, Jarosław I-263
- Banković, Z. I-214
Barbosa, Juan Javier González II-18
Bartocha, Kamil II-164
Baruque, Bruno II-363
Bastiani Medina, Shulamith S. II-18
Becerra, J.A. II-471
Bellas, F. II-471
Berlanga, Antonio II-136
Bernardos, Ana M. II-118, II-127
Bertinat, María Elisa II-42
Berzosa, Alba II-84
Besada, Juan A. II-118
Blachnik, Marcin I-222
Bogdanović, Vuk I-83
Borges, Fabio A.S. I-347
Brzeziński, Dariusz II-155
Burduk, Anna II-381, II-389
Burduk, Robert II-245
Buza, Krisztian II-253
Byrdziak, Adrian I-222
Bystrzycki, Orestes I-222
- Caamaño, P. II-471
Calvo-Rolle, José Luis II-327, II-352
Camara, Monica II-437
Cano, Alberto I-172, I-388
Casar, José R. II-127
Cases, Blanca II-479
Chaves, R. I-132, I-148
Chen, Pei-Yu I-51
Chira, Camelia I-91, I-380, II-10
Chlebus, Edward II-373, II-381
Chmielnicki, Wiesław II-205
Cho, Sung-Bae I-444, I-452, I-460
Choinski, Dariusz I-124
- Cilla, Rodrigo II-136
Corchado, Emilio II-352, II-363, II-437
Cruz-Reyes, Laura II-26
Cucchiara, R. I-239
Cyganek, Bogusław I-436
- d'Anjou, A. II-447
de Carvalho Jr., José G. II-92
de la Cal, Enrique II-84
de la Cruz, J.M. II-180
de Miguel, Gonzalo II-118
de Oliveira Penna Tavares, Gabriela I-32
de S. Lemos, Marcus Vincius I-347
de Souza, Jano M. II-92
Díaz-Díaz, Norberto II-279
Ditzel, Maarten II-100
Divina, Federico II-303
Domínguez, Juan Luis II-271
Duro, R.J. II-471
- Elías, Arturo II-1
Esgin, Eren I-296
- Fernández, A. I-1
Fernandez-Gauna, Borja II-455, II-463
Filho, Raimir Holanda I-347
Foresti, Gian Luca II-110
Fornaciari, M. I-239
Fraga, D. I-214
Fraire Huacuja, Héctor Joaquín II-18
Frontera, Guillermo II-118
Fuertes, Juan José I-75
- Garcia, Alvaro Enrique II-437
García, Jesús II-144
García, Ramón Ferreiro II-327, II-352
García, Salvador I-1, II-262
García-Gutiérrez, Jorge II-311
García-Tamargo, Marco II-84
Gavshin, Yuri I-313
Gog, Anca I-380, II-10
Gómez-Río, M. I-132
Gómez-Santillán, Claudia II-26
Gómez-Vela, Francisco II-279

- Gongora, Mario A. I-198, I-231
 González-Ferreiro, Eduardo II-311
 González-Nalda, Pablo II-479
 Górriz, Juan M. I-99, I-132, I-148
 Grabowik, Cezary II-405
 Graña, Manuel II-336, II-344, II-447, II-455, II-463
 Granmo, Ole-Christoffer I-11
 Gródek, Mateusz I-222
 Gruszczyński, Sławomir I-436
 Guijarro, María II-180
- Hajdu, Andras II-189
 Hajdu, Lajos II-189
 Hanckmann, Patrick II-100
 Heinze, Theodor II-413
 Heras, Stella I-396
 Hernando, Beatriz II-437
 Herrera, Francisco I-1, II-262
 Herrera, P. Javier II-180
 Hopgood, Adrian A. I-198
 Houeland, Tor Gunnar II-197
- Illán, I. I-148
- Jackowska-Strumillo, Lidia I-356
 Jankowski, Jaroslaw I-338
 Jaramillo-Vacio, Rubén I-91
 John, Robert I-231
 Jonas, Agnes II-189
 Jöns, S. I-91
 Jordán, Jaume I-396
 Józefowska, Joanna I-288
 Józefowski, Lukasz I-288
 Julián, Vicente I-396
- Kajdanowicz, Tomasz II-221
 Kalinowski, Krzysztof II-405
 Kawanabe, Motoaki II-34, II-51
 Kazienko, Przemyslaw II-221
 Klepaczko, Artur I-140
 Kopiec, Dawid I-329
 Kordos, Mirosław I-222
 Kosmala, Jan I-404
 Kovacs, Laszlo II-189
 Kowalski, Arkadiusz II-381
 Kozłowski, Jacek I-222
 Krenczyk, Damian II-397
 Królak, Aleksandra I-305
 Krömer, Pavel I-372
- Krot, Kamil II-373
 Kruusmaa, Maarja I-313
 Kucharzak, Michał I-364
 Kuliberda, Michał II-373
 Kulikowski, Juliusz L. I-22
 Kurzynski, Marek II-229
 Kwolek, Bogdan I-271
- Laham, Amer II-352, II-437
 Lang, Elmar I-99
 Lasota, Tadeusz I-107, II-213
 Ławrynowicz, Agnieszka I-288
 Leal, Liam B. I-347
 Ledezma-Orozco, Sergio I-91
 Lee, Young-Seol I-460
 Lin, Frank Yeong-Sung I-51
 Lisiecki, Andrzej I-288
 Llinas, James I-31
 López, M. I-132, I-148
 Lopez-Gude, Jose Manuel II-455, II-463
 Ludermir, Teresa B. I-164
 Lughofer, Edwin I-107
 Lukaszewski, Tomasz I-288
 Luna, J.M. I-388
 Lung, Rodica Ioana II-75
 Lysiak, Rafal II-229
- Maciejewski, Henryk I-321
 Maiora, Josu II-344
 Maña, Manuel J. II-271
 Markowska-Kaczmar, Urszula I-420
 Márquez Chamorro, Alfonso E. II-303
 Martínez-Álvarez, F. II-287
 Martínez-Ballesteros, M. II-319
 Martyna, Jerzy I-329
 Mata, Jacinto II-271
 Matei, O. II-67
 Mateos-García, Daniel II-311
 Matthews, Stephen G. I-198
 Mello, P. I-239
 Metzger, Mieczysław I-124
 Miller, Simon I-231
 Milutinović, Dragana II-421
 Min, Jun-Ki I-444
 Minda, Paweł I-420
 Miranda, David II-311
 Molina, José M. II-136, II-144
 Morales-Esteban, A. II-287

- Moreno, R. II-447
 Motyka, Zenon I-222
 Moya, J.M. I-214
 Nammous, Mohammad Kheir I-412
 Nanopoulos, Alexandros II-253
 Nenortaite, Jovita I-247
 Nocon, Witold I-124
 Ochoa-Zezzatti, Alberto I-91, II-1
 Ociepa, Krzysztof I-420
 Olivares, A. I-148
 Olmo, J.L. I-388
 Olszowy, Dariusz I-420
 Oommen, B. John I-11, I-43
 Owczarek, Agnieszka I-206
- Pacheco, Marco Aurelio Cavalcanti I-32
 Pachón, Victoria II-271
 Padilla, Alejandro II-1
 Padula, Darío II-42
 Pajares, Gonzalo II-180
 Pal, Anshika I-190
 Pankowski, Tadeusz I-255
 Pascual, Javier II-51
 Patricio, Miguel A. II-136, II-144
 Pawlikowski, Roman I-420
 Pazos R, Rodolfo A. II-18
 Pérez-Rosas, Verónica II-26
 Perzyk, Marcin I-222
 Piwowarski, Mateusz I-338
 Płaczek, Bartłomiej II-59
 Platoš, Jan I-372
 Podolak, Igor T. I-156, II-164
 Polańczyk, Maciej I-206
 Ponce, Julio II-1
 Pop, P.C. I-67
 Porras, Santiago II-363
 Porwik, Piotr I-428
 Prati, A. I-239
 Prieto, A. II-471
 Prudêncio, Ricardo B.C. I-164
 Przytulska, Małgorzata I-22
 Quiroz-Castellanos, Marcela II-26
 Rabelo, Ricardo A.L. I-347
 Ramírez, Javier I-99, I-132, I-148
 Ramirez, Milton R. II-92
 Rangel, Pablo II-92
- Redondo, Raquel II-437
 Riquelme, J.C. II-287, II-319
 Riquelme-Santos, Jose C. II-311
 Rodriguez-Baena, Domingo S. II-279
 Rodríguez-Bocca, Pablo II-42
 Roman, A. I-156
 Romero, Pablo II-42
 Rondio, Jacek I-305
 Ruškić, Nenad I-83
 Rymut, Boguslaw I-271
- Saeed, Khalid I-404, I-412
 Salas-Gonzalez, Diego I-99, I-132, I-148
 Santillan, Claudia G. Gómez II-18
 Savio, Alexandre II-336
 Schacht, Alexander II-413
 Schaeffer, Satu Elisa II-26
 Schlögl, Matthias I-99
 Schmidt-Thieme, Lars II-253
 Sedano, Javier II-84, II-437
 Segovia, F. I-148
 Senkul, Pinar I-296
 Serrano, Miguel A. II-144
 Shukla, Anupam I-190
 Simić, Dragan II-421, II-429
 Simić, Svetlana II-421, II-429
 Skolud, Bozena II-397
 Ślot, Krzysztof I-206
 Slowik, Adam I-59, I-67
 Smętek, Magdalena I-116
 Smilgyte, Kristina I-247
 Snášel, Václav I-372
 Snidarо, Lauro II-110
 Soares, Carlos I-164
 Sosnowski, Maciej I-428
 Sottara, D. I-239
 Sremac, Siniša I-83
 Stańczyk, Urszula II-172, II-295
 Stąpor, Katarzyna II-205
 Stefanowski, Jerzy II-155
 Strug, Barbara I-280
 Strzelecki, Michał I-140, I-206
 Suknaja, Vesna II-421
 Szczepański, Adam I-412
 Szczypinski, Piotr I-140
- Tanackov, Ilija I-83, II-429
 Tarrío, Paula II-127
 Telec, Zbigniew II-213

- Tepić, Jovan I-83
Tiwari, Ritu I-190
Toman, Henrietta II-189
Travieso, Carlos M. I-75
Trawiński, Bogdan I-107, I-116, II-213
Trawiński, Grzegorz II-213
Trawiński, Krzysztof I-107
Triguero, Isaac II-262
Troncoso, A. II-287
- Valdez, Guadalupe Castilla II-18
Valero, Soledad I-396
Vallejo, J.C. I-214
van den Broek, Sebastiaan II-100
van Iersel, Miranda II-100
Vaquerizo, Belén II-363
Ventura, Sebastián I-172, I-388
Vera, Vicente II-437
Verkhogliad, Petro I-43
Vidaurre, Carmen II-34, II-51
Villanúa, Jorge II-336
Villanueva, David Terán II-18
Villar, José R. II-84
- Visentini, Ingrid II-110
von Löwis, Martin II-413
- Walkowiak, Krzysztof I-182, I-364
Waluś, Michał I-404
Wang, Xian II-127
Watrobski, Jarosław I-338
Wen, Ya-Fang I-51
Wesolowski, Tomasz I-428
Wierschke, Robert II-413
Wilk, Tomasz II-237
Wiśniewski, Michał I-182
Wojcikiewicz, Wojciech II-34
Woloszynski, Tomasz II-229
Woźniak, Michał II-237
Wrobel, Krzysztof I-428
- Yazidi, Anis I-11
Yen, Hong-Hsu I-51
Yoon, Jong-Won I-452
- Zafra, Amelia I-172
Zawistowski, Paweł I-263
Zulueta, Ekaitz II-455